INAUGURAL-DISSERTATION

zur Erlangung der Doktorwürde
der Naturwissenschaftlich-Mathematischen Gesamtfakultät
der Ruprecht-Karls-Universität Heidelberg

vorgelegt von
Master of Science Francesco Silvestri
aus Eberbach am Neckar

Tag der mündlichen Prüfung: .....

Representations of Partition Problems and the Method of Moments

Betreuer:
Prof. Dr. Gerhard Reinelt
Prof. Dr. Christoph Schnörr

to my parents

Zusammenfassung

Die vorliegende Dissertation verfolgt zwei Ziele. Das erste besteht darin, verschiedene Darstellungen von Partitionen zu formulieren, zu erklären und miteinander in Verbindung zu bringen, die zur Modellierung von Partitionsproblemen verwendet werden können. Das zweite besteht darin, durch Einsatz der Momentmethode, einem Ansatz aus der polynomiellen Optimierung, konvexe Relaxierungen dieser Darstellungen zu konstruieren, um die globalen Optimallösungen der zugehörigen Partitionsprobleme abzuschätzen. Für Probleme wie dem Euklidisches k-Clustering weicht diese Methodik stark von den herkömmlichen Ansätzen ab, die sich vorwiegend mit Heuristiken und lokalen Optima beschäftigen. Da die Momentmethode konvexe Probleme konstruiert, liegt der Fokus der Arbeit darauf, Darstellungen zu finden und auszunutzen, deren Lösungsräume nur eine triviale Symmetriegruppe zulassen, damit die Lösungen der Relaxierung zu zulässigen Lösungen gerundet werden können.

Die in dieser Arbeit behandelten Darstellungen sind Assignment-, Partitions- und Projektionsmatrizen, sowie simpliziale Überdeckungen für eine verallgemeinerte Version des Euklidischen k-Clustering. Es werden Verbindungen sowie Übergänge zwischen den Matrizenklassen hergestellt und mit der Literatur verglichen, und es wird explizit nachgewiesen, wie Partitionsmatrizen auf natürliche Weiße durch die Momentmethode aus Assignmentmatrizen hervorgehen.

Das Ausnutzen von Projektionsmatrizen ermöglicht es uns eine neue Formulierung für die Färbungszahl einzuführen, und die aus der Momentmethode resultierenden Relaxierungen werden mit der Lovász Theta Zahl verglichen. Es wird charakterisiert, unter welchen Bedingungen beide Relaxierungen übereinstimmen, und als erstes Hauptresultat liefern Relaxierungen von Binärmatrizen in diesem Fall bessere Ergebnisse als Relaxierungen von binären Eigenwerten.

Der letzte Teil der Arbeit beschäftigt sich mit dem sogenannten affinen Euklidischen k-Clustering, welches das Euklidische k-Clustering verallgemeinert. Als zweites Hauptresultat der Arbeit wird eine neue Methodik für dieses anspruchsvolle Problem eingeführt, die simpliziale Überdeckungen des Lösungsraums ausnutzt um eindeutige Darstellungen der Optimallösungen des zugrunde liegenden Problems zu ermöglichen. Im Gegensatz zum direkten Einsatz der Momentmethode auf die Standardformulierung ermöglicht der Einsatz auf dieser Formulierung einen langsameren Anstieg der Problemgröße, bessere Parallelisierbarkeit sowie die Möglichkeit Informationen als Grundlage für eine Rundungsheuristik zu erhalten, was aus Symmetriegründen bei der Standardformulierung nicht möglich ist.

Abstract

The thesis follows two main goals. The first is to formulate, explain and link representations of partitions that can be used to model partition problems. The second is to use the method of moments, an approach from polynomial optimization, to bound the global optima of the corresponding partition problems by constructing convex relaxations of these representations. For problems like Euclidean k-clustering, this is a stark contrast to their usual treatment, which mostly involves heuristics that are content with local optima. Since the method of moments results in a convex approach, the focus lies on finding and exploiting representations that lack a non-trivial symmetry-invariant solution space in order to be able to round the relaxations to feasible solutions.

The representations considered in the thesis are assignment matrices, partition matrices, projection matrices and simplicial covers for a generalized version of Euclidean k-clustering. Connections and transformations between the matrix classes are established and compared to the literature, and it is explicitly shown how partition matrices arise naturally from assignment matrices through the method of moments.

Using projection matrices, we are able to give a new formulation of the colouring number, and the resulting relaxations from the method of moments are compared to the Lovász theta number. We characterize under which circumstances the relaxations agree and explain when they do not, indicating our first main result that in this case, relaxing binary matrix entries yields better results than relaxing binary eigenvalues.

The final part of the thesis is devoted to what we call the affine Euclidean k-clustering problem, which is a more general version of the Euclidean k-clustering problem. As our second main result of the thesis, we introduce a new method for this challenging problem, utilizing simplicial covers of the feasible region to formulate unique representations of the optimal solutions of the underlying problem. In contrast to applying the method of moments directly, applying it to our formulation yields a slower growth in size, better parallelizability and enables us to recover information that can be used for rounding, which is not possible for the standard formulation due to symmetry.

Acknowledgement

This thesis could not have been written without the support of various people:

First, I want to thank my supervisor Prof. Gerhard Reinelt for introducing me to the fascinating topic of combinatorial optimization as a student, and for guiding me along my academic career from bachelor student to Ph.D. student, while always encouraging me to follow my interests and generously supporting my endeavors.

I also want to thank my second supervisor Prof. Christoph Schnörr for his generous support and input on various topics of this thesis, and in particular for sharing his perspective on the affine Euclidean clustering problem encountered in image analysis.

Next, I would like to thank the secretaries Catherine Proux-Wieland, Evelyn Wilhelm and Barbara Werner, who did a fantastic job and helped with all administrative tasks that occurred during my time at the Faculty of Mathematics and Computer Science of Heidelberg University.

I am very grateful for the time I was able to spend with all the wonderful people from the *Combinatorial Optimization* and the *Image & Pattern Analysis* groups, resulting in great memories and fruitful discussions both in and outside of work. In particular, I would like to thank Achim Hildenbrandt, Artjom Zern, Jan Kuske, Mattia Desana, Robert Breckner and Tobias Dencker for our time working together and the nice atmosphere they helped to shape, which I will remember for a long time.

Thanks for the support and funding of the German Research Foundation (DFG). I was member of the Research Training Group (RTG) 1653 "Spatio/Temporal Probabilistic Graphical Models and Applications in Image Analysis", which I acknowledge gratefully.

I also very much enjoyed my interactions with the people from AAU in Klagenfurt during my trips to visit Prof. Rendl, and I would like to thank him for introducing me to the topics of semidefinite optimization, as well as for his support. In the same vein, I'd like to thank all people that I met during the HeKKSaGOn project that allowed me to go to Kyoto, and Prof. Fujishige for introducing me to the LP-Newton method.

I'd like to thank my friends for the time we spent together, especially during our board game nights, which greatly helped me to relax. For proofreading parts of this thesis I thank Artjom Zern, Nadine Bär, Peter Gräf and Thomas Hölters, whose comments helped a lot to improve this thesis, and to ease my mind.

My parents Rita Silvestri and Pasquale Silvestri as well as my brother Alessandro Silvestri deserve my heartfelt thanks for their unconditional support throughout my life. At the same time, I thank Nadine Bär for her loving support and encouragement, as well as for the great time we have been spending together. She contributed a lot to making my time as a Ph.D. student a precious experience.

Contents

List of Figures

List of Tables

List of Algorithms

List of Symbols

Basics

$\mathbb{N}$	non-negative integers
$\mathbb{N}_t^d$	d-dimensional non-negative integer vectors whose sum is at most t
$\mathbb{R}$	real numbers
$\mathbb{R}_+$	non-negative real numbers
2^S	power set of the finite set S
$[n]$	the set $\{1, 2, \ldots, n\}$ for $n \in \mathbb{N}$, $n \geq 1$
$[\mathbf{U}]$	orbit of $\mathbf{U}$ under a group action *or* equivalence class
$\mathfrak{S}_n$	set of permutations on $[n]$

Partitions

$\mathcal{P}^n(L)$	set of all partitions of $[n]$ with parts restricted to $L \subseteq 2^{[n]}$
$\mathcal{P}_k^n(L)$	set of all partitions of $[n]$ with k parts restricted to $L \subseteq 2^{[n]}$
$\mathcal{U}_{n,k}(L)$	set of $n \times k$ assignment matrices arising from $\mathcal{P}_k^n(L)$
$\mathcal{U}_{n,k}^{lex}(L)$	set of matrices in $\mathcal{U}_{n,k}(L)$ with lexicographic sorted columns
$\mathrm{PM}_k^n(L)$	set of $n \times n$ partition matrices arising from $\mathcal{P}_k^n(L)$
$\mathrm{CProM}_k^n(L)$	set of projection matrices arising from $\mathcal{P}_k^n(L)$

Algebra

$\boldsymbol{x}^{\boldsymbol{\alpha}}$	the monomial $\prod_{i \in [d]} x_i^{\alpha_i}$ for $\boldsymbol{\alpha} \in \mathbb{N}^d$, $\boldsymbol{x} \in \mathbb{R}^d$
$\mathbb{R}[\boldsymbol{x}]$	ring of real multivariate polynomials in $\boldsymbol{x} = (x_1, \ldots, x_d)$
$\mathbb{R}_t[\boldsymbol{x}]$	vector space of polynomials in $\mathbb{R}[\boldsymbol{x}]$ of degree at most t
$z_d(t)$	dimension of $\mathbb{R}_t[\boldsymbol{x}]$ where $\boldsymbol{x} = (x_1, \ldots, x_d)$, equal to $\binom{d+t}{d}$
$v_t(\boldsymbol{x})$	vector of moments of $\boldsymbol{x} \in \mathbb{R}^d$ up to degree t
$\mathrm{td}(f)$	truncated degree of a multivariate polynomial f given by $\left\lceil \frac{\deg(p)}{2} \right\rceil$

Combinatorial Objects

$\mathcal{G}_n$	set of all simple graphs with vertex set $[n]$
L	an independence system
L_*	an independence system without the empty set
$\mathcal{T}$	the partition $\{T_1, \ldots, T_k\}$
$\left\{ {n \atop k} \right\}$	Stirling number of the second kind

Convex Cones

$\mathcal{K}$	a proper cone
$\mathbb{R}^n_+$	cone of vectors in $\mathbb{R}^n$ with non-negative entries
$\mathcal{S}^n_+$	cone of positive semidefinite $n \times n$ matrices
$\mathcal{L}_n$	cone of vectors $(x_0, \boldsymbol{x}) \in (\mathbb{R}_+ \times \mathbb{R}^n)$ satisfying $\|\boldsymbol{x}\|_2 \leq x_0$

Polytopes

C^d_k	k-truncated d-dimensional unit cube
Δ^d_Ω	Ω-constrained standard simplex in $\mathbb{R}^d$

Vectors / Matrices

$\boldsymbol{e}_I$	characteristic vector of $I \subseteq [n]$
$\boldsymbol{e}_i$	shorthand for $\boldsymbol{e}_{\{i\}}$
$\boldsymbol{e}$	shorthand for $\boldsymbol{e}_{[n]}$ if n is clear from context
$\mathbf{I}_n$	$n \times n$ identity matrix
$\mathbf{J}_{n,m}$	$n \times m$ matrix of all ones defined as $\mathbf{J}_{n,m} = \boldsymbol{e}_{[n]}\boldsymbol{e}_{[m]}^\top$
$\mathbf{\Omega}_L$	$n \times n$ binary matrix defined by independence system $L \subseteq 2^{[n]}$
$\mathbf{A}_G$	$n \times n$ adjacency matrix of the graph $G \in \mathcal{G}_n$
$\mathbf{M}_t(f, \boldsymbol{y})$	localizing moment matrix of order t of $\boldsymbol{y}$ with respect to f

Relations

$\subseteq$	inclusion as subset (with possible equality)
$\leftrightarrow$	existence of a bijection between two sets
$\leq$	conic order induced by $\mathbb{R}^n_+$ *or* subgraph relation
$\preceq$	conic order induced by $\mathcal{S}^n_+$
$\leq_\mathcal{K}$	conic order induced by $\mathcal{K}$
$<_{lex}$	lexicographic monomial order on $\mathbb{N}^d$

Maps / Operators

$\wedge$	join in the context of orders, logical AND otherwise
$\vee$	meet in the context of orders, logical OR otherwise
$\otimes$	tensor product for matrices *or* product of independence systems
δ_Ψ	Kronecker delta assuming the truth values 0/1 of Ψ
$*$	group action
$\operatorname{rank}(\mathbf{X})$	the rank of the matrix $\mathbf{X}$
$\operatorname{col}(\mathbf{X})$	the set of column vectors of the matrix $\mathbf{X}$
$\operatorname{tr}(\mathbf{X})$	trace of the quadratic matrix $\mathbf{X}$
$\operatorname{spct}(\mathbf{X})$	set of eigenvalues of the quadratic matrix $\mathbf{X}$
$\operatorname{diag}(\mathbf{X})$	diagonal of the square matrix $\mathbf{X}$ as vector
$\operatorname{Diag}(\boldsymbol{x})$	diagonal matrix $\mathbf{X}$ with diagonal equal to $\boldsymbol{x}$
$\operatorname{supp}(\boldsymbol{x})$	set of indices $i \in [n]$ for which $\boldsymbol{x} \in \mathbb{R}^n$ is non-zero
$\lVert\boldsymbol{x}\rVert_0$	shorthand for $\lvert\operatorname{supp}(\boldsymbol{x})\rvert$
$\operatorname{dist}(\boldsymbol{x}, C)$	Euclidean distance between $\boldsymbol{x} \in \mathbb{R}^n$ and the convex set $C \subseteq \mathbb{R}^n$
$\pi_C(\boldsymbol{x})$	orthogonal projection from $\boldsymbol{x} \in \mathbb{R}^n$ onto the convex set $C \subseteq \mathbb{R}^n$
$\operatorname{bd}(C)$	boundary of the set $C \subseteq \mathbb{R}^n$
$\operatorname{int}(C)$	interior of the set $C \subseteq \mathbb{R}^n$
$\operatorname{lin}(C)$	linear span of the set $C \subseteq \mathbb{R}^n$
$\operatorname{aff}(C)$	affine hull of the set $C \subseteq \mathbb{R}^n$
$\operatorname{conv}(C)$	convex hull of the set $C \subseteq \mathbb{R}^n$
$\operatorname{bar}(C)$	barycenter of C
Id_C	identity map on C
$\operatorname{vert}(P)$	set of vertices of a polytope P
$\operatorname{CC}(G)$	set of connected components of graph G

Abbreviations

MM	method of moments
psd.	positive semidefinite

Chapter 1

Introduction

Overview

In this thesis, we are interested in the general problem of optimally partitioning a set of mathematical objects according to a given criterion. A prominent geometric example for this is *Euclidean k-clustering*, where the goal is to partition a set of points $\{\boldsymbol{b}_i \mid i \in [n]\} \subseteq \mathbb{R}^d$ into k parts such that each part has a center $\boldsymbol{x}^j$ with low average Euclidean distance towards its members, as can be seen in Figure 1.1.

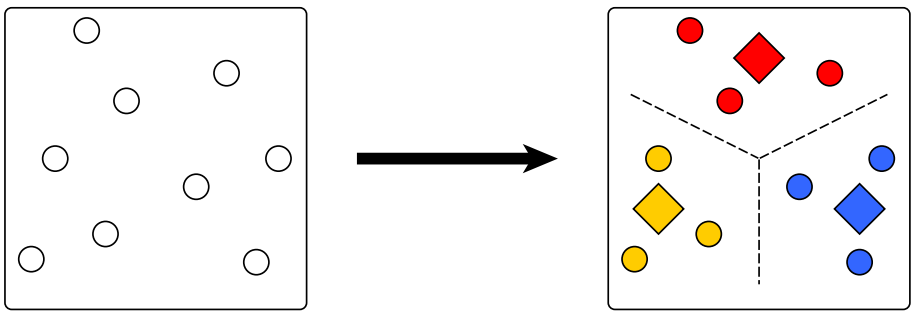

Figure 1.1: Example of Euclidean 3-clustering. Circles correspond to the input $\boldsymbol{b}_i$, diamonds to centers $\boldsymbol{x}^j$ and colours to parts.

Partition problems constitute a large class of hard and well-studied problems with underlying combinatorial structure and are ubiquitous in the fields of data science. A purely combinatorial example for a notoriously hard problem belonging to this class is the one of *graph-colouring*. Given a simple graph G, the goal is to find the least amount of colours necessary to colour each node, such that the same colour is never connected with an edge. Figure 1.2 shows some minimal colourings for small graphs.

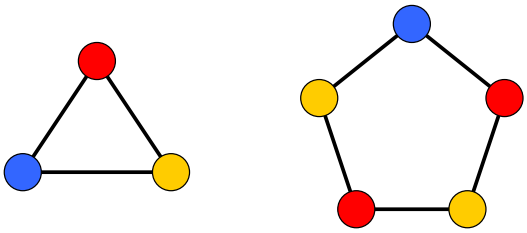

Figure 1.2: Examples of minimal colourings.

Unfortunately, our examples, as well as a lot of other interesting partition problems, are NP-hard, and a lot of work has been done to tackle these problems. Investigating these problems, it quickly becomes apparent that a key difficulty of formulating partition problems is their inherent symmetrical structure. While a partition is mathematically a *set* of parts, a computer necessarily needs to store the parts in some order, thus representing a partition as a *list* of parts. This transition introduces a fixed, but arbitrary order on the parts, which means that any solution has one representation for each permutation of its parts, leading to overblown solution spaces and naturally ill-posed problems.

These symmetries are especially problematic for approaches relying purely on convex optimization, since their optima tend to lie inside the interior of the convex hull of the optimal feasible solutions. In the worst case, such an optimum may not even provide any means to recover an actual feasible solution, as shown in Figure 1.3.

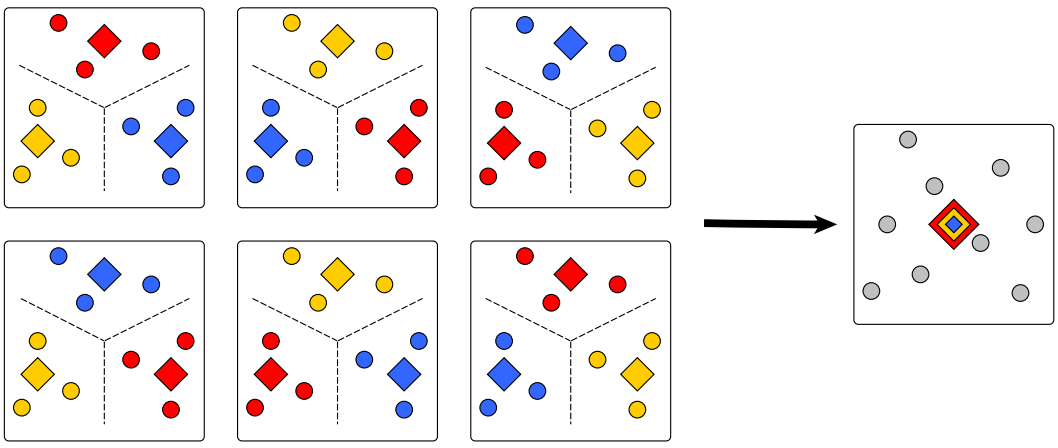

Figure 1.3: Left: All 3! = 6 optimal solutions for the Euclidean 3-clustering instance from Figure 1.1. Right: Convex combination of the optimal solutions both in colours and center locations, with no means to go back to the left.

For this reason, it is important to find proper representations of partitions that circumvent the problems induced by symmetries and are still tractable from a computational point of view as well. There are still many open questions in this regard, and the aim of this thesis is to give some insight in how to approach them.

Organization

This thesis pursues two main goals. The first is to formulate, explain and link representations of partitions. The second is to use the *method of moments*, an approach from polynomial optimization, to construct convex relaxations of these representations, in order to bound the global optima of the corresponding partition problems. For problems like Euclidean k-clustering, this is a stark contrast to their usual treatment, which mostly involves heuristics that are content with local optima.

In Chapter 2, we recall the theoretical results that are used as mathematical foundation throughout the thesis. While mostly self-contained, we keep the explanation of results to a bare minimum, and proofs are omitted. We highly recommend to inspect the listed books there for further information.

Chapter 3 complements Chapter 2 by going into the details of the underlying computational aspects of this thesis. In particular, we explore preprocessing to reduce the problem size, explain the computational setup and describe our adaption of the LP-Newton method for conic linear programs, which may be skipped on a first reading.

Our treatment of partition problems starts in Chapter 4, where they are formally defined. We show how to represent partitions in terms of *assignment matrices* and properly illustrate the issues of symmetrical solutions in a convex setting, as well as a potential fix for this in the form of orbitopes.

In Chapter 5, we review *partition matrices* used in combinatorial optimization, our first matrix class that uniquely represents each partition. In particular, we explicitly show that the method of moments connects them to the assignment matrices from the preceding chapter, a fact that has implicitly been exploited for a long time in literature.

Chapter 6 shows how to transition from partition matrices to *projection matrices*, our second matrix class that uniquely represents each partition. We use this class to introduce a new relaxation for the graph-colouring problem and show how it relates to established relaxations like the *Lovász theta number* $\vartheta(G)$.

Finally, Chapter 7 is devoted to the *affine Euclidean k-clustering problem*, which is a more general version of the Euclidean k-clustering previewed here. As one main result of this thesis, we introduce a new method for this challenging problem, utilizing *simplicial covers* of the feasible region to formulate unique representations of the optimal solutions of the underlying problem.

Parts of this thesis have been published in the following papers:

[SRS15] Francesco Silvestri, Gerhard Reinelt, and Christoph Schnörr. A convex relaxation approach to the affine subspace clustering problem. In *Pattern Recognition - 37th German Conference, GCPR 2015, Aachen, Germany, October 7-10, 2015, Proceedings*, pages 67–78, 2015

[SRS16] Francesco Silvestri, Gerhard Reinelt, and Christoph Schnörr. Symmetry-free SDP relaxations for affine subspace clustering. *ArXiv e-prints*, July 2016

[SR16] Francesco Silvestri and Gerhard Reinelt. The LP-Newton method and conic optimization. *ArXiv e-prints*, November 2016

In particular, Section 3.3 contains material from [SR16], and Chapter 7 contains material from [SRS15, SRS16].

Chapter 2

Preliminaries

This chapter recalls the underlying definitions and most important results that will be used throughout the thesis. It is assumed that the reader has some basic understanding of linear algebra, and proofs are omitted in general, but can be found in the given references.

Notation

The majority of the commonly used notation can be found in the List of Symbols, while the rest will be introduced when appropriate. In general, we will use small letters like a, x, λ for scalars or elements of a set and capital letters like S, T, U for sets. Small bold letters are used for vectors like $\boldsymbol{a}, \boldsymbol{x}, \boldsymbol{\lambda}$ and capital bold letters for matrices like $\mathbf{A}, \mathbf{X}, \mathbf{\Lambda}$. Instead of writing index sets of small size like $\{i\}$ or $\{i, j\}$, we will sometimes simply write i or ij respectively. For a formal statement Ψ, the *Kronecker delta* δ_Ψ assumes the value 1 if the statement Ψ is true and 0 otherwise; for a pair of objects a, b in the same space, $\delta_{a,b}$ denotes the shorthand for $\delta_{a=b}$.

For the sets $S \subseteq \mathbb{R}$ and $C, D \subseteq \mathbb{R}^n$, we use the shorthand set operations

$$S \cdot C := \{s \cdot c \mid s \in S,\ c \in C\},$$
$$C + D := \{c + d \mid c \in C,\ d \in D\},$$

where the latter is known as *Minkowski sum*. In particular, $-C$ is short for $-1 \cdot C$ and corresponds to a point reflection of C at $\mathbf{0}$.

2.1 Linear Algebra

For a quadratic matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, its *trace* is defined as

$$\operatorname{tr}(\mathbf{A}) := \sum_{i \in [n]} a_{ii},$$

which is equivalent to the sum of its eigenvalues. The trace induces an inner product on the space of $n \times m$ matrices called the *trace product*, which is defined by

$$\langle \mathbf{A}, \mathbf{B} \rangle := \operatorname{tr}(\mathbf{A}\mathbf{B}^\top) = \sum_{i \in [n]} \sum_{j \in [m]} a_{ij} b_{ij}.$$

For $m = 1$, this reduces to the usual notion of the scalar product for vectors in $\mathbb{R}^n$.

Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ be invertible and $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^n$. Then the rank 1 update $\mathbf{A} + \boldsymbol{u}\boldsymbol{v}^\top$ is invertible if and only if $1 + \boldsymbol{v}^\top \mathbf{A}^{-1} \boldsymbol{u} \neq 0$. Furthermore, if the inverse exists, it is given by the *Sherman-Morrison formula*

$$(\mathbf{A} + \boldsymbol{u}\boldsymbol{v}^\top)^{-1} = \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1}\boldsymbol{u}\boldsymbol{v}^\top \mathbf{A}^{-1}}{1 + \boldsymbol{v}^\top \mathbf{A}^{-1} \boldsymbol{u}}. \tag{2.1}$$

2.2 Combinatorial Structures

Graphs

A good introduction for graphs can be found in the textbook [Die05]. A *(simple) graph* G is a tuple (V, E) consisting of a finite set V called *nodes* (or vertices) and a set

$$E \subseteq (V \times V) \setminus \{(v, v) \mid v \in V\}$$

called *edges*. A *(graph) homomorphism* from graph $H = (V_H, E_H)$ to graph $G = (V_G, E_G)$ is a map $\varphi : V_H \to V_G$ such that $(i, j) \in E_H$ if and only if $(\varphi(i), \varphi(j)) \in E_G$. A graph H is called a *subgraph* of a graph G, which we denote by $H \leq G$, if there is an injective homomorphism from H to G. For a graph $G = (V_G, E_G)$ and a subset $U \subseteq V_G$, the subgraph $G[U]$ induced by U is defined as the graph $G[U] := (U, E_G \cap (U \times U))$.

A bijective graph homomorphism is called a *(graph) isomorphism*, and a graph isomorphism from a graph to itself is called a *(graph) automorphism*. A graph $G = (V, E)$ is called *vertex-transitive*, if, for every pair of nodes $i \neq j \in V$, there exists an automorphism φ_{ij} of G such that $\varphi_{ij}(i) = j$.

Given a graph $G = (V, E)$ whose node set has finite size $|V| = n \in \mathbb{N}$, we can always relabel the nodes as natural numbers by constructing a graph homomorphism from V to $[n]$ with appropriate edge set. For this reason, we can define the set of graphs of size n as

$$\mathcal{G}_n := \{G \mid G = ([n], E) \text{ is a graph } \},$$

and define with $E(G)$ the set of edges of G.

Given a graph $G \in \mathcal{G}_n$, its *adjacency matrix* is defined as the unique symmetrical $n \times n$ binary matrix $\mathbf{A}_G$ with the property that $(\mathbf{A}_G)_{ij} = 1$ if and only if $\{i, j\} \in E$. In particular, this assignment yields a bijection for

$$\mathcal{G}_n \leftrightarrow \left\{ \mathbf{A} \in \{0, 1\}^{n \times n} \;\middle|\; \mathbf{A} = \mathbf{A}^\top, \operatorname{diag}(\mathbf{A}) = \mathbf{0} \right\}.$$

The *complement (graph)* of a graph $G \in \mathcal{G}_n$ is the graph $\overline{G} \in \mathcal{G}_n$ satisfying

$$\mathbf{A}_G + \mathbf{A}_{\overline{G}} + \mathbf{I}_n = \mathbf{J}_n.$$

For a pair of nodes $i, j \in [n]$ in a graph $G \in \mathcal{G}_n$, an (i,j)-path is a non-repeating sequence of nodes $v_0, v_1, \ldots, v_m$ such that $(v_{k-1}, v_k) \in E(G)$ for all $k \in [m]$ and $v_0 = i$, $v_m = j$. The pair (i,j) is called *connected* if there exists an (i,j)-path, and being connected is an equivalence relation. The corresponding equivalence classes are called *connected components*, and denoting the set of all connected components of G by $\mathrm{CC}(G)$ induces a surjection

$$\mathrm{CC} : G \mapsto \mathrm{CC}(G) \in \mathcal{P}^n$$

from $\mathcal{G}_n$ to $\mathcal{P}^n$, the set of partitions of $[n]$. A graph $G \in \mathcal{G}_n$ is called *bipartite* if its nodes $[n]$ can be partitioned into two sets U, V such that $E(G[U]) = E(G[V]) = \emptyset$.

Independence Systems

A subset $L \subseteq 2^{[n]}$ is called an *independence system* or *abstract simplicial complex* if it is closed under taking subsets; more formally, the defining quality is the implication

$$I \in L \implies 2^I \subseteq L. \tag{2.2}$$

Without loss of generality, we will assume that whenever $L \subseteq 2^{[n]}$ is an independence system, then $\{i\} \in L$ for all $i \in [n]$; otherwise, we could consider L as independence system of a strictly contained subset of $[n]$.

It will often be useful to consider an independence system without the empty set; for this reason, we define $L_* = L \setminus \{\emptyset\}$.

Independence systems are closed under taking unions in the sense that whenever $L \subseteq 2^J$ and $L' \subseteq 2^{J'}$ are independence systems, so is the set of individual unions

$$L \otimes L' := \{I \cup I' \mid I \in L,\, I' \in L'\} \subseteq 2^{J \cup J'}.$$

Abusing notation, we will write L^k for the independence system emerging from the independence system $L \subseteq 2^{[n]}$ by using this construction on the union of k distinct copies of $[n]$, and likewise L_*^k for the construction with L_*, where we stress that $(L^k)_* \neq L_*^k$.

For later, we will also introduce the matrix $\Omega_L = \{\omega_{ij}\}_{i,j \in [n]} \in \{0,1\}^{n \times n}$, which we define pointwise as

$$\omega_{ij} = \begin{cases} 1 & \text{if } \{i,j\} \notin L, \\ 0 & \text{else.} \end{cases}$$

We can consider the matrix Ω_L as the adjacency matrix of the graph that represents forbidden pairs in L.

Order Theory

A (non-strict) *partial order* on a set C is a binary relation $\trianglelefteq$ on C that satisfies

(i) $a \trianglelefteq a$ (reflexivity),

(ii) $a \trianglelefteq b$ and $b \trianglelefteq a$ imply $a = b$ (antisymmetry),

(iii) $a \trianglelefteq b$ and $b \trianglelefteq c$ imply $a \trianglelefteq c$ (transitivity)

for all $a, b, c \in C$. A partial order $\trianglelefteq$ on C induces a binary relation $\triangleleft$ called *strict* partial order for which $a \triangleleft b$ is true if and only if $a \trianglelefteq b$ and $a \neq b$. A set C together with a partial order $\trianglelefteq$ is called a *partial ordered set*, or *poset* for short. A partial order $\trianglelefteq$ on C is called a *total order*, if for each pair $a, b \in C$, either $a \trianglelefteq b$ or $b \trianglelefteq a$.

For a subset S of a poset C, any element $u \in C$ that satisfies $s \trianglelefteq u$ for all $s \in S$ is called an *upperbound* of S. Furthermore, if u^* is an upperbound of $S \subseteq C$ with the property that $u^* \trianglelefteq u$ for all upperbounds u of S, then u^* is called the *join* (or *supremum*) of S. If $(C, \trianglelefteq)$ is a poset, then the upperbound and join of S in $(C, \trianglerighteq)$ are called *lowerbound* and *meet* (or *infimum*) in $(C, \trianglelefteq)$, respectively. The meet and join of a set $S \subseteq C$ are unique when they exist. A poset C for which every pair $a, b \in C$ has a unique meet $a \vee b$ and a unique join $a \wedge b$ is called a *lattice*.

A total order is called a *well-order* if every nonempty subset contains a lowerbound of itself.

2.3 Computational Complexity

We will only give an informal overview of some basic notions here. For a formal introduction, consider the book [AB09].

In the context of this thesis, an algorithm will be treated as a finite list of *elementary instructions* which work on some input data and produce some output data. We assume that each elementary instruction takes the same constant time to be carried out by a computer and that arithmetic operations on rational numbers can be treated as elementary instructions.

We use the term *problem* to describe an assignment of values to a subset of possible inputs to an algorithm. This way, a problem Q defines a partial function f_Q on the sets of all inputs. The inputs for which f_Q is well-defined are called *valid* (for Q). An algorithm A is *correct* (for Q), or rather *solves* Q, if it returns the value of f_Q for all valid inputs, that is, if A computes f_Q. The problem Q is called a *decision-problem* if f_Q is $0/1$ valued, where we can interpret 0 as *no* and 1 as *yes*.

The input and output of the algorithm consists of structural and numerical data and we consider the number of numbers in the input as its *input size*. For a problem Q, we can assign a runtime function $t_A : \mathbb{N} \to \mathbb{N}$ to every algorithm A by defining $t_A(n)$

as the maximal number of elementary instructions that A uses on any valid input of input size n.

The function t_A is well-defined if it only attains finite values, that is, if A *terminates* for any valid input of Q. Note that t_A only predicts the performance in the worst case and may not be indicative for the average case.

For two functions $f : \mathbb{N} \to \mathbb{N}$ and $g : \mathbb{N} \to \mathbb{N}$, the *big O notation* is defined as the relation

$$f = O(g) \Leftrightarrow \exists c > 0\ \exists n_0 \in \mathbb{N}\ \forall n > n_0 : f(n) \leq c \cdot g(n).$$

In other words, $f = O(g)$ implies that f is not growing faster than g asymptotically.

A problem Q is said to be *efficiently solvable* if there is a correct algorithm A that solves Q with $t_A \in O(n^k)$ for some fixed $k \in \mathbb{N}$, and we call A efficient in this case. The *complexity-class* P contains all efficiently solvable decision problems. A decision problem Q belongs to the class NP if there is an algorithm A such that for every valid input x with $f_Q(x) = 1$, there is a *certificate* y_x with the property that A can efficiently verify $f_Q(x) = 1$ with input (x, y_x).

While P $\subseteq$ NP is known, deciding whether P = NP or P $\neq$ NP is one of the major open problems in theoretical computer science. For this thesis, we will assume the following.

Conjecture 2.1:
Finding a solution is harder than verifying it, or, in other words, P≠NP.

A problem Q is called NP-*hard* when an efficient algorithm for Q can be converted into an efficient algorithm for every problem that belongs to the class NP. In particular, if any NP-hard problem is efficiently solvable, then P=NP.

2.4 Convex Analysis

A comprehensive treatment of convex analysis can be found in [Roc09].

A set $C \subseteq \mathbb{R}^d$ is called *convex* if for any pair $\boldsymbol{x}, \boldsymbol{y} \in C$, their connecting line segment

$$\{\mu \cdot \boldsymbol{x} + (1 - \mu) \cdot \boldsymbol{y} \mid \mu \in [0, 1]\}$$

is contained in C as well. Given any set $C \subseteq \mathbb{R}^d$, its *convex hull* conv(C) is the smallest convex set that contains C (with regards to set inclusion). A *hyperplane* $H \subseteq \mathbb{R}^d$ is a set of the form $\left\{\boldsymbol{x} \in \mathbb{R}^d \;\middle|\; \langle \boldsymbol{x}, \boldsymbol{a} \rangle = b\right\}$ for $\boldsymbol{a} \in \mathbb{R}^d$ and $b \in \mathbb{R}_+$. The two associated closed *halfspaces* are $H_- := \left\{\boldsymbol{x} \in \mathbb{R}^d \;\middle|\; \langle \boldsymbol{x}, \boldsymbol{a} \rangle \leq b\right\}$ and $H_+ := \left\{\boldsymbol{x} \in \mathbb{R}^d \;\middle|\; \langle \boldsymbol{x}, \boldsymbol{a} \rangle \geq b\right\}$. A hyperplane *supports* a convex set C if $H \cap C \neq \emptyset$ and either $C \subseteq H_+$ or $C \subseteq H_-$, and the intersection $H \cap C$ is called a *face* of C.

A real valued function $f : C \to \mathbb{R}$ defined on a convex set $C \subseteq \mathbb{R}^n$ is called *convex* if for any pair $\boldsymbol{x}, \boldsymbol{y} \in C$, its graph is below their connecting line segment, or, more

formally, if

$$f(\mu \cdot \boldsymbol{x} + (1-\mu) \cdot \boldsymbol{y}) \leq \mu \cdot f(\boldsymbol{x}) + (1-\mu) \cdot f(\boldsymbol{y})$$

for all choices of $\mu \in [0,1]$. For a convex function $f : C \to \mathbb{R}$, its *subdifferential* $\partial f(\boldsymbol{x_0})$ at a given point $\boldsymbol{x_0}$ parametrizes the supporting hyperplanes of the epigraph of f that pass through $(\boldsymbol{x_0}, f(\boldsymbol{x_0}))$. More formally, we have

$$\partial f(\boldsymbol{x}_0) := \{\boldsymbol{y} \in \mathbb{R}^n \mid f(\boldsymbol{x}) - f(\boldsymbol{x}_0) \geq \langle \boldsymbol{y}, \boldsymbol{x} - \boldsymbol{x}_0 \rangle \ \forall \boldsymbol{x} \in C\}\,.$$

The elements of the subdifferential are called *subgradients* and generalize the gradient for differentiable functions.

For a closed convex set $C \subseteq \mathbb{R}^d$ and a point $\boldsymbol{x} \in \mathbb{R}^d$, we can define their *Euclidean distance* as

$$\operatorname{dist}(\boldsymbol{x}, C) := \min\{\|\boldsymbol{x} - \boldsymbol{y}\|_2 \mid \boldsymbol{y} \in C\}\,,$$

which is a convex function in $\boldsymbol{x}$. The minimizer of $\operatorname{dist}(\boldsymbol{x}, C)$ is unique and called the *orthogonal projection* $\pi_C(\boldsymbol{x})$.

Lemma 2.2:
Let $C \subseteq \mathbb{R}^d$ be closed and convex, then

$$\frac{\boldsymbol{x} - \pi_C(\boldsymbol{x})}{\operatorname{dist}(\boldsymbol{x}, C)} \in \partial \operatorname{dist}(\boldsymbol{x}, C).$$

Convex Cones

A set $\mathcal{K} \subseteq \mathbb{R}^d$ is called a *cone* if it is invariant under positive rescaling, or, more formally, if $\mathbb{R}_+ \cdot \mathcal{K} \subseteq \mathcal{K}$. A cone is convex when it is closed under addition as well, so when $\mathcal{K} + \mathcal{K} \subseteq \mathcal{K}$. A convex cone $\mathcal{K}$ is called *pointed* if it has the property $\mathcal{K} \cap -\mathcal{K} = \{\mathbf{0}\}$ and *proper* if it is pointed, closed and has nonempty interior. The *dual cone* $\mathcal{K}^*$ of a convex cone $\mathcal{K}$ is defined as

$$\mathcal{K}^* := \{\boldsymbol{y} \in \mathbb{R}^n \mid \langle \boldsymbol{x}, \boldsymbol{y} \rangle \geq 0 \ \forall \boldsymbol{x} \in \mathcal{K}\}\,,$$

and a cone $\mathcal{K}$ for which $\mathcal{K} = \mathcal{K}^*$ is called *self-dual*.

A proper cone $\mathcal{K} \subseteq \mathbb{R}^d$ induces a corresponding *conic (partial) order* $\leq_{\mathcal{K}}$ on $\mathbb{R}^d$ by setting

$$\boldsymbol{y} \leq_{\mathcal{K}} \boldsymbol{x} \quad \Leftrightarrow \quad \boldsymbol{x} - \boldsymbol{y} \in \mathcal{K}.$$

In particular, membership $\boldsymbol{x} \in \mathcal{K}$ is equivalent to stating $\boldsymbol{x} \geq_{\mathcal{K}} \mathbf{0}$.

Example 2.3:
The set $\mathbb{R}^n$ can be considered as a convex cone with dual cone $(\mathbb{R}^n)^* = \{\mathbf{0}\}$, but is not pointed and thus not proper. Well-known proper, self-dual cones and their conic orders

include the *non-negative orthant* $(\mathbb{R}^n_+, \leq)$, the cone of symmetrical positive semidefinite $n \times n$ matrices $(\mathcal{S}^n_+, \leq)$ denoted below and the *Lorenz-cone* $(\mathcal{L}_n, \leq_{\mathcal{L}})$ defined as

$$\mathcal{L}_n = \{(x_0, \tilde{\boldsymbol{x}}) \in \mathbb{R}_+ \times \mathbb{R}^n \mid \|\tilde{\boldsymbol{x}}\|_2 \leq x_0\} .$$

Furthermore, the poset $(\mathbb{R}^n_+, \leq)$ is a lattice.

Positives Semidefinite Matrices

A quadratic, symmetrical matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is called *positive semidefinite*, or psd. for short, if the corresponding quadratic form $\boldsymbol{x}^\top \mathbf{A} \boldsymbol{x}$ is non-negative for all $\boldsymbol{x} \in \mathbb{R}^n$. Equivalently, there exists a *Cholesky decomposition* of $\mathbf{A}$, which means there is a matrix $\mathbf{V} \in \mathbb{R}^{n \times \operatorname{rank}(\mathbf{A})}$ such that $\mathbf{A} = \mathbf{V}\mathbf{V}^\top$. A direct consequence is the following lemma.

Lemma 2.4:
Let $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$ *and* $\mathbf{C} = \mathbf{B}\mathbf{A}\mathbf{B}^\top$ *where* $\mathbf{B}$ *is invertible. Then* $\mathbf{A} \succeq \mathbf{0} \iff \mathbf{C} \succeq \mathbf{0}$.

Furthermore, every psd. matrix $\mathbf{A}$ has a unique psd. square root $\mathbf{A}^{\frac{1}{2}}$ that satisfies $(\mathbf{A}^{\frac{1}{2}})^2 = \mathbf{A}$. The *Schur complement* of a psd. matrix

$$\mathbf{M} = \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^\top & \mathbf{C} \end{pmatrix} \succeq \mathbf{0} \tag{2.3}$$

with invertible $\mathbf{A}$ is defined as $\mathbf{M}/\mathbf{A} = \mathbf{C} - \mathbf{B}^\top \mathbf{A}^{-1} \mathbf{B}$. The following is a central argument.

Lemma 2.5 (Schur complement Lemma)**:**
Let $\mathbf{M}$ *be as in* (2.3). *Then* $\mathbf{M} \succeq \mathbf{0}$ *if and only if* $\mathbf{M}/\mathbf{A} \succeq \mathbf{0}$. *Furthermore,* $\mathbf{M}$ *is invertible if and only if* $\mathbf{M}/\mathbf{A}$ *is invertible.*

Polytopes

The following is a minimal introduction to polytopes, and we refer the reader to [Zie95] for a more elaborate treatment of the topic.

A convex set $P \subseteq \mathbb{R}^d$ that can be written as the convex hull of a finite set of points $\mathbf{V} = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m)$ is called a *(convex) polytope* and has the so-called *inner description*

$$P = \left\{ \boldsymbol{x} \in \mathbb{R}^d \;\middle|\; \boldsymbol{x} = \mathbf{V}\boldsymbol{\lambda},\ \boldsymbol{\lambda} \geq \mathbf{0},\ \langle \boldsymbol{\lambda}, \boldsymbol{e} \rangle = 1 \right\} .$$

Example 2.6:
The set

$$\Delta^d := \left\{ \boldsymbol{\lambda} \in \mathbb{R}^d_+ \;\middle|\; \langle \boldsymbol{\lambda}, \boldsymbol{e} \rangle = 1 \right\} = \operatorname{conv}(\{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_d\})$$

used to parametrize a polytope in its inner description is called *(standard) simplex* and is itself a polytope.

Alternatively, each polytope can be described by a finite intersection of halfspaces, which leads to its so-called *outer description*

$$P = \left\{ \boldsymbol{x} \in \mathbb{R}^d \;\middle|\; \mathbf{A}\boldsymbol{x} \leq \boldsymbol{b} \right\}.$$

The faces of a polytope $P \subseteq \mathbb{R}^d$ are polytopes themselves, and its zero-dimensional faces are called its *vertices*, denoted by $\operatorname{vert}(P)$. For any finite set $S \subseteq \mathbb{R}^d$, it holds that $\operatorname{vert}(\operatorname{conv}(S)) \subseteq S$, and $\operatorname{conv}(\operatorname{vert}(P)) = P$ holds for all polytopes $P \subseteq \mathbb{R}^d$.

Example 2.7:
The k-truncated d-dimensional *unit-cube* C_k^d has the outer description

$$C_k^d = \left\{ \boldsymbol{x} \in \mathbb{R}^d \;\middle|\; \langle \boldsymbol{x}, \boldsymbol{e} \rangle \leq k,\ 0 \leq x_i \leq 1 \ \forall i \in [d] \right\}$$

and its inner description can be constructed from its vertices

$$\operatorname{vert}(C_k^d) = \left\{ \boldsymbol{x} \in \{0,1\}^d \;\middle|\; \langle \boldsymbol{x}, \boldsymbol{e} \rangle \leq k \right\}.$$

The shorthand $C^d := C_d^d$ is used for the d-dimensional unit-cube itself.

One major result about polytopes is the following theorem.

Theorem 2.8 (Carathéodory's theorem, [Die05, Prop. 1.15])**:**
Let $V \subseteq \mathbb{R}^d$ be a finite set of points and consider a point $\boldsymbol{x} \in \mathbb{R}^d$ in the polytope $\operatorname{conv}(V)$*. Then $\boldsymbol{x} \in \operatorname{conv}(V')$ for a subset $V' \subseteq V$ of size at most $|V'| \leq \dim(\operatorname{conv}(V)) + 1$.*

2.5 Algebra

For a good introduction to computer algebra and the necessary foundations, we recommend the book [CLO15].

Polynomials, Ideals, Varieties

Given vectors $\boldsymbol{\alpha} \in \mathbb{N}^d$ and $\boldsymbol{x} \in \mathbb{R}^d$, the product $\boldsymbol{x}^{\boldsymbol{\alpha}} = \prod_{i \in [d]} x_i^{\alpha_i}$ is called a *monomial* whose *total degree* is defined as $\deg(\boldsymbol{x}^{\boldsymbol{\alpha}}) := \langle \boldsymbol{e}, \boldsymbol{\alpha} \rangle$. Let $\mathbb{R}[\boldsymbol{x}]$ denote the set of multivariate polynomials in $\boldsymbol{x} = (x_1, \ldots, x_d)$, where we extend

$$\deg(f) := \max \{ \deg(\boldsymbol{x}^{\boldsymbol{\alpha}}) \mid f_{\boldsymbol{\alpha}} \neq 0 \}$$

for any element $f(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha}} f_{\boldsymbol{\alpha}} \boldsymbol{x}^{\boldsymbol{\alpha}}$ of $\mathbb{R}[\boldsymbol{x}]$.

The vector space of multivariate polynomials of degree at most t is

$$\mathbb{R}_t[\boldsymbol{x}] := \{ f \in \mathbb{R}[\boldsymbol{x}] \mid \deg(f) \leq t \},$$

where $z_d(t) := \dim(\mathbb{R}_t[\boldsymbol{x}]) = \binom{d+t}{d}$. By defining

$$\mathbb{N}^d_t := \left\{ \boldsymbol{\alpha} \in \mathbb{N}^d \;\middle|\; \langle \boldsymbol{\alpha}, \boldsymbol{e} \rangle \leq t \right\},$$

we see that each polynomial $f \in \mathbb{R}_t[\boldsymbol{x}]$ can be written as $f(\boldsymbol{x}) = \sum_{\alpha \in \mathbb{N}^d_t} f_\alpha \boldsymbol{x}^{\boldsymbol{\alpha}}$ and we may identify $\mathbb{R}_t[\boldsymbol{x}]$ with $\mathbb{R}^{\mathbb{N}^d_t} \cong \mathbb{R}^{z_d(t)}$ by identifying the polynomial f with its vector of coefficients $\boldsymbol{f}$. In this context, we will also write $\boldsymbol{f} \in \mathbb{R}^{z_d(t)}$ and denote the canonical monomial base $(\boldsymbol{x}^{\boldsymbol{\alpha}})_{\boldsymbol{\alpha} \in \mathbb{N}^d_t}$ through the *Veronese map* $v_t(\boldsymbol{x})$, such that

$$f(\boldsymbol{x}) = \langle \boldsymbol{f}, v_t(\boldsymbol{x}) \rangle.$$

Given $\boldsymbol{y} = (y_{\boldsymbol{\alpha}})_{\boldsymbol{\alpha} \in \mathbb{N}^d} \in \mathbb{R}^{\mathbb{N}^d}$, we can use this identification to define the *Riesz functional* $L_{\boldsymbol{y}} : \mathbb{R}[\boldsymbol{x}] \to \mathbb{R}$ as

$$f \mapsto L_{\boldsymbol{y}}(f) = \langle \boldsymbol{f}, \boldsymbol{y} \rangle,$$

which can be understood as the linearization of a polynomial.

A subset $\mathcal{I} \subseteq \mathbb{R}[\boldsymbol{x}]$ is called an *ideal* if it is closed under addition ($\mathcal{I} + \mathcal{I} \subseteq \mathcal{I}$) and under multiplication of the whole multivariate polynomial ring ($\mathbb{R}[\boldsymbol{x}] \cdot \mathcal{I} \subseteq \mathcal{I}$). Any set of polynomials $\mathcal{H} \subseteq \mathbb{R}[\boldsymbol{x}]$ generates an ideal $\langle \mathcal{H} \rangle := \sum_{h \in \mathcal{H}} \mathbb{R}[\boldsymbol{x}] \cdot h$. An ideal $\mathcal{I}$ is called *finitely generated* if it can be generated by a finite set $\mathcal{H} \subseteq \mathcal{I}$, in which case $\mathcal{H}$ is called a *basis* of $\mathcal{I}$. By *Hilbert's basis theorem* [CLO15, Ch. 2, §5, Thm. 4], every ideal $\mathcal{I} \subseteq \mathbb{R}[\boldsymbol{x}]$ is finitely generated. The set

$$\mathcal{V}_{\mathbb{R}}(\mathcal{I}) := \left\{ \boldsymbol{x} \in \mathbb{R}^d \;\middle|\; h(\boldsymbol{x}) = 0 \; \forall h \in \mathcal{I} \right\}$$

is the set of common real zeros of all polynomials in $\mathcal{I}$ and called the *real variety* of $\mathcal{I}$. Given a finite subset $\mathcal{G} \subseteq \mathbb{R}[\boldsymbol{x}]$, the set

$$\left\{ \boldsymbol{x} \in \mathbb{R}^d \;\middle|\; g(\boldsymbol{x}) \geq 0 \; \forall g \in \mathcal{G} \right\}$$

is called *(closed) basic semialgebraic*. In particular, real varieties are basic semialgebraic.

Gröbner bases

A *monomial order* $\lhd$ is a strict well-order on $\mathbb{N}^d$ that is translation invariant, or more formally, which implies for all choices of $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma} \in \mathbb{N}^d$ that $\boldsymbol{\alpha} + \boldsymbol{\gamma} \lhd \boldsymbol{\beta} + \boldsymbol{\gamma}$ follows from $\boldsymbol{\alpha} \lhd \boldsymbol{\beta}$. Given a monomial order $\lhd$, we also write $\boldsymbol{\alpha} \trianglelefteq \boldsymbol{\beta}$ if either $\boldsymbol{\alpha} \lhd \boldsymbol{\beta}$ or $\boldsymbol{\alpha} = \boldsymbol{\beta}$.

Example 2.9:
The *lexicographic order* $<_{lex}$ on $\mathbb{R}^d$ is defined as

$$\boldsymbol{\alpha} <_{lex} \boldsymbol{\beta} \;\Leftrightarrow\; \alpha_m < \beta_m \text{ for } m = \min\{i \in [d] \mid \alpha_i \neq \beta_i\}$$

and the *graded lexicographic order* $<_{grlex}$ on $\mathbb{R}^d$ is defined as

$$\boldsymbol{\alpha} <_{grlex} \boldsymbol{\beta} \;\Leftrightarrow\; \langle \boldsymbol{\alpha}, \boldsymbol{e} \rangle < \langle \boldsymbol{\beta}, \boldsymbol{e} \rangle \text{ or } \langle \boldsymbol{\alpha}, \boldsymbol{e} \rangle = \langle \boldsymbol{\beta}, \boldsymbol{e} \rangle, \; \boldsymbol{\alpha} <_{lex} \boldsymbol{\beta}.$$

Both of these orders are monomial orders.

Let $p = \sum_{\boldsymbol{\alpha}} p_{\boldsymbol{\alpha}} \boldsymbol{x}^{\boldsymbol{\alpha}} \in \mathbb{R}[\boldsymbol{x}] \setminus \{0\}$ and fix a monomial order $\lhd$ to define its *leading term* $LT_{\lhd}(p) = p_{\boldsymbol{\alpha}_*} \boldsymbol{x}^{\boldsymbol{\alpha}_*}$, where $\boldsymbol{\alpha}_*$ is the maximal $\boldsymbol{\alpha}$ w.r.t. $\lhd$ such that $p_{\boldsymbol{\alpha}} \neq 0$. More generally, the leading terms $LT_{\lhd}(\mathcal{I})$ of an ideal $\mathcal{I} \neq \{0\}$ are defined as

$$LT_{\lhd}(\mathcal{I}) := \{LT_{\lhd}(p) \mid p \in \mathcal{I} \setminus \{0\}\}.$$

A *Gröbner basis* (or standard basis) of an ideal $\mathcal{I}$ is any basis $\mathcal{G} = \{g_1, \ldots, g_m\}$ of $\mathcal{I}$ with the property that $LT_{\lhd}(\mathcal{I}) = \langle LT_{\lhd}(g_1), \ldots, LT_{\lhd}(g_m)\rangle$. Additionally, $\mathcal{G}$ is called *reduced* if, for each $i \in [m]$, $LT_{\lhd}(g_i)$ is a monomial and no monomial appearing in g_i lies in $\langle LT_{\lhd}(\mathcal{G} \setminus \{g_i\})\rangle$. For every ideal $\{0\} \neq \mathcal{I} \subseteq \mathbb{R}[\boldsymbol{x}]$ and every monomial order $\lhd$, there exists a unique reduced Gröbner basis.

Quotients of Polynomial Rings

Any ideal $\mathcal{I} \subseteq \mathbb{R}[\boldsymbol{x}]$ defines an equivalence relation on $\mathbb{R}[\boldsymbol{x}]$ by setting

$$g \equiv f \mod \mathcal{I} \iff g - f \in \mathcal{I}.$$

The set of its equivalence classes $\mathbb{R}[\boldsymbol{x}]/\mathcal{I} = \{[g] \mid g \in \mathbb{R}[\boldsymbol{x}]\}$ is called the *quotient* of $\mathbb{R}[\boldsymbol{x}]$ modulo $\mathcal{I}$, and $[g] = \{f \in \mathbb{R}[\boldsymbol{x}] \mid g \equiv f \mod \mathcal{I}\} = g + \mathcal{I}$ is a ring itself with the operations $[f] + [g] = [f + g]$ and $[f] \cdot [g] = [f \cdot g]$.

Theorem 2.10 ([CLO15, Ch. 5, §3, Prop. 4])**:**
Let $\mathcal{I} \subseteq \mathbb{R}[\boldsymbol{x}]$ be an ideal. Then

$$\mathbb{R}[\boldsymbol{x}]/\mathcal{I} \cong \operatorname{lin}(\{\boldsymbol{x}^{\boldsymbol{\alpha}} \mid \boldsymbol{x}^{\boldsymbol{\alpha}} \notin LT_{\lhd}(\mathcal{I})\}) \tag{2.4}$$

as $\mathbb{R}$-vector spaces with isomorphism

$$[f] \mapsto \overline{f}.$$

In particular, we can write $[f] = \overline{f} + \mathcal{I}$ for all $f \in \mathbb{R}[\boldsymbol{x}]$.

Remark 2.11:
Given a reduced Gröbner basis for $\mathcal{I}$ and $\lhd$, the computation of the reduction map $[f] \mapsto \overline{f}$ from Theorem (2.10) can be carried out efficiently algorithmically, as outlined in [CLO15]. In particular, addition $[f] + [g]$ and multiplication $[f] \cdot [g]$ translate into $\overline{f} + \overline{g}$ and $\overline{\overline{f} \cdot \overline{g}}$ respectively, where the latter is computationally much more involved than the former.

As a generalization of the degree constrained vector space $\mathbb{R}_t[\boldsymbol{x}]$, we also define

$$\mathbb{R}_t[\boldsymbol{x}]/\mathcal{I} \cong \operatorname{lin}(\{\boldsymbol{x}^{\boldsymbol{\alpha}} \mid \boldsymbol{x}^{\boldsymbol{\alpha}} \notin LT_{\lhd}(\mathcal{I}),\ \langle \boldsymbol{\alpha}, \boldsymbol{e}\rangle \leq t\}).$$

We close this section with examples for various ideals that will be encountered in the rest of the thesis.

Example 2.12 (Binary Vectors)**:**
Let

$$\mathcal{I}(C^d) = \langle x_1^2 - x_1, \ldots, x_d^2 - x_d \rangle$$

be the ideal whose variety are the vertices of the d-dimensional unit cube C^d, e.g. the set of d-dimensional binary vectors. A reduced Gröbner basis for $\mathcal{I}(C^d)$ and $<_{grlex}$ is given by $\left\{x_i^2 \;\middle|\; i \in [d]\right\}$, and by Theorem 2.10, we get the $\mathbb{R}$-vector space isomorphisms

$$\mathbb{R}[x]/\mathcal{I}(C^d) \cong \operatorname{lin}\left(\left\{\boldsymbol{x}^{\boldsymbol{\alpha}} \;\middle|\; \boldsymbol{\alpha} \in \{0,1\}^d\right\}\right) \cong \operatorname{lin}\left(\left\{x_I \;\middle|\; I \in 2^{[d]}\right\}\right)$$

by using $(\{0,1\}^d, \vee) \cong (2^{[d]}, \cup)$ via $\boldsymbol{\alpha} \leftrightarrow \operatorname{supp}(\boldsymbol{\alpha})$. Then multiplication in $\mathbb{R}[x]/\mathcal{I}(C^d)$ translates into the multiplication $x_I \cdot x_J = x_{I \cup J}$ for all $I, J \subseteq [d]$.

Example 2.13 (Independence Systems)**:**
For an independence system $L \subseteq 2^{[d]}$, let

$$\mathcal{I}(L) := \left\langle \boldsymbol{x}^{\boldsymbol{\alpha}} \;\middle|\; \boldsymbol{\alpha} \in \mathbb{N}^d,\ \operatorname{supp}(\boldsymbol{\alpha}) \notin L \right\rangle$$

be its Stanley-Reisner ideal, whose variety is

$$\mathcal{V}_{\mathbb{R}}(L) := \mathcal{V}_{\mathbb{R}}(\mathcal{I}(L)) = \left\{\boldsymbol{x} \in \mathbb{R}^d \;\middle|\; \operatorname{supp}(\boldsymbol{x}) \in L\right\},$$

the union of linear subspaces of $\mathbb{R}^d$ whose support is in L. A reduced Gröbner basis for $\mathcal{I}(L)$ and $<_{grlex}$ is given by

$$\left\{\boldsymbol{x}^{\boldsymbol{\alpha}} \;\middle|\; \boldsymbol{\alpha} \in \{0,1\}^d,\ \operatorname{supp}(\boldsymbol{\alpha}) \text{ is minimally dependent in } L\right\}.$$

By Theorem 2.10, we get the $\mathbb{R}$-vector space isomorphism

$$\mathbb{R}[x]/\mathcal{I}(L) \cong \operatorname{lin}\left(\left\{\boldsymbol{x}^{\boldsymbol{\alpha}} \;\middle|\; \boldsymbol{\alpha} \in \mathbb{N}^d,\ \operatorname{supp}(\boldsymbol{\alpha}) \in L\right\}\right)$$

and multiplication in $\mathbb{R}[x]/\mathcal{I}(L)$ becomes

$$\boldsymbol{x}^{\boldsymbol{\alpha}} \cdot \boldsymbol{x}^{\boldsymbol{\beta}} = \begin{cases} \boldsymbol{x}^{\boldsymbol{\alpha}+\boldsymbol{\beta}} & \text{if } \operatorname{supp}(\boldsymbol{\alpha}+\boldsymbol{\beta}) \in L, \\ 0 & \text{else.} \end{cases}$$

The sum of both ideals then yields the intersection of the corresponding varieties, the L-constrained cube given by

$$\mathcal{V}_{\mathbb{R}}\left(\mathcal{I}(C^d) + \mathcal{I}(L)\right) = \{0,1\}^d \cap \mathcal{V}_{\mathbb{R}}(L) = \left\{\boldsymbol{x} \in \{0,1\}^d \;\middle|\; \operatorname{supp}(\boldsymbol{x}) \in L\right\},$$

and it can be shown that

$$\mathbb{R}[x]/\mathcal{I}(L) \cong \operatorname{lin}\left(\left\{\boldsymbol{x}^{\boldsymbol{\alpha}} \;\middle|\; \boldsymbol{\alpha} \in \{0,1\}^d \cap \mathcal{V}_{\mathbb{R}}(L)\right\}\right) \cong \operatorname{lin}\left(\{x_I \mid I \in L\}\right),$$

with multiplication

$$x_I \cdot x_J = \begin{cases} x_{I \cup J} & \text{if } I \cup J \in L, \\ 0 & \text{else.} \end{cases}$$

Example 2.14 (Affine Hyperplane)**:**
Let $h = 1 - \langle \boldsymbol{x}, \boldsymbol{e} \rangle$ be the linear polynomial whose variety is the hyperplane in $\mathbb{R}^d$ containing Δ^d and let $\mathcal{I}(\Delta^d) = \langle 1 - \langle \boldsymbol{x}, \boldsymbol{e} \rangle \rangle \subseteq \mathbb{R}[\boldsymbol{x}]$ be the corresponding ideal. For the monomial order $<_{grlex}$, a reduced Gröbner basis is h itself, and since $LT_{grlex}(h) = x_1$, it follows that $\mathbb{R}[\boldsymbol{x}]/\mathcal{I}(\Delta^d) \cong \mathbb{R}[x_2, \ldots, x_d]$, which can be realized by just eliminating all occurrences of x_1 via $x_1 = 1 - x_2 - \ldots - x_d$.

While $\mathbb{R}[\boldsymbol{x}]/\mathcal{I}(\Delta^d) \cong \mathbb{R}[x_2, \ldots, x_d]$ is intuitive, it creates an artificial asymmetry between the otherwise symmetrical variables in $\boldsymbol{x}$. The following result can be used as symmetrical alternative instead.

Theorem 2.15:
We have $\mathbb{R}_t[\boldsymbol{x}]/\mathcal{I}(\Delta^d) \cong \mathbb{R}_t^h[\boldsymbol{x}] := \{f \in \mathbb{R}[\boldsymbol{x}] \mid \deg(f) = t\}$ *via the homogenization map*

$$f(\boldsymbol{x}) \mapsto f^{h_t}(\boldsymbol{x}) := \langle \boldsymbol{x}, \boldsymbol{e} \rangle^t \cdot f\left(\tfrac{\boldsymbol{x}}{\langle \boldsymbol{x}, \boldsymbol{e} \rangle}\right). \tag{2.5}$$

Proof. By definition, $f^{h_t} \equiv f \mod \mathcal{I}(\Delta^d)$ for any $f \in \mathbb{R}_t[\boldsymbol{x}]$, so the map is well-defined, and all monomials occurring in f^{h_t} have degree t, so the image of $\mathbb{R}_t[\boldsymbol{x}]/\mathcal{I}(\Delta^d)$ is contained in $\mathbb{R}_t^h[\boldsymbol{x}]$. It suffices now to show that both vector spaces have the same dimension, and by Example 2.14,

$$\dim(\mathbb{R}_t[\boldsymbol{x}]/\mathcal{I}(\Delta^d)) = \dim(\mathbb{R}_t[x_2, \ldots, x_d]) = z_{d-1}(t) = \binom{d-1+t}{t} = \dim(\mathbb{R}_t^h[\boldsymbol{x}]),$$

where the last equation follows from [Sta11, p. 18]. □

2.6 Method of Moments

This section gives a basic description of *method of moments* (MM), and is based mostly on the book [Las15], with some minor changes of notation.

Moments

Moment matrices

For $t \in \mathbb{N}$ and $\boldsymbol{y} \in \mathbb{R}^{\mathbb{N}^d_{2t}}$, the matrix $\mathbf{M}_t(\boldsymbol{y})$ indexed by $\mathbb{N}^d_t$ is defined entrywise by

$$(\mathbf{M}_t(\boldsymbol{y}))_{\boldsymbol{\alpha},\boldsymbol{\beta}} := L_{\boldsymbol{y}}(\boldsymbol{x}^{\boldsymbol{\alpha}} \cdot \boldsymbol{x}^{\boldsymbol{\beta}}) = y_{\boldsymbol{\alpha}+\boldsymbol{\beta}} \quad \forall \boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{N}^d_t$$

and is called the *moment matrix* of order t of $\boldsymbol{y}$.

More generally, let $f \in \mathbb{R}[\boldsymbol{x}]$ be a multivariate polynomial and define

$$\mathrm{td}(f) := \left\lceil \frac{\deg(f)}{2} \right\rceil.$$

Then for $t \geq \operatorname{td}(f)$, the matrix $\mathbf{M}_t(f, \boldsymbol{y})$ indexed by $\mathbb{N}^d_{t-\operatorname{td}(f)}$ and entrywise defined as

$$(\mathbf{M}_t(f, \boldsymbol{y}))_{\boldsymbol{\alpha},\boldsymbol{\beta}} := L_{\boldsymbol{y}}(\boldsymbol{x}^{\boldsymbol{\alpha}} \cdot \boldsymbol{x}^{\boldsymbol{\beta}} \cdot f)$$

is called the *localizing moment matrix* of order t of $\boldsymbol{y}$ with respect to f. Note that each entry of $\mathbf{M}_t(f, \boldsymbol{y})$ is a linear expression in $\boldsymbol{y}$ and that we recover $\mathbf{M}_t(\boldsymbol{y}) = \mathbf{M}_t(1, \boldsymbol{y})$ as a special case.

Example 2.16:
Consider the bivariate polynomial ring $\mathbb{R}[\boldsymbol{x}]$ with $\boldsymbol{x} = (x_1, x_2)$ and the polynomials $h_i(\boldsymbol{x}) = x_i^2 - x_i$ for $i = 1, 2$. Then, for $t = 2$ we have

$$v_2(\boldsymbol{x})^\top = (1, x_1, x_2, x_1^2, x_1x_2, x_2^2)$$

and consequently

$$v_2(\boldsymbol{x})v_2(\boldsymbol{x})^\top = \left(\begin{array}{c|cc|ccc} 1 & x_1 & x_2 & x_1^2 & x_1x_2 & x_2^2 \\ \hline x_1 & x_1^2 & x_1x_2 & x_1^3 & x_1^2x_2 & x_1x_2^2 \\ x_2 & x_1x_2 & x_2^2 & x_1^2x_2 & x_1x_2^2 & x_2^3 \\ \hline x_1^2 & x_1^3 & x_1^2x_2 & x_1^4 & x_1^3x_2 & x_1^2x_2^2 \\ x_1x_2 & x_1^2x_2 & x_1x_2^2 & x_1^3x_2 & x_1^2x_2^2 & x_1x_2^3 \\ x_2^2 & x_1x_2^2 & x_2^3 & x_1^2x_2^2 & x_1x_2^3 & x_2^4 \end{array}\right).$$

Applying $L_{\boldsymbol{y}}$ componentwise on this matrix then shows

$$\mathbf{M}_2(\boldsymbol{y}) = \left(\begin{array}{c|cc|ccc} y_{(0,0)} & y_{(1,0)} & y_{(0,1)} & y_{(2,0)} & y_{(1,1)} & y_{(0,2)} \\ \hline y_{(1,0)} & y_{(2,0)} & y_{(1,1)} & y_{(3,0)} & y_{(2,1)} & y_{(1,2)} \\ y_{(0,1)} & y_{(1,1)} & y_{(0,2)} & y_{(2,1)} & y_{(1,2)} & y_{(0,3)} \\ \hline y_{(2,0)} & y_{(3,0)} & y_{(2,1)} & y_{(4,0)} & y_{(3,1)} & y_{(2,2)} \\ y_{(1,1)} & y_{(2,1)} & y_{(1,2)} & y_{(3,1)} & y_{(2,2)} & y_{(1,3)} \\ y_{(0,2)} & y_{(1,2)} & y_{(0,3)} & y_{(2,2)} & y_{(1,3)} & y_{(0,4)} \end{array}\right),$$

which has entries $\boldsymbol{y} \in \mathbb{R}^{\mathbb{N}^2_4} \cong \mathbb{R}^{15}$ and thus lies in a 15-dimensional subspace. Note that the blocks in both matrices contain entries corresponding to constant total degree. Since $\operatorname{td}(h_i) = 1$, we also get the componentwise maps

$$L_{\boldsymbol{y}} : h_i(\boldsymbol{x})v_1(\boldsymbol{x})v_1(\boldsymbol{x})^\top \mapsto \mathbf{M}_2(h_i, \boldsymbol{y})$$

which is explicitly given for the case of $i = 1$ as

$$h_1(\boldsymbol{x})v_1(\boldsymbol{x})v_1(\boldsymbol{x})^\top = \left(\begin{array}{c|cc} x_1^2 - x_1 & x_1^3 - x_1^2 & x_1^2x_2 - x_1x_2 \\ \hline x_1^3 - x_1^2 & x_1^4 - x_1^2 & x_1^3x_2 - x_1^2x_2 \\ x_1^2x_2 - x_1x_2 & x_1^3x_2 - x_1^2x_2 & x_1^2x_2^2 - x_1x_2^2 \end{array}\right)$$

and

$$\mathbf{M}_2(h_1, \boldsymbol{y}) = \left(\begin{array}{c|cc} y_{(2,0)} - y_{(1,0)} & y_{(3,0)} - y_{(2,0)} & y_{(2,1)} - y_{(1,1)} \\ \hline y_{(3,2)} - y_{(2,0)} & y_{(4,0)} - y_{(3,0)} & y_{(3,1)} - y_{(2,1)} \\ y_{(2,1)} - y_{(1,1)} & y_{(3,1)} - y_{(2,1)} & y_{(2,2)} - y_{(1,2)} \end{array}\right).$$

Measures and moments

Let $\mathcal{N}(C) \subseteq \mathbb{R}[\boldsymbol{x}]$ be the convex cone of polynomials that are nonnegative on $C \subseteq \mathbb{R}^d$ and denote its dual cone by

$$\mathcal{N}^*(C) := \left\{ \boldsymbol{y} \in \mathbb{R}^{\mathbb{N}^d} \;\middle|\; L_y(f) \geq 0 \;\forall f \in \mathcal{N}(C) \right\}.$$

Denote by $\mathcal{M}_+(C)$ the space of finite (nonnegative) Borel measures supported on C and by $\mathcal{P}(C)$ the subset of probability measures on C. We can recover the cone of the corresponding moments

$$\left\{ \boldsymbol{y} \in \mathbb{R}^{\mathbb{N}^d} \;\middle|\; \exists \mu \in \mathcal{M}_+(C) \colon y_\alpha = \int_C \boldsymbol{x}^{\boldsymbol{\alpha}} \cdot d\mu \quad \forall \boldsymbol{\alpha} \in \mathbb{N}^d \right\} \subseteq \mathcal{N}^*(C), \tag{2.6}$$

where equality holds if C is compact [Las15, Lemma 4.7].

The Method

Reformulation of Optimization Problems

Let $C \subseteq \mathbb{R}^d$ be a compact set and $f(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha} \in \mathbb{N}^d_t} f_{\boldsymbol{\alpha}} \boldsymbol{x}^{\boldsymbol{\alpha}}$ be a real-valued multivariate polynomial, then

$$\inf_{\boldsymbol{x} \in C} f(\boldsymbol{x}) = \inf_{\mu \in \mathcal{P}(C)} \int_C f \cdot d\mu \tag{2.7}$$

can be reduced to a convex linear programming problem. Indeed, we have that

$$\int_C f \cdot d\mu = \int_C \sum_{\boldsymbol{\alpha} \in \mathbb{N}^d_t} f_{\boldsymbol{\alpha}} \boldsymbol{x}^{\boldsymbol{\alpha}} \cdot d\mu = \sum_{\boldsymbol{\alpha} \in \mathbb{N}^d_t} f_{\boldsymbol{\alpha}} \int_C \boldsymbol{x}^{\boldsymbol{\alpha}} \cdot d\mu = L_y(f),$$

where

$$y_{\boldsymbol{\alpha}} = \int_C \boldsymbol{x}^{\boldsymbol{\alpha}} \cdot d\mu$$

is the moment of order $\boldsymbol{\alpha}$.

Consequently,

$$\inf \left\{ L_y(f) \;\middle|\; y_0 = 1,\ \boldsymbol{y} \in \mathcal{N}^*(C) \right\} \tag{2.8}$$

is a relaxation of problem (2.7) with the benefit of being a reformulation whenever equality holds in (2.6).

Note that the constraint $y_0 = 1$ normalizes $\boldsymbol{y} \in \mathcal{N}^*(C)$ to represent a probability measure in $\mathcal{P}(C) \subsetneq \mathcal{M}_+(C)$.

Although problem (2.8) is a convex linear programming problem, the characterization of $\boldsymbol{y} \in \mathcal{N}^*(C)$ (known as C-moment problem in the literature) may be notoriously hard for general C.

However, for compact *semi-algebraic* C given as

$$C = \left\{ \boldsymbol{x} \in \mathbb{R}^d \;\middle|\; g(\boldsymbol{x}) \geq 0 \quad \forall g \in \mathcal{G} \right\} \tag{2.9}$$

for some *finite* set of polynomials $\mathcal{G} \subseteq \mathbb{R}[\boldsymbol{x}]$, an explicit characterization of $\mathcal{N}^*(C)$ is available. Since C is assumed to be compact, we will assume without loss of generality that

$$0 \leq R^2 - \|\boldsymbol{x}\|^2 \in \mathcal{G},$$

where R is a sufficiently large positive constant.

Remark 2.17:
In fact, we would only need any function u in the quadratic module generated by $\mathcal{G}$ to have a compact superlevel set $\left\{ \boldsymbol{x} \in \mathbb{R}^d \;\middle|\; u(\boldsymbol{x}) \leq 0 \right\}$ for the following.

This representation allows the application of a theorem on positivity by Putinar [Las15, Theorem 2.15], which leads to

$$\begin{aligned} \mathcal{N}^*(C) &= \left\{ \boldsymbol{y} \in \mathbb{R}^{\mathbb{N}^d} \;\middle|\; \mathbf{M}_t(\boldsymbol{y}) \succeq \mathbf{0},\ \mathbf{M}_t(g, \boldsymbol{y}) \succeq \mathbf{0} \quad \forall g \in \mathcal{G},\ \forall t \in \mathbb{N} \right\} \\ &=: \mathcal{N}^*_{\succeq}(\mathcal{G}). \end{aligned}$$

In particular, problem (2.8) is equivalent to

$$\inf \left\{ L_{\boldsymbol{y}}(f) \;\middle|\; y_{\mathbf{0}} = 1,\ \boldsymbol{y} \in \mathcal{N}^*_{\succeq}(\mathcal{G}) \right\}.$$

To summarize, if f is a polynomial and C a compact semi-algebraic set, then problem (2.7) is equivalent to a convex linear programming problem with an infinite number of linear constraints on an infinite number of decision variables.

Semidefinite Relaxations

Now, for $t \geq \operatorname{td}(f)$, consider the finite-dimensional truncations

$$\rho_t = \inf \left\{ L_{\boldsymbol{y}}(f) \;\middle|\; y_{\mathbf{0}} = 1,\ \boldsymbol{y} \in \mathcal{N}^*_t(\mathcal{G}) \right\} \tag{2.10}$$

of problem (2.8) where

$$\mathcal{N}^*_t(\mathcal{G}) := \left\{ \boldsymbol{y} \in \mathbb{R}^{\mathbb{N}^d_{2t}} \;\middle|\; \begin{array}{rll} \mathbf{M}_t(\boldsymbol{y}) & \succeq \mathbf{0}, & \\ \mathbf{M}_t(g, \boldsymbol{y}) & \succeq \mathbf{0} & \forall g \in \mathcal{G} : t \geq \operatorname{td}(g) \end{array} \right\}.$$

By construction, $\{\mathcal{N}^*_t(\mathcal{G})\}_{t \in \mathbb{N}}$ generates a hierarchy of relaxations for $\mathcal{N}^*(C)$ in problem (2.8), where each $\mathcal{N}^*_t(\mathcal{G})$, is concerned with moment and localizing matrices of fixed size t. The lowerbounds ρ_t monotonically converge toward the optimal value of (2.7) [Las15, Theorem 6.2] and finite convergence may take place, which can be efficiently checked [Las15, Theorem 6.6].

Furthermore, in the best case of finite convergence, (2.10) will yield the global optimal value and a convex combination of global optimal solutions as minimizer, which can be efficiently decomposed into optimal solutions [Las15, Sct. 6.1.2].

In the noncompact case, the ρ_t are still monotonically increasing lower bounds of (2.7), but convergence to the optimum is not guaranteed.

Remark 2.18:
In the literature, this construction is known as *Method of Moments* (MM), where it is assumed that $t \geq \max_i \operatorname{td}(g_i)$ in addition to $t \geq \operatorname{td}(f)$ in order to start with a complete description of all the constraints used in the problem. Our slightly different definition is more flexible by enabling us to start with an incomplete set of constraints of low degree while still fitting into the overall hierarchy.

It should be noted that using a value of t that truncates most of the 'relevant' inequalities for the problem is not likely to yield a useful lower bound.

For convenience, we will also introduce a shorthand notation for polynomial equations $h(\boldsymbol{x}) = 0$ (imposed by having both $h(\boldsymbol{x}) \geq 0$ and $-h(\boldsymbol{x}) \geq 0$) by setting

$$\mathcal{N}_t^*(\mathcal{H}, \mathcal{G}) := \left\{ \boldsymbol{y} \in \mathbb{R}^{\mathbb{N}^d_{2t}} \;\middle|\; \begin{array}{rcl} \mathbf{M}_t(\boldsymbol{y}) & \succeq & \mathbf{0} \\ \mathbf{M}_t(g, \boldsymbol{y}) & \succeq & \mathbf{0} \quad \forall g \in \mathcal{G} : t \geq \operatorname{td}(g) \\ \mathbf{M}_t(h, \boldsymbol{y}) & = & \mathbf{0} \quad \forall h \in \mathcal{H} : t \geq \operatorname{td}(h) \end{array} \right\}. \tag{2.11}$$

If the description of a basic semialgebraic set

$$C = \left\{ \boldsymbol{x} \in \mathbb{R}^d \;\middle|\; h(\boldsymbol{x}) = 0 \; \forall h \in \mathcal{H},\; g(\boldsymbol{x}) \geq 0 \; \forall g \in \mathcal{G} \right\}$$

is clear from context, we will also abuse notation to write $\mathcal{N}_t^*(C)$ instead of $\mathcal{N}_t^*(\mathcal{H}, \mathcal{G})$.

Chapter 3

Computational Aspects

The goal of this chapter is to outline the computational aspects of MM, that is, how to implement and formulate the relaxations encountered in Section 2.6 in practice.

As a preprocessing step, we explain in Section 3.1 how the results from Section 2.5 can be exploited to reduce the size of the moment matrices when optimizing over a variety. Section 3.2 then describes the software and hardware that we used to carry out all experiments throughout the thesis. Lastly, Section 3.3 gives a new method to solve conic linear problems.

3.1 Method of Moments on a Variety

When using MM on a set K whose description includes polynomial equations, the size of the moment matrices involved can be reduced. To see this, let (2.9) be the intersection

$$C = \left\{ \boldsymbol{x} \in \mathbb{R}^d \;\middle|\; g(\boldsymbol{x}) \geq 0 \quad \forall g \in \mathcal{G} \right\} \cap \mathcal{V}_{\mathbb{R}}\left(\langle \mathcal{H} \rangle\right)$$

of an basic semialgebraic set and a real variety. From an algebraic point of view, the constraints $\mathbf{M}_t(h, \boldsymbol{y}) = \mathbf{0}$ in (2.11) are not very natural, since they depend on a specific generating system of their underlying ideal $\mathcal{I} = \langle \mathcal{H} \rangle$ and neglect the overall structure of $\mathcal{I}$. In addition, we have the issue of redundancy in the form of inherent linear dependent columns in the moment matrices, as shown by the following lemma.

Lemma 3.1:
For $h, g \in \mathbb{R}[\boldsymbol{x}]$, let $t \in \mathbb{N}$ satisfy $t \geq \deg(h) + \mathrm{td}(g)$. If $\boldsymbol{y} \in \mathbb{R}^{\mathbb{N}^d}$ satisfies $\mathbf{M}_{t+1}(h, \boldsymbol{y}) = \mathbf{0}$, then

$$\mathbf{M}_t(g, \boldsymbol{y}) \cdot \boldsymbol{h} = \mathbf{0}.$$

Proof. Let $h(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha} \in \mathbb{N}^d} h_{\boldsymbol{\alpha}} \cdot \boldsymbol{x}^{\boldsymbol{\alpha}}$ and $g(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha} \in \mathbb{N}^d} g_{\boldsymbol{\alpha}} \cdot \boldsymbol{x}^{\boldsymbol{\alpha}}$. Due to linearity of $L_{\boldsymbol{y}}$, for

each $\boldsymbol{\gamma} \in \mathbb{N}^d_{t-\mathrm{td}(g)}$ it holds that

$$\begin{aligned}(\mathbf{M}_t(g,\boldsymbol{y})\cdot \boldsymbol{h})_{\boldsymbol{\gamma}} &= \sum_{\boldsymbol{\alpha}\in\mathbb{N}^d_{\deg(h)}} h_{\boldsymbol{\alpha}}\cdot L_{\boldsymbol{y}}(\boldsymbol{x}^{\boldsymbol{\alpha}}\cdot\boldsymbol{x}^{\boldsymbol{\gamma}}\cdot g) = L_{\boldsymbol{y}}(\boldsymbol{x}^{\boldsymbol{\gamma}}\cdot g\cdot h)\\ &= \sum_{\boldsymbol{\beta}\in\mathbb{N}^d_{\deg(g)}} g_{\boldsymbol{\beta}}\cdot L_{\boldsymbol{y}}(\boldsymbol{x}^{\boldsymbol{\beta}}\cdot\boldsymbol{x}^{\boldsymbol{\gamma}}\cdot h). \qquad (3.1)\end{aligned}$$

Then the inequality $\deg(\boldsymbol{x}^{\boldsymbol{\beta}+\boldsymbol{\gamma}}) \le 2\,(t+1-\mathrm{td}(h))$ is sharp, depending on the parities of $\deg(g)$ and $\deg(h)$. Consequently, $\mathbf{M}_{t+1}(h,\boldsymbol{y})$ is the smallest moment matrix that sets all entries $L_{\boldsymbol{y}}(\boldsymbol{x}^{\boldsymbol{\beta}}\cdot\boldsymbol{x}^{\boldsymbol{\gamma}}\cdot h)$ occuring in (3.1) to 0. □

Lemma 3.1 needs $\mathbf{M}_{t+1}(h,\boldsymbol{y}) = \mathbf{0}$ rather than $\mathbf{M}_t(h,\boldsymbol{y}) = \mathbf{0}$ when $\deg(g)$ is even while $\deg(h)$ is odd, as shown in the next example.

Example 3.2:
Let $d = 2$, $g(\boldsymbol{x}) = 1$ and $h(x) = x_1^3 - x_2^2 \in \mathbb{R}[\boldsymbol{x}]$. For $t = 3$, we have

$$\mathbf{M}_2(h,\boldsymbol{y}) = \begin{pmatrix} y_{(3,0)}-y_{(0,2)} & y_{(4,0)}-y_{(1,2)} & y_{(3,1)}-y_{(0,3)}\\ y_{(4,0)}-y_{(1,2)} & y_{(5,0)}-y_{(2,2)} & y_{(4,1)}-y_{(1,3)}\\ y_{(3,1)}-y_{(0,3)} & y_{(4,1)}-y_{(1,3)} & y_{(3,2)}-y_{(0,4)}\end{pmatrix},$$

but

$$\mathbf{M}_2(g,\boldsymbol{y})\cdot\boldsymbol{h} = \begin{pmatrix} y_{(3,0)}-y_{(0,2)}\\ y_{(4,0)}-y_{(1,2)}\\ y_{(3,1)}-y_{(0,3)}\\ y_{(5,0)}-y_{(2,2)}\\ y_{(4,1)}-y_{(1,3)}\\ y_{(3,2)}-y_{(0,4)}\\ y_{(6,0)}-y_{(3,2)}\\ y_{(5,1)}-y_{(2,3)}\\ y_{(4,2)}-y_{(1,4)}\\ y_{(3,3)}-y_{(0,5)}\end{pmatrix} = \begin{pmatrix} 0\\0\\0\\0\\0\\0\\ y_{(6,0)}-y_{(3,2)}\\ y_{(5,1)}-y_{(2,3)}\\ y_{(4,2)}-y_{(1,4)}\\ y_{(3,3)}-y_{(0,5)}\end{pmatrix}$$

is not necessarily $\mathbf{0}$. Note that the non-zero entries all contain linearizations of polynomials containing monomials of degree $2t$.

In particular, we get the following corollary.

Corollary 3.3:
For $h, g \in \mathbb{R}[\boldsymbol{x}]$, let $t \in \mathbb{N}$ satisfy $t \ge \deg(h) + \mathrm{td}(g)$. Then

$$\mathbf{M}_t(g,\boldsymbol{y})\cdot\boldsymbol{h} = \mathbf{0}$$

is a valid constraint for $\mathcal{N}_t^*(\{h\},\{g\})$ and can be used to reduce the size of $\mathbf{M}_t(g,\boldsymbol{y}) \succeq \mathbf{0}$.

Proof. Lemma 3.1 shows that the constraint is valid, and Lemma 2.4 shows that we can simultaneously reduce one column and row of $\mathbf{M}_t(g, \boldsymbol{y})$ to $\mathbf{0}$, making it obsolete for the psd. constraint. □

To remedy this, one can instead set up MM in the quotient ring $\mathbb{R}[\boldsymbol{x}]/\mathcal{I}$ as opposed to $\mathbb{R}[\boldsymbol{x}]$, working with the projections of the inequalities given by $\mathcal{G}$ onto $\mathbb{R}[\boldsymbol{x}]/\mathcal{I}$. This approach results in the theory of theta bodies, as explained in [BPT13, Chapter 6]. There, it is shown that compared to the approach outlined so far, working in the quotient ring can yield tighter relaxations with smaller moment matrices, albeit at the cost of doing computations in $\mathbb{R}[\boldsymbol{x}]/\mathcal{I}$ instead of $\mathbb{R}[\boldsymbol{x}]$.

Definition 3.4:
Given an ideal $\mathcal{I} \subseteq \mathbb{R}[\boldsymbol{x}]$, $f \in \mathbb{R}[\boldsymbol{x}]$ and $t \geq \operatorname{td}(f)$, the matrix $\mathbf{M}_t^{\mathcal{I}}(f, \boldsymbol{y})$ indexed by

$$\left\{ \boldsymbol{\alpha} \in \mathbb{N}^d_{t-\operatorname{td}(f)} \;\middle|\; \boldsymbol{x}^{\boldsymbol{\alpha}} \notin LT_{grlex}(\mathcal{I}) \right\}$$

and entrywise defined as

$$(\mathbf{M}_t^{\mathcal{I}}(f, \boldsymbol{y}))_{\boldsymbol{\alpha},\boldsymbol{\beta}} := L_y\left(\overline{\boldsymbol{x}^{\boldsymbol{\alpha}} \cdot \boldsymbol{x}^{\boldsymbol{\beta}} \cdot f}\right),$$

is called the *reduced localizing moment matrix* of order t of $\boldsymbol{y}$ with respect to f.

Example 3.5:
Recall Example 2.16, where we considered moment matrices for $t = 2$ and the bivariate polynomials $h_i(\boldsymbol{x}) = x_i^2 - x_i$, which define a basis for $\mathcal{I}(C^2)$. Instead of using the constraints

$$\mathbf{M}_2(\boldsymbol{y}) \succeq \mathbf{0}, \quad \mathbf{M}_2(h_1, \boldsymbol{y}) = \mathbf{0} \quad \text{and} \quad \mathbf{M}_2(h_2, \boldsymbol{y}) = \mathbf{0},$$

to describe C^2, we can instead consider $v_2(\boldsymbol{x})v_2(\boldsymbol{x})^\top \bmod \mathcal{I}(C^2)$ to get

$$v_2(\boldsymbol{x})v_2(\boldsymbol{x})^\top \equiv \left(\begin{array}{c|cc|cc|c} 1 & x_1 & x_2 & x_1 & x_1x_2 & x_2 \\ \hline x_1 & x_1 & x_1x_2 & x_1 & x_1x_2 & x_1x_2 \\ x_2 & x_1x_2 & x_2 & x_1x_2 & x_1x_2 & x_2 \\ \hline x_1 & x_1 & x_1x_2 & x_1 & x_1x_2 & x_1x_2 \\ x_1x_2 & x_1x_2 & x_1x_2 & x_1x_2 & x_1x_2 & x_1x_2 \\ x_2 & x_1x_2 & x_2 & x_1x_2 & x_1x_2 & x_2 \end{array}\right) \mod \mathcal{I}(C^2). \tag{3.2}$$

As one can see, Corollary 3.3 shows that the coefficient vectors of h_1 and h_2 given by

$$\boldsymbol{h}_1 = (0, -1, 0, 1, 0, 0) \text{ and } \boldsymbol{h}_2 = (0, 0, -1, 0, 0, 1)$$

belong to the kernel of the matrix-polynomial (3.2) independently of the evaluation of the variable $\boldsymbol{x}$. Consequently, using a basis of $\mathbb{R}_2[\boldsymbol{x}]/\mathcal{I}(C^2)$ removes the linear depen-

dent columns and rows arising this way and yields the reduced matrices

$$\left(\begin{array}{c|c|c|c} 1 & x_1 & x_2 & x_1x_2 \\ \hline x_1 & x_1 & x_1x_2 & x_1x_2 \\ x_2 & x_1x_2 & x_2 & x_1x_2 \\ \hline x_1x_2 & x_1x_2 & x_1x_2 & x_1x_2 \end{array}\right) \mapsto \mathbf{M}_2^{\mathcal{I}(C^2)}(\boldsymbol{y}) = \left(\begin{array}{c|c|c|c} y_{(0,0)} & y_{(1,0)} & y_{(0,1)} & y_{(1,1)} \\ \hline y_{(1,0)} & y_{(1,0)} & y_{(1,1)} & y_{(1,1)} \\ y_{(0,1)} & y_{(1,1)} & y_{(0,1)} & y_{(1,1)} \\ \hline y_{(1,1)} & y_{(1,1)} & y_{(1,1)} & y_{(1,1)} \end{array}\right),$$

where the latter only needs 4 variables out of the total 15 variables contained in $\boldsymbol{y} \in \mathbb{R}^{\mathbb{N}_4^2}$ used in the original approach.

This motivates the usage of the following theorem.

Theorem 3.6 (Moment Matrix Reduction)**:**
Let $\mathcal{H}, \mathcal{G} \subseteq \mathbb{R}[\boldsymbol{x}]$, $\langle \mathcal{H} \rangle = \mathcal{I}$ and $t \geq \max\{\mathrm{td}(h) \mid h \in \mathcal{H}\}$, then for

$$\overline{\mathcal{N}_t^*}(\mathcal{H}, \mathcal{G}) := \left\{ \boldsymbol{y} \in \mathbb{R}^{\mathbb{N}_{2t}^d} \;\middle|\; \begin{array}{lll} \mathbf{M}_t^{\mathcal{I}}(\boldsymbol{y}) & \succeq \mathbf{0} & \\ \mathbf{M}_t^{\mathcal{I}}(g, \boldsymbol{y}) & \succeq \mathbf{0} & \forall g \in \mathcal{G} : t \geq \mathrm{td}(g) \\ \mathbf{M}_t(h, \boldsymbol{y}) & = \mathbf{0} & \forall h \in \mathcal{H} \end{array} \right\}$$

we get the inclusion

$$\mathcal{N}^*(\mathcal{H}, \mathcal{G}) \subseteq \overline{\mathcal{N}_t^*}(\mathcal{H}, \mathcal{G}) \subseteq \mathcal{N}_t^*(\mathcal{H}, \mathcal{G}).$$

In particular, the constraints $\mathbf{M}_t(h, \boldsymbol{y}) = \mathbf{0}$ are only necessary to compute the entries of $\boldsymbol{y} \in \overline{\mathcal{N}_t^*}(\mathcal{H}, \mathcal{G})$ that don't appear in the matrix $\mathbf{M}_t^{\mathcal{I}}(\boldsymbol{y})$.

The reason why $\mathbb{R}[\boldsymbol{x}]/\mathcal{I}$ and its degree-truncations admit smaller moment matrices in Theorem 3.6 is the reduction given by Theorem 2.10. Replacing the standard monomial basis by the basis given in Theorem 2.10 removes all linear dependencies that are implied by polynomials in $\mathcal{I}$ through Corollary 3.3. However, depending on $\mathcal{I}$, the computation of a Gröbner basis may not be available in practice. Fortunately, the ideals shown in Examples 2.12, 2.13 and 2.14 already cover the most important cases we will examine. In particular, reduced moment matrices using the basis of $\mathbb{R}[x]/\mathcal{I}(C^d)$ in Example 2.12 are called *combinatorial moment matrices* and have long been studied in the context of combinatorial optimization [Lau03].

In the case that a Gröbner basis is not available, we can still use Corollary 3.3 iteratively as a preprocessing step to reduce the size of $\mathbf{M}_t(g, \boldsymbol{y})$ significantly to the point where the columns do not contain the support of any entry in $\mathbf{M}_t(h, \boldsymbol{y})$.

3.2 Conic Linear Programming

The task of optimizing over the sets $\mathcal{N}_t^*(\mathcal{H}, \mathcal{G})$ or $\overline{\mathcal{N}_t^*}(\mathcal{H}, \mathcal{G})$ belongs to the problem class of *conic linear problems* (or *CLP*), which all can be transformed into the standard form

$$\min\{\langle \boldsymbol{c}, \boldsymbol{x} \rangle \mid \mathbf{A}\boldsymbol{x} = \boldsymbol{b},\ \boldsymbol{x} \in \mathcal{K}\}, \tag{3.3}$$

for some convex cone $\mathcal{K} \subseteq \mathbb{R}^d$.

We will assume that the reader is familiar with the basic concepts of (conic) linear programming and suggest the book [AL12] for an in-depth treatment of both theory and practice. Currently, the field of conic linear programming is dominated by interior point methods, and this class of methods has seen numerous improvements due to being studied extensively in recent years [NT08]. Unfortunately, scaling becomes a major problem with these methods, making them prohibitive for large problems.

To fill this gap, another line of research is concerned with well-scaling first-order methods [EZC10]. For example, [OCPB16] recently proposed a combination of operator splitting and homogeneous self-dual embedding to tackle this problem and was able to beat state-of-the-art interior point methods on large instances. However, while first-order methods may perform faster in general, they usually do so at the expense of accuracy.

Setup

In order to solve any conic linear problems arising in this thesis, we utilized the widely used package SDPT3 [TTT96, TTT03] for MATLAB version 8.1.0.604 (R2013) with an Intel i5 of 3.2 GHz $\times$ 4 and 16 GB of memory. In particular, the package can solve problems given in the standard form (3.3) when $\mathcal{K} \subseteq \mathbb{R}^d$ is a convex cone given as the Cartesian product of any combination of the cones $\mathbb{R}^n$, $\mathbb{R}^n_+$, $\mathcal{S}^n_+$ and $\mathcal{L}_n$ in various dimensions n, which is true for all problems that we will encounter.

Remark 3.7:
In this context, the psd. matrix cone $\mathcal{S}^n_+$ is identified with the vector space of its upper triangular entries in $\mathbb{R}^{\binom{n}{2}}$ and thus vectorized.

The package uses an infeasible primal-dual path-following algorithm and thus belongs to the class of interior point methods. While the algorithm can be considered as an oracle for the sake of this thesis, we encourage the interested reader to read up on the underlying ideas in [TTT03].

3.3 The LP-Newton Method for CLPs

As an alternative to the interior point method implemented in SDPT3, and to give a self contained treatment of how to solve (3.3), we introduce the *conic* LP-*Newton method* (or CLP-Newton method), which is a generalization of the LP-*Newton method* devised in [FHYZ08].

The original method computes the end-point of the intersection of a line and a zonotope, and L and P stand for *line* and (convex) *polyhedron*, respectively. In particular, it was shown that for the case of $\mathcal{K} = \mathbb{R}^n_+$, solving (3.3) can be recast as finding the

endpoint of such an intersection under mild assumptions, and we will show that the same is true for any self-dual cone $\mathcal{K}$.

While interior point methods either start in or eventually enter the feasible region and then traverse it towards an optimal solution, the CLP-Newton-method starts with an infeasible point and converges towards an optimal feasible solution exclusively from outside of the feasible set. Although both approaches utilize the Newton method in some form, they function very differently, and the CLP-Newton method is conceptually closer to algorithms designed for feasibility problems.

Overview

This section is divided into two parts.

In the first part, we restate the results from [FHYZ08] in the setting of CLPs. For this, we introduce conic zonotopes and construct directly the adapted CLP-Newton method for them in Section (3.3.1). Since the minimum-norm-point algorithm is necessary for the projection step of the algorithm, it is recalled in Section (3.3.2).

In the second part, we show how to make the method explicit for widespread convex cones and evaluate the method based on experiments. In particular, Section (3.3.3) contains conditions for $\mathcal{K}$ to be exploited by the algorithm and considers the cones $\mathbb{R}^n_+$, $\mathcal{L}_n$ and $\mathcal{S}^n_+$, which are all covered by SDPT3 as well. Finally, Section (3.3.4) reports experiments on $\mathcal{L}_n$ and $\mathcal{S}^n_+$, which shows how the algorithm performs in cases that were not considered in the original paper [FHYZ08].

3.3.1 The CLP-Newton Method

Conic Zonotopes

Throughout this section, let $\mathcal{K} \subseteq \mathbb{R}^n$ be a proper self-dual cone as in Section 2.4. Recall that $\leq_{\mathcal{K}}$ defines a partial order on $\mathbb{R}^n$ by defining for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ that

$$\boldsymbol{y} \leq_{\mathcal{K}} \boldsymbol{x} \quad \Leftrightarrow \quad \boldsymbol{x} - \boldsymbol{y} \in \mathcal{K},$$

where $\boldsymbol{0} \leq_{\mathcal{K}} \boldsymbol{x}$ is short for membership $\boldsymbol{x} \in \mathcal{K}$. Extending this concept, we also define

$$\boldsymbol{y} <_{\mathcal{K}} \boldsymbol{x} \quad \Leftrightarrow \quad \boldsymbol{x} - \boldsymbol{y} \in \mathcal{K}, \quad \boldsymbol{x} \neq \boldsymbol{y}.$$

For any two points $\boldsymbol{l} \leq_{\mathcal{K}} \boldsymbol{u} \in \mathbb{R}^n$, we define their $\mathcal{K}$-*box* as

$$[\boldsymbol{l}, \boldsymbol{u}]_{\mathcal{K}} := \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{l} \leq_{\mathcal{K}} \boldsymbol{x} \leq_{\mathcal{K}} \boldsymbol{u}\}.$$

Throughout this section, we will assume that $\boldsymbol{l} <_{\mathcal{K}} \boldsymbol{u}$, which implies the inclusion of the line segment between $\boldsymbol{l}$ and $\boldsymbol{u}$ in their $\mathcal{K}$-box, showing that

$$\operatorname{conv}(\{\boldsymbol{l}, \boldsymbol{u}\}) \subseteq [\boldsymbol{l}, \boldsymbol{u}]_{\mathcal{K}}$$

is nonempty and nontrivial. Now let $\mathbf{A} \in \mathbb{R}^{m \times n}$ and define

$$Z = \{z \mid z = \mathbf{A}\boldsymbol{x}, \boldsymbol{x} \in [\boldsymbol{l}, \boldsymbol{u}]_{\mathcal{K}}\},$$

which we will call a $\mathcal{K}$-*zonotope* in $\mathbb{R}^m$. This generalizes the already established *zonotopes* [Zie95], which correspond to $\mathbb{R}^n_+$-zonotopes in our setting.

Remark 3.8:
Linear optimization over Z is equivalent to linear optimization over $[\boldsymbol{l}, \boldsymbol{u}]_{\mathcal{K}}$, as

$$\max\{\langle \boldsymbol{c}, z\rangle \mid z \in Z\} = \max\left\{\langle \mathbf{A}^\top \boldsymbol{c}, \boldsymbol{x}\rangle \;\middle|\; \boldsymbol{x} \in [\boldsymbol{l}, \boldsymbol{u}]_{\mathcal{K}}\right\}.$$

CLP reformulation

Consider the CLP

$$\max\{\langle \boldsymbol{c}, \boldsymbol{x}\rangle \mid \mathbf{A}\boldsymbol{x} = \boldsymbol{b},\ \boldsymbol{x} \in [\boldsymbol{l}, \boldsymbol{u}]_{\mathcal{K}}\}. \tag{Box-CLP}$$

This is slightly different from the standard form (3.3) for CLPs, but can be cast into the form of (Box-CLP) given appropriate bounds on the feasible region, which are often available or can be easily computed.

Following [FHYZ08], we define an $(m+1) \times n$ matrix

$$\bar{\mathbf{A}} = \begin{pmatrix} \mathbf{A} \\ \boldsymbol{c}^\top \end{pmatrix}$$

as well as a $\mathcal{K}$-zonotope

$$\bar{Z} = \{z \mid z = \bar{\mathbf{A}}\boldsymbol{x},\ \boldsymbol{x} \in [\boldsymbol{l}, \boldsymbol{u}]_{\mathcal{K}}\}$$

and the line

$$LL = \left\{ \begin{pmatrix} \boldsymbol{b} \\ \gamma \end{pmatrix} \;\middle|\; \gamma \in \mathbb{R} \right\}.$$

Using this notation, problem (Box-CLP) can be restated as

$$\gamma^* = \max\left\{ \gamma \;\middle|\; \begin{pmatrix} z \\ \gamma \end{pmatrix} \in LL \cap \bar{Z} \right\}, \tag{Box-CLP′}$$

where $\gamma \in \mathbb{R}$. The rest of this section is dedicated to solving (Box-CLP′).

The Method

In order to describe the CLP-Newton method, we need to introduce additional definitions. To start, let

$$\gamma_0 = \max\{\langle \boldsymbol{c}, \boldsymbol{x}\rangle \mid \boldsymbol{x} \in [\boldsymbol{l}, \boldsymbol{u}]_{\mathcal{K}}\}, \tag{3.4}$$

which is a relaxation of (Box-CLP′) and as such yields an upperbound of γ^*. We will also need the orthogonal projection π_C on a given convex set $C \subseteq \mathbb{R}^n$ from Section 2.4. Lastly, we introduce the shorthand

$$\bar{\boldsymbol{b}}(\gamma) = \begin{pmatrix} \boldsymbol{b} \\ \gamma \end{pmatrix} \quad \forall \gamma \in \mathbb{R}$$

to parametrize LL and simplify notation. The following observation is the basis for the CLP-Newton method.

Consider the continuous, convex scalar function $g : \mathbb{R} \to \mathbb{R}$ given by

$$\gamma \mapsto g(\gamma) := \left\|\bar{\boldsymbol{b}}(\gamma) - \pi_{\bar{Z}}\left(\bar{\boldsymbol{b}}(\gamma)\right)\right\|_2 = \operatorname{dist}(\bar{\boldsymbol{b}}(\gamma), \bar{Z}).$$

Then $g(\gamma) = 0$ if and only if $\bar{\boldsymbol{b}}(\gamma) \in \bar{Z}$, and since $\bar{\boldsymbol{b}}(\gamma)$ parametrizes the line LL, the set of zeros of g is a parametrization of the set $LL \cap \bar{Z}$ via $\gamma \mapsto \bar{\boldsymbol{b}}(\boldsymbol{\gamma})$. Moreover, it follows by definition that the zeros of g also coincide with all possible values of the objective function of (Box-CLP′), which necessarily form a closed interval. This leads us to the following result.

Lemma 3.9:
The optimal value γ^* of (Box-CLP′) is equal to

$$\max\{\gamma \in \mathbb{R} \mid g(\gamma) = 0\}. \tag{MZ}$$

Now, applying the CLP-Newton method to solve (Box-CLP′) consists of solving (MZ) with a generalized Newton method instead. The idea is that since g is convex and γ_0 is an upperbound to γ^*, we can start such a method at γ_0 and are guaranteed to converge towards γ^* from above. In general, g will be non-differentiable, and we need to use the subdifferential instead of the usual derivative. For this, the following lemma shows that we can extract a point in the subdifferential of g from a projection onto $\bar{Z}$.

Lemma 3.10:
For all $\gamma \in \mathbb{R}$ we have

$$\frac{\gamma - \left\langle \boldsymbol{e}_{m+1}, \pi_{\bar{Z}}(\bar{\boldsymbol{b}}(\gamma))\right\rangle}{\operatorname{dist}\left(\bar{\boldsymbol{b}}(\gamma), \bar{Z}\right)} \in \partial g(\gamma).$$

Proof. The function $g(\gamma) = \operatorname{dist}\left(\bar{\boldsymbol{b}}(\gamma), \bar{Z}\right)$ is the composition of an affine map

$$\bar{\boldsymbol{b}}(\gamma) = \begin{pmatrix} \boldsymbol{0} \\ 1 \end{pmatrix} \gamma + \begin{pmatrix} \boldsymbol{b} \\ 0 \end{pmatrix}$$

and a distance function, so we can use the chain rule for subderivatives [Roc09, Ch. A] to yield

$$\partial g(\gamma) = \boldsymbol{e}_{m+1}^{\top} \cdot \partial_{\boldsymbol{x}=\bar{\boldsymbol{b}}(\gamma)} \operatorname{dist}\left(\boldsymbol{x}, \bar{Z}\right).$$

By Lemma 2.2, we have

$$\frac{\boldsymbol{x} - \pi_{\bar{Z}}(\boldsymbol{x})}{\operatorname{dist}\left(\boldsymbol{x}, \bar{Z}\right)} \in \partial \operatorname{dist}(\boldsymbol{x}, \bar{Z}),$$

and thus

$$\frac{\gamma - \left\langle \boldsymbol{e}_{m+1}, \pi_{\bar{Z}}(\bar{\boldsymbol{b}}(\gamma))\right\rangle}{\operatorname{dist}\left(\bar{\boldsymbol{b}}(\gamma), \bar{Z}\right)} = \frac{\left\langle \boldsymbol{e}_{m+1}, \bar{\boldsymbol{b}}(\gamma) - \pi_{\bar{Z}}(\bar{\boldsymbol{b}}(\gamma))\right\rangle}{\operatorname{dist}\left(\bar{\boldsymbol{b}}(\gamma), \bar{Z}\right)} \in \partial g(\gamma).$$

□

Now that we can extract a point in the subdifferential from a projection onto $\bar{Z}$, we are able to state the following algorithm.

Algorithm 3.1: The CLP-Newton Method (CLPN)

Data: Data $\mathbf{A}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{l}, \boldsymbol{u}$ for (Box-CLP′), error tolerance $\varepsilon > 0$.
Result: An approximate solution $\boldsymbol{x}_k$ for (Box-CLP′) or the detection that (Box-CLP′) is infeasible.

Compute $\gamma_0 = \max\{\langle \boldsymbol{c}, \boldsymbol{x}\rangle \mid \boldsymbol{x} \in [\boldsymbol{l}, \boldsymbol{u}]_{\mathcal{K}}\}$;
for $k = 1, 2, \ldots$ **do**
  Find $\boldsymbol{x}_k$ such that $\bar{\mathbf{A}}\boldsymbol{x}_k = \pi_{\bar{Z}}(\bar{\boldsymbol{b}}(\gamma_{k-1}))$;
  if $\left\|\bar{\mathbf{A}}\boldsymbol{x}_k - \bar{\boldsymbol{b}}(\gamma_{k-1})\right\|_2 < \varepsilon$ **then**
    return $\boldsymbol{x}_k$;
  Set $(\boldsymbol{z}_k^{\top}, \zeta_k)^{\top} = \bar{\mathbf{A}}\boldsymbol{x}_k$;
  if $\zeta_k \geq \gamma_{k-1}$ **then**
    return *"Problem (Box-CLP′) is infeasible"*;
  Compute $\gamma_k = \zeta_k - \|\boldsymbol{b} - \boldsymbol{z}_k\|_2^2 \cdot (\gamma_{k-1} - \zeta_k)^{-1}$;

After initializing γ_0 as the starting value, the k-th step of Algorithm 3.1 checks for stopping criteria and then performs a Newton step on the current iterate γ_{k-1} as follows.

First, $g(\gamma_{k-1})$ is computed, and if $g(\gamma_{k-1}) \in [0, \varepsilon)$ for a given precision ε, then γ_{k-1} is accepted as a zero of g, thus terminating the algorithm.

In the next part, it is checked whether the new iterate passed the minimum of g, indicating that g has no zeros, thus proving the problem is infeasible.

If neither of these conditions is fulfilled, a new iterate γ_k can be computed by performing a Newton step. To do this, we choose $h_k \in \partial g(\gamma_{k-1})$ as in Lemma 3.10 in the recursion

$$\gamma_k = \gamma_{k-1} - \frac{g(\gamma_{k-1})}{h_k}$$

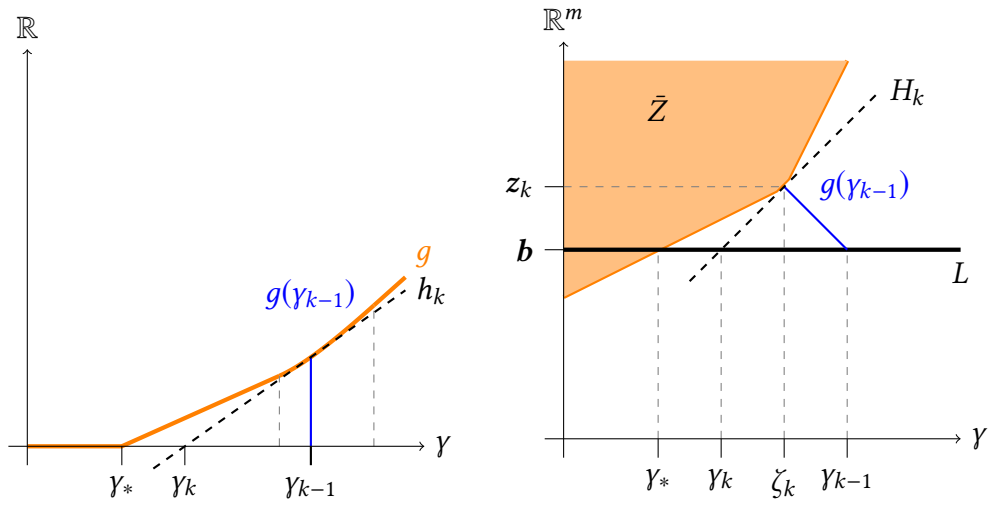

Figure 3.1: Visualization of the update step (3.5). Left: Newton step using g. Right: Geometric deduction from supporting hyperplane H_k.

to get

$$\gamma_k = \gamma_{k-1} - \frac{\operatorname{dist}(\bar{\boldsymbol{b}}(\gamma_{k-1}), \bar{Z})^2}{\gamma_{k-1} - \zeta_k} = \zeta_k - \frac{\|\boldsymbol{b} - \boldsymbol{z}_k\|_2^2}{\gamma_{k-1} - \zeta_k}. \tag{3.5}$$

We can also understand the Newton step geometrically by first noting that

$$H_k = \left\{ \begin{pmatrix} \boldsymbol{z} \\ \zeta \end{pmatrix} \in \mathbb{R}^{m+1} \;\middle|\; \left\langle \begin{pmatrix} \boldsymbol{z} \\ \zeta \end{pmatrix} - \begin{pmatrix} \boldsymbol{z}_k \\ \zeta_k \end{pmatrix}, \begin{pmatrix} \boldsymbol{b} \\ \gamma_{k-1} \end{pmatrix} - \begin{pmatrix} \boldsymbol{z}_k \\ \zeta_k \end{pmatrix} \right\rangle = 0 \right\}$$

is a supporting hyperplane of $\bar{Z}$ at $\pi_{\bar{Z}}(\bar{b}(\gamma_{k-1}))$. Then any feasible point in $LL \cap \bar{Z}$ is necessarily contained in the halfspace defined by H_k not containing $\bar{\boldsymbol{b}}(\gamma_{k-1})$, and (3.5) computes the intersection $LL \cap H_k$. Figure 3.1 compares the two approaches to deduce (3.5).

The following was shown in the original paper [FHYZ08] and generalizes to our setting.

Lemma 3.11 ([FHYZ08])**:**
The following statements hold for all values of k attained in Algorithm 3.1.

(i) If $\gamma_{k-1} > \zeta_k$, then $\zeta_k > \gamma_k$.

(ii) If $\gamma_{k-1} < \zeta_k$ or $\gamma_{k-1} = \zeta_k$ and $\boldsymbol{z}_k \neq \boldsymbol{b}$, then CLPN correctly assesses infeasibility of (Box-CLP′).

(iii) If $\gamma_{k-1} = \zeta_k$ and $\boldsymbol{z}_k = \boldsymbol{b}$, then γ_{k-1} is equal to the optimal value of (Box-CLP′).

Remark 3.12:
Since g is convex, Algorithm 3.1 falls into the class of generalized Newton methods, which immediately shows asymptotic convergence in case that (Box-CLP′) is feasible and finite termination in case that there is no feasible solution.

3.3.2 The Minimum-Norm-Point Algorithm

Algorithm 3.1 is limited by its routines to compute γ_0 and evaluate $\pi_{\bar{Z}}$, so both operations need to be efficient for practical usage. Fortunately, we can use Algorithm 3.2, an adaption of the minimum-norm-point algorithm taken from [Bac11, Jag13], to reduce the computation of $\pi_{\bar{Z}}$ to a series of conic LPs of the form (3.4).

Algorithm 3.2: Minimum-Norm-Point Algorithm (MNP) for $\operatorname{dist}(y, \bar{Z})$

Data: Data $\boldsymbol{l}, \boldsymbol{u}, \bar{\mathbf{A}}$ for (Box-CLP′), $\bar{\boldsymbol{b}} \in \mathbb{R}^{m+1}$, $\boldsymbol{s}_0 \in [\boldsymbol{l}, \boldsymbol{u}]_{\mathcal{K}}$, error tolerance ε.
Result: $\hat{\boldsymbol{x}} \in [\boldsymbol{l}, \boldsymbol{u}]_{\mathcal{K}}$ such that $0 \leq \|\bar{\mathbf{A}}\hat{\boldsymbol{x}} - \bar{\boldsymbol{b}}\|_2 - \operatorname{dist}(\bar{\boldsymbol{b}}, \bar{Z}) \leq \varepsilon$.

```
Set P = {s_0} and k = 0;
for k = 1, 2, ... do
    Compute x_k = argmin { ||Ā x − b̄||_2^2 | x ∈ aff(P) };
    if x_k ∈ conv(P) then
        Compute s_k = argmin { ⟨s, Ā^⊤(Ā x_k − b̄)⟩ | s ∈ [l, u]_K };
        if 2⟨Ā(x_k − s_k), Ā x_k − b̄⟩ < ε then
            return x_k;
        else
            Set P = P ∪ {s_k};
    else
        Compute μ̂ = max { μ | μ ∈ [0, 1], x_{k−1} + μ(x_k − x_{k−1}) ∈ Z̄ };
        Set x_k = x_{k−1} + μ̂(x_k − x_{k−1});
        Set P to the minimal subset P′ ⊆ P such that x_k ∈ conv(P′);
```

While finite termination of MNP is established by the following theorem, it is an open problem whether the runtime of MNP can be bounded by a polynomial in the input size.

Theorem 3.13 ([Jag13])**:**
Algorithm 3.2 produces a sequence $\{\boldsymbol{x}_k\}_{k\in\mathbb{N}}$ such that for $k > 1$ and

$$h_k := 2\langle \bar{\mathbf{A}}(\boldsymbol{x}_{k-1} - \boldsymbol{s}_k), \bar{\mathbf{A}}\boldsymbol{x}_{k-1} - \boldsymbol{y}\rangle,$$

we get

$$0 \leq \|\bar{\mathbf{A}}\boldsymbol{x}_k - \boldsymbol{y}\|_2 - \operatorname{dist}(\boldsymbol{y}, \bar{Z}) \leq h_{k+1} \leq \frac{27 \operatorname{diam}(\bar{Z})^2}{4(k+2)}.$$

In particular, the algorithm works correctly and terminates after a finite number of steps.

Remark 3.14:
In [Jag13], it is shown that the preceding theorem also applies to other variants of the Frank-Wolfe algorithm that can approximate $\pi_{\bar{Z}}(\bar{\boldsymbol{b}}(\gamma_k))$. However, preliminary experiments have shown that among the variants we tested, the minimum-norm-point algorithm was the fastest. For another discussion of MNP, consider [Bac11, Sct. 9.2].

In the following, we will formalize a suitable class of problems for MNP, based on the difficulty of solving the related conic linear optimization problem of the form (3.4).

Definition 3.15:
$\mathcal{K} \subseteq \mathbb{R}^n$ is called *suitable for* CLPN if the problem

$$\max \{\langle \boldsymbol{c}, \boldsymbol{x} \rangle \mid \boldsymbol{x} \in [\boldsymbol{l}, \boldsymbol{u}]_{\mathcal{K}} \subseteq \mathbb{R}^n\} \tag{3.6}$$

can be solved in time $O(n^2)$.

This property ensures that the bottleneck of computing MNP is the evaluation of projection $\boldsymbol{x}_k = \pi_{\text{aff}(P)}(\bar{\boldsymbol{b}})$, corresponding to solving a linear system of dimension at most $n \times n$. At the same time, it guarantees an efficient way to compute the starting point γ_0.

Remark 3.16:
The set of cones suitable for CLPN is closed under taking Cartesian products. In particular, if $\mathcal{K} = \bigotimes_{i \in I} \mathcal{K}_i$, then the corresponding $\mathcal{K}$-zonotope can be decomposed into several smaller $\mathcal{K}_i$-zonotopes, making (3.6) solvable in parallel for each $\mathcal{K}_i$ and potentially accelerating the algorithm by a huge margin.

3.3.3 Linear Optimization on $\mathcal{K}$-Zonotopes

In the following, we will investigate the structure of (3.6) to get a better understanding when $\mathcal{K}$ is suitable for CLPN, and apply these insights on three well-studied cones from the literature.

Necessary conditions

Since (3.6) involves linear optimization over a convex set, the optimal solutions will necessarily be extreme points, as summarized in the following result.

Lemma 3.17 (Extreme points of $[\boldsymbol{l}, \boldsymbol{u}]_{\mathcal{K}}$)**:**
Problem (3.6) *is equivalent to*

$$\max \left\{ \langle \boldsymbol{c}, \boldsymbol{x} \rangle \;\middle|\; \boldsymbol{x} \in \{\boldsymbol{l}, \boldsymbol{u}\} \cup \left(\text{bd}(\boldsymbol{l} + \mathcal{K}) \cap \text{bd}(\boldsymbol{u} - \mathcal{K}) \right) \right\}.$$

Proof. The set of extreme points of $[\boldsymbol{l}, \boldsymbol{u}]_{\mathcal{K}}$ is

$$\text{bd}([\boldsymbol{l}, \boldsymbol{u}]_{\mathcal{K}}) = \left(\text{bd}(\boldsymbol{l} + \mathcal{K}) \cap (\boldsymbol{u} - \mathcal{K}) \right) \cup \left((\boldsymbol{l} + \mathcal{K}) \cap \text{bd}(\boldsymbol{u} - \mathcal{K}) \right).$$

Without loss of generality, we can assume for an optimal solution $\boldsymbol{x}^*$ of (3.6) that

$$\boldsymbol{x}^* \in \operatorname{bd}(\boldsymbol{l} + \mathcal{K}) \cap \operatorname{int}(\boldsymbol{u} - \mathcal{K}) \setminus \{\boldsymbol{l}\},$$

so that we can write $\boldsymbol{x}^* = \boldsymbol{l} + \boldsymbol{y}$ with $\boldsymbol{y} \in \mathcal{K} \setminus \{\boldsymbol{0}\}$. Then for small $\varepsilon > 0$, we maintain

$$\boldsymbol{l} + (1 \pm \varepsilon)\boldsymbol{y} \in \operatorname{bd}(\boldsymbol{l} + \mathcal{K}) \cap \operatorname{int}(\boldsymbol{u} - \mathcal{K}) \setminus \{\boldsymbol{l}\},$$

and the optimality of $\boldsymbol{x}^*$ implies $\langle \boldsymbol{c}, \boldsymbol{y} \rangle = 0$. But then $\langle \boldsymbol{c}, \boldsymbol{x}^* \rangle = \langle \boldsymbol{c}, \boldsymbol{l} \rangle$ and we can choose $\boldsymbol{l}$ as maximizer instead. □

The related dual problem of (3.6) also allows us some insights.

Lemma 3.18:
Let $\boldsymbol{c} = \boldsymbol{c}_+ + \boldsymbol{c}_-$ be the Moreau decomposition where

$$\boldsymbol{c}_+ = \pi_{\mathcal{K}}(\boldsymbol{c}) \quad \text{and} \quad \boldsymbol{c}_- = \pi_{-\mathcal{K}^*}(\boldsymbol{c}) = -\pi_{\mathcal{K}}(-\boldsymbol{c}),$$

since $\mathcal{K}$ is self-dual. Then the dual of (3.6) is equivalent to

$$\min \{ \langle \boldsymbol{l} - \boldsymbol{u}, \boldsymbol{y} \rangle \mid \boldsymbol{y} \geq_{\mathcal{K}} -\boldsymbol{c}_+,\ \boldsymbol{y} \geq_{\mathcal{K}} \boldsymbol{c}_- \}\,. \tag{3.7}$$

Proof. The dual problem reads

$$\min \{ \langle \boldsymbol{l}, \boldsymbol{y}_2 \rangle - \langle \boldsymbol{u}, \boldsymbol{c}_1 \rangle \mid \boldsymbol{y}_2 - \boldsymbol{y}_1 = \boldsymbol{c},\ \boldsymbol{y}_1, \boldsymbol{y}_2 \in \mathcal{K} \}\,, \tag{3.8}$$

since $\mathcal{K}$ is self-dual. We can reparametrize $\boldsymbol{y}_1 = \boldsymbol{y} - \boldsymbol{c}_-$ and $\boldsymbol{y}_2 = \boldsymbol{y} + \boldsymbol{c}_+$ to satisfy the equality constraint and get

$$\min \{ \langle \boldsymbol{l} - \boldsymbol{u}, \boldsymbol{y} \rangle + \langle \boldsymbol{l}, \boldsymbol{c}_+ \rangle + \langle \boldsymbol{u}, \boldsymbol{c}_- \rangle \mid \boldsymbol{y} \geq_{\mathcal{K}} -\boldsymbol{c}_+,\ \boldsymbol{y} \geq_{\mathcal{K}} \boldsymbol{c}_- \}\,.$$

Since $\langle \boldsymbol{l}, \boldsymbol{c}_+ \rangle + \langle \boldsymbol{u}, \boldsymbol{c}_- \rangle$ is constant, the result follows. □

While Lemma 3.18 might seem rather uninteresting on its own, it has implications for cones whose cone orders $\leq_{\mathcal{K}}$ have special structural properties.

Remark 3.19:
The optimal solution of the dual (3.7) is a least upperbound on the set $\{\boldsymbol{c}_-, -\boldsymbol{c}_+\}$ in the partial ordered set $(\mathbb{R}^n, \leq_{\mathcal{K}})$. If $(\mathbb{R}^n, \leq_{\mathcal{K}})$ is a lattice, then the solution of the dual can be recovered from the join $\boldsymbol{c}_- \vee -\boldsymbol{c}_+$ in $(\mathbb{R}^n, \leq_{\mathcal{K}})$.

The nonnegative orthant $\mathcal{K} = \mathbb{R}^n_+$

Since the original paper [FHYZ08] treats this case, it is naturally suited for CLPN. In contrast to our general setting, the cone $\mathbb{R}^n_+$ satisfies the useful property that the minimum-norm-point algorithm converges to the optimal solution in a finite number of iterations [Wol]. As such, the LP-Newton method converges in a finite number of steps [FHYZ08], and an important open problem is to decide whether this number can be bounded by a polynomial in the input size.

To see that (3.6) can easily be solved in linear time, note that a solution $\boldsymbol{x}^*$ is given by greedily choosing the largest increase of the objective function componentwise by setting

$$x_i^* = \begin{cases} l_i & \text{if } c_i < 0, \\ u_i & \text{else.} \end{cases} \tag{3.9}$$

Another way to see the optimality of the greedy algorithm is by noting that $\leq$ satisfies the lattice property. Then Remark 3.19 shows $\boldsymbol{y} = \mathbf{0}$ in (3.7), since

$$y_i = (\boldsymbol{c}_- \vee -\boldsymbol{c}_+)_i = \max\{0, -|c_i|\} = 0,$$

which confirms (3.9) through the dual variables $\boldsymbol{y}_1 = -\pi_{\mathbb{R}^n_-}(\boldsymbol{c})$ and $\boldsymbol{y}_2 = \pi_{\mathbb{R}^n_+}(\boldsymbol{c})$ in (3.8).

The Lorentz-cone $\mathcal{K} = \mathcal{L}_n$

Recall from Section 2.4 that the Lorentz-cone is given as

$$\mathcal{L}_n = \{(x_0, \tilde{\boldsymbol{x}}) \in \mathbb{R}_+ \times \mathbb{R}^n \mid \|\tilde{\boldsymbol{x}}\|_2 \leq x_0\}\,.$$

A useful property of the Lorentz-cone is the explicit description of its boundary

$$\mathrm{bd}(\mathcal{L}_n) = \{(x_0, \tilde{\boldsymbol{x}}) \in \mathbb{R}_+ \times \mathbb{R}^n \mid \|\tilde{\boldsymbol{x}}\|_2 = x_0\}\,,$$

which we can exploit by applying Lemma 3.17.

To simplify notation, we will denote the vector $\boldsymbol{a} = (a_0, a_1, \ldots, a_n) \in \mathbb{R}^{n+1}$ by writing $\boldsymbol{a}^\top = (a_0, \tilde{\boldsymbol{a}}^\top)$, where $\tilde{\boldsymbol{a}} = (a_1, \ldots, a_n) \in \mathbb{R}^n$ and $a_0 \in \mathbb{R}$. In addition, we will assign a special set $\mathcal{E}(\boldsymbol{w})$ to each $\boldsymbol{w} \in \mathcal{L}_n$, which we formally define as

$$\mathcal{E}(\boldsymbol{w}) = \left\{\boldsymbol{x} \in \mathbb{R}^{n+1} \;\middle|\; \|\tilde{\boldsymbol{x}}\|_2^2 = x_0^2, \quad \|\tilde{\boldsymbol{w}} - \tilde{\boldsymbol{x}}\|_2^2 = (w_0 - x_0)^2\right\}. \tag{3.10}$$

Using this notation, we can show the following representation result.

Lemma 3.20:
For any $\boldsymbol{w} = (w_0, \tilde{\boldsymbol{w}}) \in \mathrm{int}(\mathcal{L}_n)$, define the parameters

$$\bar{\boldsymbol{w}} := \frac{1}{w_0}\tilde{\boldsymbol{w}}, \quad \bar{w}_0 := \frac{w_0^2 - \|\tilde{\boldsymbol{w}}\|_2^2}{2w_0}, \quad \mathbf{Q} := \mathbf{I}_n - \bar{\boldsymbol{w}}\bar{\boldsymbol{w}}^\top \quad \textit{and} \quad \gamma := \sqrt{\tfrac{1}{2}w_0\bar{w}_0}.$$

Then

$$\mathcal{E}(\boldsymbol{w}) = \left\{ \boldsymbol{x} \in \mathbb{R}^{n+1} \;\middle|\; x_0 = \langle \tilde{\boldsymbol{x}}, \bar{\boldsymbol{w}} \rangle + \bar{w}_0,\ \left\| \mathbf{Q}^{\frac{1}{2}} (\tilde{\boldsymbol{x}} - \tfrac{1}{2}\tilde{\boldsymbol{w}}) \right\|_2^2 = \gamma^2 \right\}$$

and in particular, $\mathcal{E}(\boldsymbol{w})$ is an n-dimensional ellipsoid.

Proof. Subtracting the equations in (3.10) immediately shows containment in the hyperplane

$$H = \left\{ \boldsymbol{x} \in \mathbb{R}^{n+1} \,\middle|\, x_0 = \langle \tilde{\boldsymbol{x}}, \bar{\boldsymbol{w}} \rangle + \bar{w}_0 \right\}, \tag{3.11}$$

where $\bar{\boldsymbol{w}}$ and $\bar{w}_0$ are well defined since $\boldsymbol{w} \in \operatorname{int}(\mathcal{L}_n)$.

Using (3.11) in either equation in (3.10) on x_0 yields an equation of the form

$$0 = \tilde{\boldsymbol{x}}^\top \mathbf{Q} \tilde{\boldsymbol{x}} - 2\langle \bar{w}_0 \bar{\boldsymbol{w}}, \tilde{\boldsymbol{x}} \rangle - \bar{w}_0^2$$

where $\mathbf{Q}$ is positive definite since $\|\bar{\boldsymbol{w}}\|_2 < 1$. Completing the square yields the equivalent condition

$$\|\mathbf{Q}^{\frac{1}{2}} (\tilde{\boldsymbol{x}} - \bar{w}_0 \mathbf{Q}^{-1} \bar{\boldsymbol{w}})\|_2^2 = \bar{w}_0^2 + \bar{w}_0^2 \cdot \bar{\boldsymbol{w}}^\top \mathbf{Q}^{-1} \bar{\boldsymbol{w}},$$

which defines an n-dimensional ellipsoid.

Using the Sherman-Morrison formula (2.1) we can simplify

$$\bar{w}_0 \mathbf{Q}^{-1} \bar{\boldsymbol{w}} = \tfrac{1}{2} \tilde{\boldsymbol{w}}$$

and

$$\bar{w}_0^2 + \bar{w}_0^2 \cdot \bar{\boldsymbol{w}}^\top \mathbf{Q}^{-1} \bar{\boldsymbol{w}} = \tfrac{1}{2} w_0 \bar{w}_0 = \gamma^2.$$

□

Lemma 3.21:
For any $\boldsymbol{w} \in \mathcal{L}_n$, the linear optimization problem

$$\max \{ \langle \boldsymbol{c}, \boldsymbol{x} \rangle \mid \boldsymbol{x} \in \mathcal{E}(\boldsymbol{w}) \} \tag{3.12}$$

can be solved in $O(n)$.

Proof. We will distinguish the cases $\boldsymbol{w} \in \operatorname{bd}(\mathcal{L}_n)$ and $\boldsymbol{w} \in \operatorname{int}(\mathcal{L}_n)$, which can be checked in $O(n)$.

For $\boldsymbol{w} \in \operatorname{bd}(\mathcal{L}_n)$, we claim that $\mathcal{E}(\boldsymbol{w}) = \operatorname{conv}(\{\mathbf{0}, \boldsymbol{w}\})$. By assumption, $\|\tilde{\boldsymbol{w}}\|_2 = w_0$ and $\|\tilde{\boldsymbol{x}}\|_2 = x_0$, so

$$\|\tilde{\boldsymbol{w}} - \tilde{\boldsymbol{x}}\|_2 = \|\tilde{\boldsymbol{w}}\|_2 - \|\tilde{\boldsymbol{x}}\|_2 \quad \forall \boldsymbol{x} \in \mathcal{E}(\boldsymbol{w}),$$

which is only possible if $\tilde{\boldsymbol{x}}$ is a multiple of $\tilde{\boldsymbol{w}}$. Consequentially, x_0 is the same multiple of w_0, which shows the claim. Then (3.12) is equal to either 0 or $\langle \boldsymbol{c}, \boldsymbol{w} \rangle$.

For $\boldsymbol{w} \in \operatorname{int}(\mathcal{L}_n)$, we can rewrite the result of Lemma 3.20 as

$$\mathcal{E}(\boldsymbol{x}) = \left\{ \boldsymbol{x} \in \mathbb{R}^{n+1} \;\middle|\; x_0 = \langle \tilde{\boldsymbol{x}}, \bar{\boldsymbol{w}} \rangle + \bar{w}_0,\ \tilde{\boldsymbol{x}} = \mathbf{Q}^{-\frac{1}{2}} \boldsymbol{y} + \tfrac{1}{2}\tilde{\boldsymbol{w}},\ \|\boldsymbol{y}\|_2^2 = \gamma^2 \right\}.$$

This yields a reparametrization of (3.12) in terms of $\boldsymbol{y}$ with objective value

$$\langle \boldsymbol{c}, \boldsymbol{x} \rangle = \langle \tilde{\boldsymbol{c}} + \tfrac{c_0}{w_0}\tilde{\boldsymbol{w}}, \tilde{\boldsymbol{x}} \rangle + c_0 \bar{w}_0 \equiv \langle \tilde{\boldsymbol{c}} + \tfrac{c_0}{w_0}\tilde{\boldsymbol{w}}, \mathbf{Q}^{-\frac{1}{2}}\boldsymbol{y} + \tfrac{1}{2}\tilde{\boldsymbol{w}} \rangle \equiv \langle \mathbf{Q}^{-\frac{1}{2}}(\tilde{\boldsymbol{c}} + \tfrac{c_0}{w_0}\tilde{\boldsymbol{w}}), \boldsymbol{y} \rangle,$$

where $\equiv$ denotes equality up to a constant difference. We need to distinguish two cases:

If $\mathbf{Q}^{-\frac{1}{2}}(\tilde{\boldsymbol{c}} + \frac{c_0}{w_0}\tilde{\boldsymbol{w}}) = \mathbf{0}$, then every solution is optimal, and we set

$$\boldsymbol{y}^* = \gamma \cdot \frac{\boldsymbol{e}}{\sqrt{n}}.$$

Otherwise, we set

$$\boldsymbol{y}^* = \gamma \cdot \frac{\mathbf{Q}^{-\frac{1}{2}}(\tilde{\boldsymbol{c}} + \frac{c_0}{w_0}\tilde{\boldsymbol{w}})}{\|\mathbf{Q}^{-\frac{1}{2}}(\tilde{\boldsymbol{c}} + \frac{c_0}{w_0}\tilde{\boldsymbol{w}})\|_2},$$

since the optimal solution of a linear optimization problem over a scaled Euclidean ball is parallel to the objective function.

In both cases, we recover the optimal solution of (3.12) by using the parametrization

$$\tilde{\boldsymbol{x}}^* = \mathbf{Q}^{-\frac{1}{2}}\boldsymbol{y}^* + \tfrac{1}{2}\tilde{\boldsymbol{w}}, \quad x_0^* = \langle \tilde{\boldsymbol{x}}^*, \bar{\boldsymbol{w}} \rangle + \bar{w}_0.$$

To see that $\boldsymbol{x}^*$ can be computed in linear time, note that this is equivalent to carrying out a multiplication by $\mathbf{Q}^{-\frac{1}{2}}$ in time $O(n)$. This can be achieved by using a specific formula for $\mathbf{Q}^{-\frac{1}{2}}$, as we will show now. For this, we first compute a set of parameters depending on $\tilde{\boldsymbol{w}}$.

If $\tilde{\boldsymbol{w}} \neq \mathbf{0}$, we set

$$\alpha := \frac{1 - \frac{2\gamma}{w_0}}{\|\tilde{\boldsymbol{w}}\|_2^2}, \quad \beta := \frac{\alpha}{1 - \alpha\|\tilde{\boldsymbol{w}}\|_2^2} = \frac{\frac{w_0}{2\gamma} - 1}{\|\tilde{\boldsymbol{w}}\|_2^2},$$

and $\alpha = \beta = 0$ otherwise. Using $w_0 > \|\tilde{\boldsymbol{w}}\|_2$, one can verify that $\alpha, \beta \geq 0$ and a straightforward computation shows the identities

$$\mathbf{Q}^{\frac{1}{2}} = \mathbf{I}_n - \alpha\tilde{\boldsymbol{w}}\tilde{\boldsymbol{w}}^\top, \quad \mathbf{Q}^{-\frac{1}{2}} = \mathbf{I}_n + \beta\tilde{\boldsymbol{w}}\tilde{\boldsymbol{w}}^\top, \tag{3.13}$$

where one identity can be reduced to the other by the Sherman-Morrison formula (2.1). Then (3.13) shows that multiplication by $\mathbf{Q}^{-\frac{1}{2}}$ amounts to

$$\mathbf{Q}^{-\frac{1}{2}}\tilde{\boldsymbol{x}} = \tilde{\boldsymbol{x}} + \beta\langle \tilde{\boldsymbol{x}}, \tilde{\boldsymbol{w}} \rangle \tilde{\boldsymbol{w}},$$

where the right-hand side can be computed in $O(n)$. □

Theorem 3.22:
The cone $\mathcal{L}_n$ is suitable for CLPN.

Proof. Through translation we can assume that $\boldsymbol{l} = \mathbf{0}$ and focus on the case

$$\max\left\{\langle \boldsymbol{c}, \boldsymbol{x}\rangle \;\middle|\; \boldsymbol{x} \in [\mathbf{0}, \boldsymbol{w}]_{\mathcal{L}_n} \subseteq \mathbb{R}^{n+1}\right\} \tag{3.14}$$

where $\boldsymbol{w} = \boldsymbol{u} - \boldsymbol{l} \in \mathcal{L}_n \setminus \{\mathbf{0}\}$ and consequently $w_0 = u_0 - l_0 > 0$. By Lemma 3.17, it suffices to compute

$$\max\{\langle \boldsymbol{c}, \boldsymbol{x}\rangle \mid \boldsymbol{x} \in \mathrm{bd}(\mathcal{L}_n) \cap \mathrm{bd}(\boldsymbol{w} - \mathcal{L}_n) =: \mathcal{E}'(\boldsymbol{w})\}$$

and compare this value to $\langle \boldsymbol{c}, \mathbf{0}\rangle = 0$ and $\langle \boldsymbol{c}, \boldsymbol{w}\rangle$. Thus, we have

$$\begin{aligned}\mathcal{E}'(\boldsymbol{w}) &= \left\{\boldsymbol{x} \in \mathbb{R}^{n+1} \;\middle|\; \|\tilde{\boldsymbol{x}}\|_2 = x_0, \|\tilde{\boldsymbol{w}} - \tilde{\boldsymbol{x}}\|_2 = w_0 - x_0\right\} \\ &= \{\boldsymbol{x} \in \mathcal{E}(\boldsymbol{w}) \mid x_0 \in [0, w_0]\}\end{aligned}$$

by (3.10) and claim that $\mathcal{E}'(\boldsymbol{w}) = \mathcal{E}(\boldsymbol{w})$.

To see this, we can use Lemma 3.21 with objective $\boldsymbol{c}^\top = (\pm 1, \mathbf{0}^\top)$ to get

$$\max\{\pm x_0 \mid \boldsymbol{x} \in \mathcal{E}(\boldsymbol{w})\} = \tfrac{1}{2} w_0 \pm \tfrac{1}{2}\|\tilde{\boldsymbol{w}}\|_2 \in [0, w_0],$$

where the bounds follow from $\boldsymbol{w} \in \mathcal{L}_n$. Then (3.14) is equivalent to

$$\max\{\langle \boldsymbol{c}, \boldsymbol{x}\rangle \mid \boldsymbol{x} \in \mathcal{E}(\boldsymbol{w})\},$$

which can be done in linear time according to Lemma 3.21. □

As a summary of the preceding results, we close this subsection with Algorithm 3.3, an explicit linear time algorithm for solving (3.14).

Algorithm 3.3: Explicit solution for (3.14)

Data: Data $\boldsymbol{l}, \boldsymbol{u}, \boldsymbol{c}$ for problem (3.14), $\bar{w}_0, \beta, \gamma$ as in Theorem 3.22.
Result: Solution $\boldsymbol{x}^* \in [\boldsymbol{l}, \boldsymbol{u}]_{\mathcal{K}}$ to (3.14).

Initialize $\boldsymbol{w} = \boldsymbol{u} - \boldsymbol{l}$, $\boldsymbol{x}^* = \mathbf{0}$;
if $\|\tilde{\boldsymbol{w}}\|_2^2 < w_0^2$ **then**
  $\tilde{\boldsymbol{y}}^* = \tilde{\boldsymbol{c}} + \frac{c_0}{w_0}\tilde{\boldsymbol{w}}$;
  $\tilde{\boldsymbol{y}}^* = \tilde{\boldsymbol{y}}^* + \beta\langle \tilde{\boldsymbol{y}}^*, \tilde{\boldsymbol{w}}\rangle \tilde{\boldsymbol{w}}$;
  if $\tilde{\boldsymbol{y}}^* = \mathbf{0}$ **then**
    $\tilde{\boldsymbol{y}}^* = \gamma \cdot \frac{\boldsymbol{e}}{\sqrt{n}}$;
  else
    $\tilde{\boldsymbol{y}}^* = \gamma \cdot \frac{\tilde{\boldsymbol{y}}^*}{\|\tilde{\boldsymbol{y}}^*\|_2}$;
  $\tilde{\boldsymbol{x}}^* = \tilde{\boldsymbol{y}}^* + (\beta\langle \tilde{\boldsymbol{y}}^*, \tilde{\boldsymbol{w}}\rangle + \frac{1}{2})\tilde{\boldsymbol{w}}$;
  $x_0^* = \frac{1}{w_0}\langle \tilde{\boldsymbol{x}}^*, \tilde{\boldsymbol{w}}\rangle + \bar{w}_0$;
$\boldsymbol{x}^* = \boldsymbol{x}^* + \boldsymbol{l}$;
return argmax $\{\langle \boldsymbol{y}, \boldsymbol{c}\rangle \mid \boldsymbol{y} \in \{\boldsymbol{l}, \boldsymbol{x}^*, \boldsymbol{u}\}\}$;

Remark 3.23:
The parameters $\bar{w}_0$, β and γ only depend on $\boldsymbol{w} = \boldsymbol{u} - \boldsymbol{l}$. When optimizing multiple times over $[\boldsymbol{l}, \boldsymbol{u}]_{\mathcal{K}}$ with different objective functions, like in our setting, these parameters can be stored and need to be computed only once.

The positive semidefinite cone $\mathcal{K} = \mathcal{S}_+^n$

Let $\mathcal{S}_+^n$ denote the cone of symmetrical $n \times n$ matrices that are positive semidefinite and let $\preceq$ be the corresponding conic order. For statements regarding complexity, note that we can embed $\mathcal{S}_+^n \subseteq \mathbb{R}^N$ for $N = \binom{n}{2} \in O(n^2)$.

Then (3.6) reads

$$\max \{\langle \mathbf{C}, \mathbf{X} \rangle \mid \mathbf{L} \preceq \mathbf{X} \preceq \mathbf{U}\}.$$

In order to solve this problem, we will transform it into a more suitable form. Just as with the case of the Lorentz cone, we first use the substitution $\mathbf{Y} = \mathbf{X} - \mathbf{L}$ to get the equivalent problem

$$\max \{\langle \mathbf{C}, \mathbf{Y} \rangle \mid \mathbf{0} \preceq \mathbf{Y} \preceq \mathbf{W}\},$$

where $\mathbf{W} := \mathbf{U} - \mathbf{L}$ and the constant $\langle \mathbf{C}, \mathbf{L} \rangle$ was dropped from the objective. To simplify the argument, we will assume without loss of generality that $\mathbf{W} \in \operatorname{int}(\mathcal{S}_+^n)$, since singular $\mathbf{W}$ can be reduced to this case by changing to a suitable subspace. Using the Cholesky decomposition $\mathbf{W} = \mathbf{V}\mathbf{V}^\top$, we can rewrite $\mathbf{Y} = \mathbf{V}\mathbf{Z}\mathbf{V}^\top$ to get the equivalent problem

$$\max \{\langle \mathbf{C}', \mathbf{Z} \rangle \mid \mathbf{0} \preceq \mathbf{Z} \preceq \mathbf{I}_n\}, \tag{3.15}$$

where $\mathbf{C}' = \mathbf{V}^\top \mathbf{C} \mathbf{V}$.

By applying these preprocessing steps, we reduced the original constraints to box-constraints on the eigenvalues of $\mathbf{Z}$, which allow us to solve the problem explicitly.

Theorem 3.24:
Let $\mathbf{C}' = \mathbf{B}^\top \mathbf{D} \mathbf{B}$ be the eigenvalue decomposition of $\mathbf{C}'$. Then the solution to (3.15) is given by

$$\mathbf{Z} = \mathbf{B}^\top \mathbf{\Lambda}^* \mathbf{B},$$

where $\mathbf{\Lambda}^*$ is a diagonal matrix with $\operatorname{diag}(\mathbf{\Lambda}^*) = \boldsymbol{\lambda}^*$ and $\boldsymbol{\lambda}^*$ is the solution of

$$\max \{\langle \operatorname{diag}(\mathbf{D}), \boldsymbol{\lambda} \rangle \mid \boldsymbol{\lambda} \in [0,1]^n\}. \tag{3.16}$$

Proof. Let $\boldsymbol{d} = \operatorname{diag}(\mathbf{D})$ and let $\boldsymbol{\lambda}$ denote the eigenvalues of $\mathbf{Z}$. Then the Hoffman-Wielandt inequality [Ren10, Sct. 18.3.7] states

$$\langle \mathbf{C}', \mathbf{Z} \rangle \leq \langle \boldsymbol{d}, \mathbf{P}\boldsymbol{\lambda} \rangle,$$

where $\mathbf{P}$ is a permutation that assigns the i-th biggest entry of $\boldsymbol{\lambda}$ to the i-th biggest entry of $\boldsymbol{d}$ for all $i \in [n]$. Since $\boldsymbol{\lambda} \in [0,1]^n$, the right-hand side is maximal when $\boldsymbol{\lambda}$ is the solution $\boldsymbol{\lambda}^*$ of (3.16) and $\mathbf{P}$ the identity. Then one can verify that the left hand-side also attains this upperbound by choosing $\mathbf{Z} = \mathbf{B}^\top \operatorname{Diag}(\boldsymbol{\lambda}^*)\mathbf{B}$. □

Now that we showed that solving (3.15) reduces to the computation (3.9), it is important to note that the computational burden actually lies in the preprocessing. From the point of view of complexity, we have the following.

Theorem 3.25:
The cone $\mathcal{S}_+^n$ *is suitable for* CLPN.

Proof. The complexity of computing the eigenvalues of a $n \times n$ matrix as well as matrix multiplication is contained in $\mathcal{O}(n^3)$. Since $S_+^n \subseteq \mathbb{R}^N$ with $n \in \mathcal{O}(N^{1/2})$, we get an algorithm in $\mathcal{O}(n^3) \subseteq \mathcal{O}(N^{3/2}) \subseteq \mathcal{O}(N^2)$. □

Remark 3.26:
The preceding theorem is noteworthy in terms of Remark 3.19, since the conic order $\preceq$ induced by $\mathcal{S}_+^n$ is explicitly known to *not* satisfy the lattice property. In particular, if we have a proper interval $\mathbf{L} \prec \mathbf{U}$, then Slater's condition holds and we expect strong duality to hold in Lemma 3.18, so that the preceding theorem yields an oracle for elements of the set of least upperbounds of $\{\mathbf{C}_-, -\mathbf{C}_+\}$ in $(\mathbb{R}^N, \preceq)$.

3.3.4 Experiments

In this section, we show some experiments done with a simple implementation of both MNP and CLPN. As a reference, we used the widespread SDPT3 package [TTT03].

Data generation

SOCP

Based on parameter tuples $\frac{n}{m}$, we generated random instances for $\mathcal{K} = \mathcal{L}_n$. We set $\boldsymbol{l} = \mathbf{0}$, $u_0 = 10$ and $\tilde{\boldsymbol{u}}$ to a random vector with entries in $[-0.5, 0.5]$, which was afterwards normalized such that $\|\tilde{\boldsymbol{u}}\|_2$ was a random number in the interval $[0, 10]$.

The vector $\boldsymbol{c}$ was randomly chosen with entries in $[-0.5, 0.5]$ and $\mathbf{A}$ was chosen as a random $m \times n$ matrix with entries in $[0, 1]$. To guarantee feasibility, we included the midpoint of $[\boldsymbol{l}, \boldsymbol{u}]_{\mathcal{L}_n}$ into the feasible region by setting $\boldsymbol{b} = \frac{1}{2}\mathbf{A}(\boldsymbol{u} - \boldsymbol{l})$.

SDP

Based on parameter tuples $\frac{n}{m}$, we generated random instances for $\mathcal{K} = \mathcal{S}_+^n$. We set $\mathbf{L} = \mathbf{0}$ and constructed $\mathbf{U}$ in the following way: We first constructed a random $n \times n$ matrix $\mathbf{V}$ with values in $[0, 1]$ and set $\mathbf{U} = \mathbf{V}\mathbf{V}^\top + \frac{1}{10}\mathbf{I}_n$. Afterwards, $\mathbf{U}$ was normalized such that $\operatorname{tr}(\mathbf{U}) = 10$.

The remaining parameters were chosen in the same way as for SOCP: $\mathbf{C}$ was randomly chosen with entries in $[-0.5, 0.5]$ and $\mathcal{A}$ was chosen as a random $m \times n^2$ linear

operator with entries in $[0, 1]$. To guarantee feasibility, we included the midpoint of $[\mathbf{L}, \mathbf{U}]_{\mathcal{S}^n_+}$ into the feasible region by setting $\mathbf{B} = \frac{1}{2}\mathcal{A}(\mathbf{U} - \mathbf{L})$.

Plots

In the following plots, each data point $\frac{n}{m}$ corresponds to the average of 25 instances randomly generated according to the procedure outlined before with parameters $\frac{n}{m}$. The error tolerance for CLPN was set to 10^{-6} and the error tolerance ε given in the plots apply to the subroutine MNP.

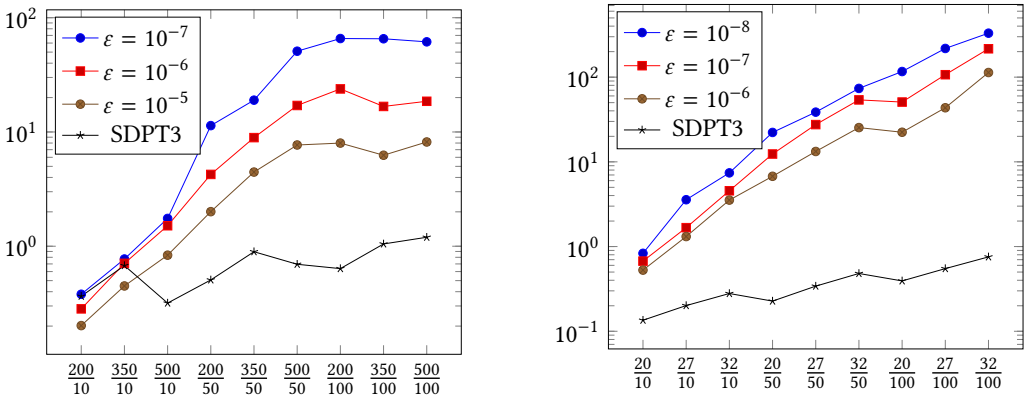

Figure 3.2: Runtime (sec) for parameters $\frac{n}{m}$. Left: SOCP. Right: SDP.

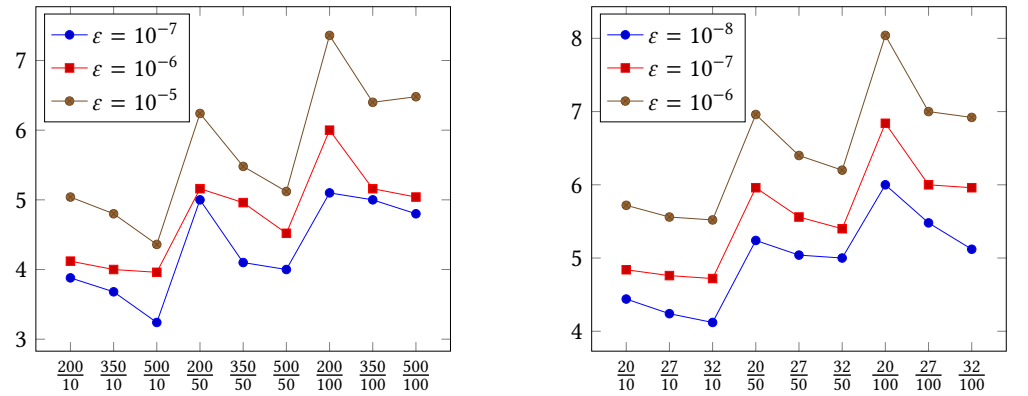

Figure 3.3: Newton-steps for parameters $\frac{n}{m}$. Left: SOCP. Right: SDP.

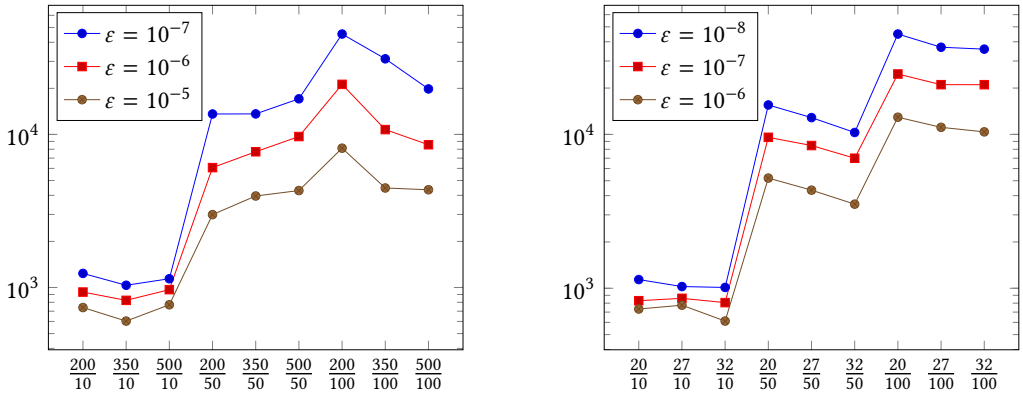

Figure 3.4: MNP computations for parameters $\frac{n}{m}$. Left: SOCP. Right: SDP.

The plots in Figure 3.2 show that the choice of accuracy for MNP has a great impact on the overal running time of the algorithm. While reducing the accuracy can speed up the algorithm significantly, going below the accuracy given in the plots often resulted in major problems in converging to the solution, so care has to be taken by choosing this parameter.

Overall, the data in Figure 3.3 resembles the results of [FHYZ08] for the case of $\mathbb{R}^n_+$, in the sense that only a few Newton-steps are necessary to get a close approximate solution. Figure 3.4 also shows that, like in the original paper, the number m of constraints seems to a have a much larger impact on the performance than the number of the variables n, since much more subproblems have to be solved.

Conclusion

Unfortunately, the experiments show that our implementation of the CLP-Newton method is much slower than the reference algorithm. However, the runtime of the algorithm could be drastically reduced by a faster algorithm for the minimum-norm-point subroutine, since the number of Newton-steps remain small independent of the underlying cones. In particular, the algorithm might be improved by a more rigorous treatment of the necessary accuracy for the subproblems, since the experiments indicate much slower progress with increasing accuracy.

Chapter 4

Partitions and Assignment Matrices

In this chapter, we formally introduce partition problems in Section 4.1 and discuss the (dis)advantages of representing them as assignment matrices in Section 4.2, where we discuss problems related to symmetries and how to solve them with the theory of orbitopes.

4.1 Overview

Partitions

For a fixed number $n \in \mathbb{N}$, a *partition* of $[n]$ is a subset $\{T_j \mid j \in [k]\} \subseteq 2^{[n]}$ such that

(i) $T_j \neq \emptyset \quad \forall j \in [k]$,

(ii) $\bigcup_{j \in [k]} T_j = [n]$,

(iii) $T_i \cap T_j = \emptyset \quad \forall i, j \in [k]$.

The sets that make up a partition are called *parts*. Throughout the thesis, we will use the shorthand notation $\mathcal{T} = \{T_j \mid j \in [k]\}$ and assume that $L \subseteq 2^{[n]}$ is an independence system.

Definition 4.1 (Set of Partitions)**:**
For $k \in [n]$, let $\mathcal{P}_k^n$ denote the set of all partitions of $[n]$ that consist of exactly k parts. Then the set

$$\mathcal{P}_k^n(L) := \{\mathcal{T} \in \mathcal{P}_k^n \mid \mathcal{T} \subseteq L\}$$

denotes the set of all partitions of $[n]$ whose k parts belong to L. More general, we let

$$\mathcal{P}^n(L) = \bigcup_{k \in [n]} \mathcal{P}_k^n(L)$$

be the set of all partitions of $[n]$ with regards to L and we shortly write $\mathcal{P}^n := \mathcal{P}^n(2^{[n]})$ as well.

Lemma 4.2 (Cardinality, [Sta11, p. 73ff.])**:**
The cardinality of $\mathcal{P}_k^n$ is given by $\left\{ {n \atop k} \right\}$, the Stirling numbers of the second kind. *They have the explicit description*

$$\left\{ {n \atop k} \right\} = \frac{1}{k!} \sum_{j \in [k]} (-1)^{k-j} \binom{k}{j} j^n,$$

and satisfy the recurrence

$$\left\{ {n \atop k} \right\} = k \cdot \left\{ {n-1 \atop k} \right\} + \left\{ {n-1 \atop k-1} \right\}$$

where $\left\{ {k \atop k} \right\} = \left\{ {n \atop 1} \right\} = 1$ for all $n, k \in \mathbb{N}$.

The cardinality of $\mathcal{P}^n$ is given by the Bell number *$B(n)$ and satisfies the formulas*

$$B(n) = \sum_{k \in [n]} S(n,k) = \sum_{i=0}^{n} \binom{n}{i} B(i),$$

where $B(0) = 1$.

k-partition Problems

Let $k \in [n]$. A function $f : \mathcal{P}_k^n(L) \to \mathbb{R}$ is called *k-partition function* and its associated optimization problem

$$\min \left\{ f(\mathcal{T}) \;\middle|\; \mathcal{T} \in \mathcal{P}_k^n(L) \right\}$$

is called a *k-partition problem*. For reference, we will denote this problem as $(f, \mathcal{P}_k^n(L))$.

A k-partition function $f_k : \mathcal{P}_k^n(L) \to \mathbb{R}$ is *separable (with regards to $f : L \to \mathbb{R}$)*, if for each $\mathcal{T} \in \mathcal{P}_k^n(L)$ we have

$$f_k(\mathcal{T}) = \sum_{j \in [k]} f(T_j),$$

and the corresponding k-partition problem is called separable as well.

Example 4.3 (Max-Cut)**:**
Consider a graph $G \in \mathcal{G}_n$ with a weighted adjacency matrix $\mathbf{C} \in \mathbb{R}^{n \times n}$ to define a function $\mathrm{cut}_{\mathbf{C}} : 2^{[n]} \to \mathbb{R}$ through

$$\mathrm{cut}_{\mathbf{C}}(T) = \sum_{i,j \in T} c_{ij} \quad \forall\, T \subseteq [n].$$

Then the famous *Max-k-Cut* problem can be stated as the separable k-partition problem $(\mathrm{cut}_k, \mathcal{P}_2^{[n]})$ where

$$\mathrm{cut}_k(\mathcal{T}) = \sum_{j \in [k]} \mathrm{cut}_{\mathbf{C}}(T_j)$$

is separable with regards to $\mathrm{cut}_{\mathbf{C}}$ as defined above.

Remark 4.4:
The problem Max-2-Cut is often simply denoted as Max-Cut in literature.

As a corollary of the preceding example, we see that partition problems are NP-hard in general.

Theorem 4.5 ([Sch03, Vol. C, Theorem 75.1])**:**
The problem Max-2-Cut *is* NP*-hard.*

Partition Problems

A function $f : \mathcal{P}^n(L) \to \mathbb{R}$ is simply called *partition function* and the associated problem

$$\min \{f(\mathcal{T}) \mid \mathcal{T} \in \mathcal{P}^n(L)\}$$

is called a *partition problem*, which will be denoted as $(f, \mathcal{P}^n(L))$.

Likewise, a partition function $f_{[n]}$ is *separable (with regards to $f : L \to \mathbb{R}$)*, if the restriction of $f_{[n]}$ to k-partitions is separable with regards to f for every $k \in [n]$, that is

$$f_{[n]}(\mathcal{T}) = \sum_{j \in [k]} f(T_j) \quad \forall \mathcal{T} \in \mathcal{P}_k^n(L)$$

holds for all $k \in [n]$.

A partition function f is called a *counting function* if it only depends on the number of parts in a partition and is strictly increasing. In particular, if f is a counting function, then there is a strictly increasing function $g : \mathbb{N} \to \mathbb{R}$ such that

$$f(\mathcal{T}) = g(k)$$

holds for all k-partitions $\mathcal{T} \in \mathcal{P}_k^n(L)$. We may abuse notation by denoting the counting function by $g(k)$ directly instead.

A partition problem whose partition function is a counting function is called a *minimum-cover problem* and has the following combinatorial interpretation.

Lemma 4.6:
Let $(g, \mathcal{P}^n(L))$ be a minimum-cover problem. Then solving $(f, \mathcal{P}^n(L))$ is equivalent to finding

$$\min \left\{ k \in \mathbb{N} \,\middle|\, \mathcal{P}_k^n(L) \neq \emptyset \right\},$$

and we can assume without loss of generality that $g = \mathrm{Id}$.

Example 4.7 (Graph Colouring)**:**
Consider a graph $G \in \mathcal{G}_n$ and denote by $S_G \subseteq 2^{[n]}$ the *stable sets* of G where

$$S_G = \{S \subseteq [n] \mid E(G[S]) = \emptyset\},$$

that is, $S \in S_G$ if and only if the induced subgraph $G[S]$ does not contain any edges. Note that this way, S_G satisfies the axioms (2.2) of an independence system.

Then the minimum-cover problem $\left(\mathrm{Id}, \mathcal{P}_k^n(S_G)\right)$ is the problem of finding the smallest partition of $[n]$ into stable sets of G, which is known as the *graph colouring problem*. In particular, the *chromatic number* $\chi(G)$ can be expressed as

$$\chi(G) = \min\left\{k \in \mathbb{N} \;\middle|\; \mathcal{P}_k^n(S_G) \neq \emptyset\right\}.$$

A more detailed treatment of the colouring problem is the topic of Section 6.3.1.

Lastly, we also consider *regularized partition functions* $f_g : \mathcal{P}^n(L) \to \mathbb{R}$ that can be written as

$$f_g = f_{[n]} + g$$

where $f_{[n]}$ is a separable partition function and g is a counting function that serves as a *regularizer*.

Example 4.8 (Regularized Euclidean clustering)**:**
Consider a set of points $\{\boldsymbol{b}_i\}_{i\in[n]} \subseteq \mathbb{R}^d$ in Euclidean space. Then we can define an optimal-value function $f\colon 2^{[n]} \to \mathbb{R}$ by setting

$$f(T) = \min_{\boldsymbol{x}\in\mathbb{R}^d} \sum_{i\in T} \|\boldsymbol{x} - \boldsymbol{b}_i\|_2^2$$

for all $T \subseteq [n]$ to get a partition function $f_{[n]}$ separable with regards to f. Now using $g(k) = k^2$ and some penalty parameter $\mu > 0$, the regularized partition function $f_{\mu\cdot g}$ corresponds to quadratically regularized *Euclidean clustering*. The treatment of Euclidean clustering problems is the topic of Section 6.3.2.

4.2 Assignment Matrices

In the following, we will describe how to fit partitions into an optimization framework by giving a standard representation of partitions as *assignment matrices*. Given a set $I \subseteq [n]$, its *characteristic vector* $\boldsymbol{e}_I \in \{0,1\}^n$ is coordinate-wise defined as

$$(\boldsymbol{e}_I)_i = \begin{cases} 1 & \text{if } i \in I, \\ 0 & \text{else.} \end{cases}$$

Geometrically, the map $I \mapsto \boldsymbol{e}_I$ bijectively maps the power set 2^n to the vertices of the n-dimensional unit cube C^n. In particular, we have

$$I \cap J = \emptyset \;\Leftrightarrow\; \langle \boldsymbol{e}_I, \boldsymbol{e}_J \rangle = 0 \tag{4.1}$$

and

$$\boldsymbol{e}_{(I\cup J)} = \boldsymbol{e}_I + \boldsymbol{e}_J - \boldsymbol{e}_{(I\cap J)}. \tag{4.2}$$

Our goal is now to express k-partitions through $n \times k$ binary matrices by treating their columns as characteristic vectors. To this end, let

$$V(L) = \{\boldsymbol{e}_I \in \{0,1\}^n \mid I \in L_*\} = \{\boldsymbol{x} \in \{0,1\}^n \mid \operatorname{supp}(\boldsymbol{x}) \in L_*\}$$

be the vertex-set associated with L and define the set of L-constrained assignment matrices

$$\mathcal{U}_{n,k}(L) = \left\{\mathbf{U} \in \{0,1\}^{n\times k} \;\middle|\; \mathbf{U}\boldsymbol{e}_{[k]} = \boldsymbol{e}_{[n]},\ \operatorname{col}(\mathbf{U}) \subseteq V(L)\right\}. \tag{4.3}$$

We will also use the notation

$$\mathcal{U}_{n,k} := \mathcal{U}_{n,k}(2^{[n]}) = \left\{\mathbf{U} \in \{0,1\}^{n\times k} \;\middle|\; \mathbf{U}\boldsymbol{e}_{[k]} = \boldsymbol{e}_{[n]},\ \mathbf{0} \notin \operatorname{col}(\mathbf{U})\right\},$$

which immediately shows the alternative expression

$$\mathcal{U}_{n,k}(L) = \mathcal{U}_{n,k} \cap \mathcal{V}_{\mathbb{R}}\left(L_*^k\right)$$

for (4.3).

By examining (4.1) and (4.2), we can verify that each matrix in $\mathcal{U}_{n,k}(L)$ can be turned into a partition by treating the columns as characteristic vectors. Unfortunately, we have multiple matrices in $\mathcal{U}_{n,k}(L)$ that yield the same partition.

Example 4.9:
Both matrices

$$\begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \text{ and } \begin{pmatrix} 0 & 1 \\ 0 & 1 \\ 1 & 0 \end{pmatrix}$$

represent the partition $\{\{1,2\},\{3\}\}$ by treating their columns as characteristic vectors.

This lack of unique representation arises since the elements of $\mathcal{U}_{n,k}(L)$ are *ordered* columns, while the elements of $\mathcal{P}_k^n(L)$ are *unordered* sets. More formally, the group $\mathfrak{S}_k$ induces a group action on $\mathcal{U}_{n,k}(L)$ by permutation of its columns. Setting

$$\pi * (\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k) = \left(\boldsymbol{u}_{\pi(1)}, \ldots, \boldsymbol{u}_{\pi(k)}\right), \quad \forall \pi \in \mathfrak{S}_k,\ \forall \left(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k\right) \in \mathcal{U}_{n,k}(L),$$

recall that the *orbit* of a matrix $\mathbf{U} \in \mathcal{U}_{n,k}(L)$ under this *group action* is the set

$$[\mathbf{U}] := \{\pi * \mathbf{U} \mid \pi \in \mathfrak{S}_k\},$$

and we expect each orbit to correspond to a partition in $\mathcal{P}_k^n(L)$. To properly see this, let us introduce an order on $\mathcal{U}_{n,k}(L)$ to define a unique representative for each orbit by choosing the maximal member with regards to that order.

Definition 4.10 (Maximal Orbit Representatives, [KP08])**:**
Define an order $\prec_{lex}$ on $\mathbb{R}^{n\times k}$ by setting $\mathbf{A} \prec_{lex} \mathbf{B}$ if and only if $a_{ij} < b_{ij}$ for the smallest (i,j) w.r.t. $<_{lex}$ such that $a_{ij} \neq b_{ij}$. Then $\mathcal{U}_{n,k}^{lex}(L)$ is defined as the set of matrices $\mathbf{U} \in \mathcal{U}_{n,k}(L)$ which are maximal with regards to $\prec_{lex}$ in their orbit $[\mathbf{U}]$.

Example 4.11:
The set $\mathcal{U}_{3,2}$ listed in ascending order of $\prec_{lex}$ is given by

$$\begin{pmatrix}0&1\\0&1\\1&0\end{pmatrix} \prec_{lex} \begin{pmatrix}0&1\\1&0\\0&1\end{pmatrix} \prec_{lex} \begin{pmatrix}0&1\\1&0\\1&0\end{pmatrix} \prec_{lex} \begin{pmatrix}1&0\\0&1\\0&1\end{pmatrix} \prec_{lex} \begin{pmatrix}1&0\\0&1\\1&0\end{pmatrix} \prec_{lex} \begin{pmatrix}1&0\\1&0\\0&1\end{pmatrix}.$$

With this definition at hand, we can state the core representation result.

Theorem 4.12:
We have

$$\mathcal{P}_k^n(L) \leftrightarrow \mathcal{U}_{n,k}^{lex}(L),$$

or rather

$$\mathcal{P}_k^n(L) \times \mathfrak{S}_k \leftrightarrow \mathcal{U}_{n,k}(L). \tag{4.4}$$

Proof. We define the map $\varphi\colon \mathcal{P}_k^n(L) \to \mathcal{U}_{n,k}^{lex}(L)$, which assigns $\mathcal{T} \in \mathcal{P}_k^n(L)$ to

$$\varphi(\mathcal{T}) =: \left(\pi_{\mathcal{T}} * (\boldsymbol{e}_{T_1}, \dots, \boldsymbol{e}_{T_k})\right) \in \mathcal{U}_{n,k}^{lex}(L),$$

where $\pi_{\mathcal{T}} \in \mathfrak{S}_k$ is chosen such that $\varphi(\mathcal{T})$ is maximal in its orbit.

This map is bijective: Its inverse takes the columns $\boldsymbol{u}_i$ of $\mathbf{U} \in \mathcal{U}_{n,k}^{lex}(L)$, extracts their support $T_i = \operatorname{supp}(\boldsymbol{u}_i)$ and outputs $\{T_1, \dots, T_k\}$. This inverse is well-defined, since by definition of $\boldsymbol{u}_i \in V(L)$, we have $T_i \in L_*$ nonempty, and since $\mathbf{U}\boldsymbol{e}_{[k]} = \boldsymbol{e}_{[n]}$, they are disjoint and satisfy $\bigcup_{j\in[k]} T_j = [n]$.

For the second part, it now suffices to argue that

$$\mathcal{U}_{n,k}^{lex}(L) \times \mathfrak{S}_k \leftrightarrow \mathcal{U}_{n,k}(L).$$

To see this, consider the map $\psi\colon \mathcal{U}_{n,k}^{lex}(L) \times \mathfrak{S}_k \to \mathcal{U}_{n,k}(L)$ given as

$$\psi(\mathbf{U}, \pi) = \pi * \mathbf{U}.$$

Since any $\mathbf{U} \in \mathcal{U}_{n,k}^{lex}(L)$ has k distinct columns, ψ is injective, and since it is surjective by definition of $\mathcal{U}_{n,k}^{lex}(L)$, the statement follows. □

Fortunately, for $k = 2$, we can describe $\mathcal{U}_{n,2}^{lex}(L)$ explicitly.

Theorem 4.13:
Let $n \in \mathbb{N}$, then

$$\mathcal{U}_{n,2}^{lex}(L) = \left\{\mathbf{U} \in \mathcal{U}_{n,2}(L) \;\middle|\; u_{11} = 1\right\}.$$

Proof. Let $\mathbf{U} = (\boldsymbol{u}_1, \boldsymbol{u}_2) \in \mathcal{U}_{n,2}(L)$ and $\mathbf{U}' = (\boldsymbol{u}_2, \boldsymbol{u}_1)$. Then the orbit of $\mathbf{U}$ is given by $[\mathbf{U}] = \{\mathbf{U}, \mathbf{U}'\}$ and by definition, $(\boldsymbol{u}_1)_1 \neq (\boldsymbol{u}_2)_1$. Thus

$$\mathbf{U}' \prec_{lex} \mathbf{U} \Leftrightarrow (\boldsymbol{u}_2)_1 < (\boldsymbol{u}_1)_1 \Leftrightarrow (\boldsymbol{u}_1)_1 = 1,\ (\boldsymbol{u}_2)_1 = 0,$$

which is what we wanted to show. □

Of course, Theorem 4.13 is not very surprising. Since we already now that the first element will necessarily be part of one of the sets, the theorem merely states that we can list this set as the first one.

4.2.1 Symmetry induced Problems

Theorem 4.12 implies that in general, it is a very bad idea to use $\mathcal{U}_{n,k}(L)$ as a modelling basis. Since we would like to optimize functions defined on $\mathcal{P}_k^n(L)$, we can state (4.4) as

$$\mathcal{P}_k^n(L) \leftrightarrow \mathcal{U}_{n,k}(L)/\mathfrak{S}_k,$$

which means that the representation of any k-partition function f in this setting must be constant on the orbits $\mathcal{U}_{n,k}(L)/\mathfrak{S}_k$. This will necessarily introduce problems in any approach to this problem involving the whole set $\mathcal{U}_{n,k}(L)$, since both the feasible and the optimal set become highly symmetrical without applying any techniques for symmetry breaking.

Since we ultimately want to tackle k-partition problems with a convex formulation, it is paramount to highlight these problems in the convex setting. To this end, we will ignore the constraints given by L for now and assume $L = 2^{[n]}$ to focus on the easiest case $\mathcal{U}_{n,k}$.

The smallest convex body containing $\mathcal{U}_{n,k}$ is naturally $\operatorname{conv}(\mathcal{U}_{n,k})$, but we will use

$$\mathcal{U}_{n,k} = \bigcup_{\mathrm{U} \in \mathcal{U}_{n,k}^{lex}} [\mathrm{U}]$$

to consider the subset $\operatorname{conv}([\mathrm{U}]) \subseteq \operatorname{conv}(\mathcal{U}_{n,k})$ for $\mathbf{U} \in \operatorname{conv}(\mathcal{U}_{n,k})$ as well. To this end, we get the following characterization.

Theorem 4.14:
For all $\mathrm{U} \in \mathcal{U}_{n,k}^{lex}$*, we have*

$$\operatorname{bar}(\operatorname{conv}(\mathcal{U}_{n,k})) = \operatorname{bar}(\operatorname{conv}([\mathrm{U}])) = \tfrac{1}{k}\mathbf{J}_{n,k}$$

where bar *denotes the* barycenter*. In particular, all convexified orbits intersect in* $\frac{1}{k}\mathbf{J}_{n,k}$*, and if* $f : \mathbb{R}^{n\times k} \to \mathbb{R}$ *is a convex function invariant under the group action of* $\mathfrak{S}_k$*, then*

$$\min\left\{f(\mathbf{X}) \,\middle|\, \mathbf{X} \in \operatorname{conv}(\mathcal{U}_{n,k})\right\} = \min\left\{f(\mathbf{X}) \mid \mathbf{X} \in \operatorname{conv}([\mathrm{U}])\right\} = f(\tfrac{1}{k}\mathbf{J}_{n,k}).$$

Proof. The barycenter of a set is necessarily invariant under its symmetries, so we consider the *Reynolds operator*

$$\varphi : \mathbf{U} \mapsto \frac{1}{k!}\sum_{\pi\in\mathfrak{S}_k} (\pi * \mathbf{U})$$

that maps any matrix $\mathbf{U} \in \mathcal{U}_{n,k}$ to the average over its orbit. In particular, this means that $\varphi(\mathbf{U}) \in \operatorname{conv}([\mathbf{U}])$. Since

$$\varphi(\mathbf{U}) = \frac{1}{k!}\left(\sum_{\pi\in\mathfrak{S}_k} \boldsymbol{u}_{\pi(1)}, \ldots, \sum_{\pi\in\mathfrak{S}_k} \boldsymbol{u}_{\pi(k)}\right) = \frac{1}{k}\left(\sum_{j\in[k]} \boldsymbol{u}_j, \ldots, \sum_{j\in[k]} \boldsymbol{u}_j\right) = \frac{1}{k}\mathbf{J}_{n,k}$$

is constant on $\mathcal{U}_{n,k}$, the point $\frac{1}{k}\mathbf{J}_{n,k}$ is necessarily the barycenter of each convexified orbit $\operatorname{conv}([\mathbf{U}])$ and $\operatorname{conv}(\mathcal{U}_{n,k})$. Furthermore, φ is also constant on the whole of $\operatorname{conv}(\mathcal{U}_{n,k})$ due to linearity.

Now, for any $\mathbf{U} \in \operatorname{conv}(\mathcal{U}_{n,k})$, the second statement follows from

$$f(\mathbf{U}) = \frac{1}{k!}\sum_{\pi\in\mathfrak{S}_k} f(\pi * \mathbf{U}) \geq f(\varphi(\mathbf{U})) = f\left(\tfrac{1}{k}\mathbf{J}_{n,k}\right),$$

where the first equation follows from the invariance of f. □

We visualize Theorem 4.14 before we look at its implications.

Example 4.15:
Consider the set

$$\mathcal{U}_{3,2} = \left\{\begin{pmatrix}0&1\\0&1\\1&0\end{pmatrix}, \begin{pmatrix}0&1\\1&0\\0&1\end{pmatrix}, \begin{pmatrix}0&1\\1&0\\1&0\end{pmatrix}, \begin{pmatrix}1&0\\0&1\\1&0\end{pmatrix}, \begin{pmatrix}1&0\\0&1\\1&0\end{pmatrix}, \begin{pmatrix}1&0\\1&0\\0&1\end{pmatrix}\right\},$$

and recall that it is listed ascending in $\prec_{lex}$. Then the corresponding set

$$\mathcal{U}_{3,2}^{lex} = \left\{\begin{pmatrix}1&0\\0&1\\0&1\end{pmatrix}, \begin{pmatrix}1&0\\0&1\\1&0\end{pmatrix}, \begin{pmatrix}1&0\\1&0\\0&1\end{pmatrix}\right\}$$

contains exactly one element of each orbit. For visualization purposes, we will use the linear projection $\rho : \mathbb{R}^{3\times 2} \to \mathbb{R}^2$ given by

$$\rho(\mathbf{U}) = \frac{1}{\sqrt{6}} \cdot \operatorname{diag}\left(\begin{pmatrix}\sqrt{3}&0&-\sqrt{3}\\-1&2&-1\end{pmatrix} \cdot \mathbf{U}\right),$$

and note that the projection is chosen in such a way that the line spanned by $\mathbf{J}_{3,2}$ is mapped to the origin. Then Figure 4.1 shows how the convex hull of each orbit intersects $\frac{1}{2}\mathbf{J}_{3,2}$, and that the convex hull of $\mathcal{U}_{3,2}^{lex}$ is much smaller due to the absence of symmetry.

Theorem 4.14 shows that if we want to work with $\operatorname{conv}(\mathcal{U}_{n,k})$ and a convex objective function f, mild assumptions on f will lead to the trivial result of $\frac{1}{k}\mathbf{J}_{n,k}$, which, interpreted as a probabilistic statement, merely states that each partition is as likely as any other to be the optimal solution to the problem at hand. This is more likely when the objective function f is nonlinear, but may also happen in the linear case, when the underlying algorithm does not guarantee an extreme point as solution.

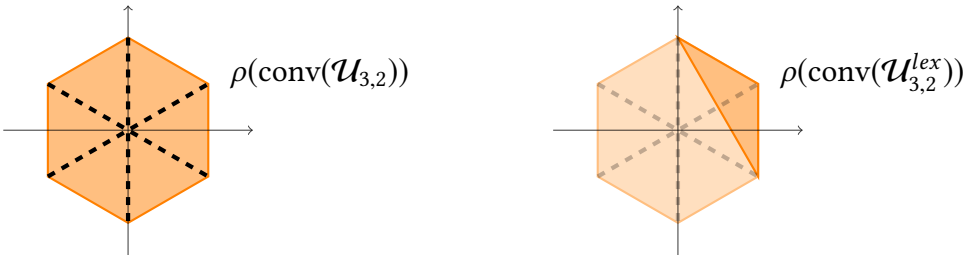

Figure 4.1: Visualization of Theorem 4.14. The shaded area shows the projection of convex hulls of $\mathcal{U}_{3,2}$ and $\mathcal{U}^{lex}_{3,2}$ respectively. The dashed lines are the projections of the convex hulls of the orbits, highlighting how they meet in the barycenter.

4.2.2 Orbitopes

Even if we work with the simplex-algorithm, which guarantees vertex solutions, Figure 4.1 still shows that it would be beneficial to rather work with the reduced polytope $\text{conv}(\mathcal{U}^{lex}_{n,k})$ instead. This is particular true for branch and bound algorithms, which may build up an extensive tree due to enumerating members of the same orbit in different branches.

To this end, Kaibel et al. used the term *partition-orbitope* for the set $\text{conv}(\mathcal{U}^{lex}_{n,k})$ and showed both an explicit description of exponential size [KP08], as well as a compactly lifted representation of linear size [FK09].

Remark 4.16:
The term *orbitope* has also been used for a related, but different construction in literature and should not be confused. In [Stu11], orbitopes are defined as convex hulls of the orbits of a compact algebraic group acting linearly on a real vector space.

This notion does not coincide with our definition, since we consider the convex hull of selected representatives from *multiple* orbits. In particular, $\mathcal{U}^{lex}_{n,k}(L)$ has a non-empty intersection with each orbitope arising from $\mathcal{U}_{n,k}(L)$ according to the second definition. This can be seen in Figure 4.1, where the dashed lines are projections of the orbitopes of the second definition.

We summarize the main result of the latter paper without getting into the details.

Theorem 4.17 ([FK09])**:**
There is an extended formulation $P_{n,k} \in \mathbb{R}^{O(nk)}$ for $\text{conv}(\mathcal{U}^{lex}_{n,k})$ such that $P_{n,k}$ is integral and can be described by $O(nk)$ inequalities. Furthermore, linear optimization over $P_{n,k}$, and, as a consequence, over $\text{conv}(\mathcal{U}^{lex}_{n,k})$ can be done in time $O(nk)$.

The idea here is that one can identify the matrices in $\mathcal{U}^{lex}_{n,k}$ with certain flows through a directed, acyclic network arising from a slightly altered $n \times k$ grid graph. In this setting, linear optimization can be reduced to a longest $s - t$ path problem and solved in linear time in the size of the grid.

As a special case of Theorem 4.17, we will explicitly show a description of the simple orbitopes $\operatorname{conv}(\mathcal{U}_{n,2}^{lex})$.

Theorem 4.18:
Let $n \in \mathbb{N}$, then

$$\operatorname{conv}(\mathcal{U}_{n,2}^{lex}) = \left\{ \begin{pmatrix} 1 & 0 \\ \boldsymbol{e} - \boldsymbol{u} & \boldsymbol{u} \end{pmatrix} \in \mathbb{R}^{n\times 2} \;\middle|\; \boldsymbol{u} \in [0,1]^{n-1},\ \langle \boldsymbol{u}, \boldsymbol{e} \rangle \geq 1 \right\}.$$

Proof. Let C denote the set on the right handed side. Then by Theorem 4.13, $\mathcal{U}_{n,2}^{lex} \subseteq C$, and since C is convex, this holds true when taking the convex hull on both sides. To see the inverse, note that this is equivalent to showing that

$$\operatorname{conv}\left(\left\{\boldsymbol{u} \in \{0,1\}^{n-1} \;\middle|\; \boldsymbol{u} \neq \boldsymbol{0}\right\}\right) = \left\{\boldsymbol{u} \in [0,1]^{n-1} \;\middle|\; \langle \boldsymbol{u}, \boldsymbol{e} \rangle \geq 1\right\},$$

by the characterization of $\mathcal{U}_{n,2}^{lex}$ in Theorem 4.13. This can be verified by showing that the vertices of both sets coincide, which follows by inspection of the facet defined by $\langle \boldsymbol{u}, \boldsymbol{e} \rangle \geq 1$. □

While the theory of orbitopes takes care of all k-partition functions that can be expressed as linear functions over $\mathcal{U}_{n,k}$, they don't help much with non-linear functions. In particular, we argue that asymptotically, as n grows for fixed k, optimizing over $\operatorname{conv}(\mathcal{U}_{n,k}^{lex})$ can only help with the first few elements. To see this, let the aggregation of $\operatorname{conv}(\mathcal{U}_{n,k}^{lex})$ be defined as

$$\mathbf{A}(n,k) := \sum_{\mathbf{U} \in \mathcal{U}_{n,k}^{lex}} \mathbf{U}.$$

Lemma 4.19:
We have the recurrence

$$\mathbf{A}(n,k) = \begin{pmatrix} k \cdot \mathbf{A}(n-1,k) \\ \left\{ {n-1 \atop k} \right\} \cdot \boldsymbol{e}_{[k]}^{\top} \end{pmatrix} + \begin{pmatrix} \mathbf{A}(n-1,k-1) & \mathbf{0} \\ \mathbf{0} & \left\{ {n-1 \atop k-1} \right\} \end{pmatrix}$$

where

$$\mathbf{A}(n,1) = \boldsymbol{e}_{[n]} \text{ and } \mathbf{A}(k,k) = \mathbf{I}_k$$

for all $k, n \in \mathbb{N}$.

Proof. Denote the right hand side by $\mathbf{B}(n,k)$ and decompose it as the sum

$$\mathbf{B}(n,k) = \sum_{j \in [k]} \sum_{\mathbf{U} \in \mathcal{U}_{n-1,k}^{lex}} \begin{pmatrix} \mathbf{U} \\ \boldsymbol{e}_j^{\top} \end{pmatrix} + \sum_{\mathbf{U}' \in \mathcal{U}_{n-1,k-1}^{lex}} \begin{pmatrix} \mathbf{U}' & \mathbf{0} \\ \mathbf{0} & 1 \end{pmatrix}.$$

We argue that each matrix in this sum belongs $\mathcal{U}_{n,k}^{lex}$ and occurs only once. To see the latter, note that both individual summations clearly involve different matrices, and we

can never have $\mathbf{U} = \begin{pmatrix}\mathbf{U}' & \mathbf{0}\end{pmatrix}$ for any pair $\mathbf{U} \in \mathcal{U}^{lex}_{n-1,k}, \mathbf{U}' \in \mathcal{U}^{lex}_{n-1,k-1}$, since the last column of $\mathbf{U}$ needs to have at least one non-zero entry to represent a k-partition.

To see membership in $\mathcal{U}^{lex}_{n,k}$, note that $\begin{pmatrix}\mathbf{U}\\ \boldsymbol{e}_j^\top\end{pmatrix}$ is maximal among its orbit w.r.t. $\prec_{lex}$ since $\mathbf{U}$ already is. For $\left(\begin{smallmatrix}\mathbf{U}' & \mathbf{0}\\ \mathbf{0} & 1\end{smallmatrix}\right)$, the same is true since $\mathbf{U}'$ is maximal among its $\mathfrak{S}_{k-1}$-orbit w.r.t. $\prec_{lex}$ and since the last column $\left(\begin{smallmatrix}\mathbf{0}\\ 1\end{smallmatrix}\right)$ is the smallest vector in this order.

We finish the proof by claiming that $\mathbf{B}(n,k)$ sums over as many matrices as $\mathbf{A}(n,k)$. Since $\mathcal{U}^{lex}_{n,k} \cong \mathcal{P}^n_k$, we necessarily have $\mathbf{A}(n,k)\boldsymbol{e}_{[k]} = \left\{ {n \atop k} \right\} \cdot \boldsymbol{e}_{[n]}$. At the same time, we have

$$\mathbf{B}(n,k)\boldsymbol{e}_{[k]} = \left(k \cdot \left\{ {n-1 \atop k} \right\} + \left\{ {n-1 \atop k-1} \right\}\right) \cdot \boldsymbol{e}_{[n]} = \left\{ {n \atop k} \right\} \cdot \boldsymbol{e}_{[n]}$$

by Lemma 4.2, which shows the claim. □

Since the barycenter of $\operatorname{conv}(\mathcal{U}_{n,k})$ was given as the average over its vertices by Theorem 4.14, we want to investigate the average of vertices of $\operatorname{conv}(\mathcal{U}^{lex}_{n,k})$, which is given by

$$\hat{\mathbf{A}}(n,k) = \mathbf{A}(n,k) / \left\{ {n \atop k} \right\}.$$

Denote with $\hat{\boldsymbol{a}}(n,k,i)$ the i-th row of $\hat{\mathbf{A}}(n,k)$. It follows from Lemma 4.19 that for $1 \leq i < n$, we have the recurrence

$$\hat{\boldsymbol{a}}(n,k,i) = \frac{k \cdot \left\{ {n-1 \atop k} \right\}}{\left\{ {n \atop k} \right\}} \cdot \hat{\boldsymbol{a}}(n-1,k,i) + \frac{\left\{ {n-1 \atop k-1} \right\}}{\left\{ {n \atop k} \right\}} \cdot \begin{pmatrix}\hat{\boldsymbol{a}}(n-1,k-1,i) & 0\end{pmatrix}. \tag{4.5}$$

While giving explicit formulas is very hard in the general case, we have the following simple theorem for the case of bi-partitions.

Theorem 4.20:
For $k = 2$ and $1 < i \leq n$, we have

$$\hat{\boldsymbol{a}}(n,2,i) = \begin{pmatrix}\frac{2^{n-2}-1}{2^{n-1}-1} & \frac{2^{n-2}}{2^{n-1}-1}\end{pmatrix}.$$

In particular,

$$\lim_{n\to\infty} \hat{\boldsymbol{a}}(n,2,i) = \frac{1}{2} \cdot \boldsymbol{e}_{[2]}.$$

Proof. We show this result by induction on n. For $n = 2$, it follows by definition that $\mathbf{A}(2,2) = \mathbf{I}_2$ and so

$$\hat{\boldsymbol{a}}(2,2,2) = \begin{pmatrix}0 & 1\end{pmatrix} = \begin{pmatrix}\frac{1-1}{2-1} & \frac{1}{2-1}\end{pmatrix}.$$

For $n > 2$, we use the fact that $\left\{ {n \atop 2} \right\} = 2^{n-1} - 1$. To see this, note that we can enumerate all 2-partitions of $[n]$ twice by enumerating the pairs $(T, [n]\backslash T)$, where $T \in 2^{[n]}\backslash\{\emptyset, [n]\}$.

Then for $1 < i < n$, (4.5) turns into

$$\begin{aligned}\hat{\boldsymbol{a}}(n,2,i) &= \frac{2 \cdot \left\{ {n-1 \atop 2} \right\}}{\left\{ {n \atop 2} \right\}} \cdot \hat{\boldsymbol{a}}(n-1,2,i) + \frac{\left\{ {n-1 \atop 1} \right\}}{\left\{ {n \atop 2} \right\}} \cdot \begin{pmatrix} \hat{\boldsymbol{a}}(n-1,1,i) & 0 \end{pmatrix} \\ &= \frac{2 \cdot (2^{n-2}-1)}{2^{n-1}-1} \cdot \begin{pmatrix} \frac{2^{n-3}-1}{2^{n-2}-1} & \frac{2^{n-3}}{2^{n-2}-1} \end{pmatrix} + \frac{1}{2^{n-1}-1} \cdot \begin{pmatrix} 1 & 0 \end{pmatrix} \\ &= \begin{pmatrix} \frac{2^{n-2}-1}{2^{n-1}-1} & \frac{2^{n-2}}{2^{n-1}-1} \end{pmatrix}.\end{aligned}$$

For $i = n$, we also have

$$\hat{\boldsymbol{a}}(n,2,i) = \frac{\left\{ {n-1 \atop 2} \right\}}{\left\{ {n \atop 2} \right\}} \boldsymbol{e}_{[2]} + \frac{\left\{ {n-1 \atop 1} \right\}}{\left\{ {n \atop 2} \right\}} \boldsymbol{e}_2 = \begin{pmatrix} \frac{2^{n-2}-1}{2^n-1} & \frac{2^{n-2}}{2^n-1} \end{pmatrix}.$$

□

In general, we can observe a similar process for $k > 2$ in the sense that

$$\left\| \hat{\boldsymbol{a}}(n,k,i) - \frac{1}{k} \boldsymbol{e}_{[k]} \right\|_2$$

is quickly decreasing both in i and in n. In particular, Figure 4.2 suggests the following conjecture.

Conjecture 4.21:
The limits

$$\lim_{n\to\infty} \operatorname{dist}\left(\operatorname{conv}(\mathcal{U}_{n,k}^{lex}), \operatorname{bar}(\operatorname{conv}(\mathcal{U}_{n,k})) \right) \leq \lim_{n\to\infty} \left\| \hat{\mathbf{A}}(n,k) - \frac{1}{k} \mathbf{J}_{n,k} \right\|_2$$

exist, are finite and grow sublinearly with k.

Unfortunately, this conjecture would imply that for fixed k and growing values of n, there is a good chance for the optimum of non-linear convex functions over $\operatorname{conv}(\mathcal{U}_{n,k}^{lex})$ to stay relatively close to $\operatorname{bar}(\operatorname{conv}(\mathcal{U}_{n,k}))$, since the distance is independent of the dimension of the underlying space. In particular, using $\operatorname{conv}(\mathcal{U}_{n,k}^{lex})$ over $\operatorname{conv}(\mathcal{U}_{n,k})$ might only lead to marginal improvements, which is why we will consider reformulation techniques to break the underlying symmetry in the following chapters.

One main obstacle in proving the conjecture is the unavailability of sharp bounds for the fractions $\left\{ {n-1 \atop k} \right\} / \left\{ {n \atop k} \right\}$ that appear in the convex combination used to update the rows in (4.5). These fractions play a major role in statistics to compute an unbiased estimation with minimum variation of certain distributions, as explained in [Ber75]. For fixed fractions $\frac{k}{n}$, we have the asymptotic behaviour

$$\frac{\left\{ {n-1 \atop k} \right\}}{\left\{ {n \atop k} \right\}} \sim \frac{\alpha(\frac{k}{n})}{n},$$

where $\alpha(x)$ is the solution of $\frac{1-\exp(-\alpha)}{\alpha} = x$ according to [Har68], which implies that

$$\lim_{n\to\infty} \frac{k\left\{ {n-1 \atop k} \right\}}{\left\{ {n \atop k} \right\}} \to 1.$$

This means that new rows $\hat{\boldsymbol{a}}(n,k,n)$ are already initialized increasingly close to $\frac{1}{k} \cdot \boldsymbol{e}_{[k]}$ and do not change much through the update (4.5), supporting the conjecture.

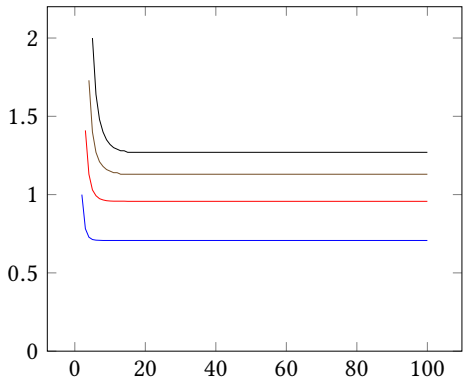

Figure 4.2: Plots of $\left\|\hat{\mathbf{A}}(n,k) - \frac{1}{k}\mathbf{J}_{n,k}\right\|_2$ against $n = k, k+1, \ldots, 100$ for increasing values of k. The plots for $k = 2, 3, 4, 5$ are shown from bottom to top respectively.

Chapter 5

Partition Matrices

This chapter treats partition matrices, a class of matrices that have been long used in combinatorial optimization for clustering problems like Max-k-cut or graph colouring. We introduce them in Section 5.1 and show how they relate to combinatorial moment matrices in Section 5.2. Finally, in Section 5.3 we show how these matrices are connected to assignment matrices through MM.

5.1 Overview

Another approach to remove symmetry from partition problems is to map the set $\mathcal{U}_{n,k}$ of assignment matrices onto the set of k-partition matrices, which are defined as follows.

Definition 5.1:
A binary matrix $\mathbf{W} \in \{0,1\}^{n\times n}$ is called *(k-)partition matrix*, if there are k integers $n_1, \dots, n_k$ and a permutation $\pi \in \mathfrak{S}_n$ such that $n = \sum_{j\in[k]} n_j$ and

$$\pi(\mathbf{W}) = \begin{pmatrix} \mathbf{J}_{n_1} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_{n_2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0} & \mathbf{J}_{n_k} \end{pmatrix}.$$

The set of all k-partition matrices will be denoted as PM_k^n.

The idea here is that partition matrices contain the information whether two elements belong to the same part of a given partition. To this end, note that for a given partition matrix $\mathbf{W}$, we have

$$w_{ij} = \begin{cases} 1 & \text{if } i \text{ and } j \text{ belong to the same set,} \\ 0 & \text{else.} \end{cases}$$

As a consequence, we have $\operatorname{diag}(\mathbf{W}) = \boldsymbol{e}$ and each block corresponds to a part of the partition.

We can explicitly describe the set of k-partition matrices with the following lemma.

Lemma 5.2 ([Ren10, Lemma 18.3])**:**
The set PM_k^n *can be explicitly described as*

$$\mathrm{PM}_k^n = \{\mathbf{W} \in \{0,1\}^{n\times n} \mid \operatorname{diag}(\mathbf{W}) = \boldsymbol{e},\ \operatorname{rank}(\mathbf{W}) = k,\ \mathbf{W} \geq \mathbf{0}\}.$$

With this representation in hand, we can now extend this concept to L-constrained k-partitions.

Definition 5.3:
The L-constrained k-partition matrices are given as

$$\mathrm{PM}_k^n(L) := \left\{\mathbf{W} \in \mathrm{PM}_k^n \;\middle|\; \operatorname{col}(\mathbf{W}) \subseteq V(L)\right\} = \mathrm{PM}_k^n \cap \mathcal{V}_{\mathbb{R}}\left(L_*^n\right).$$

By definition, we only allow columns that represent nonempty sets belonging to L. We are now able to show that there is a one to one correspondence between partitions and partition matrices.

Lemma 5.4:
We have $\mathcal{U}_{n,k}^{lex}(L) \leftrightarrow \mathrm{PM}_k^n(L)$ *via the surjection* $\varphi : \mathcal{U}_{n,k}(L) \to \mathrm{PM}_k^n(L)$ *given by*

$$\mathbf{U} \mapsto \mathbf{U}\mathbf{U}^\top.$$

Furthermore, φ *is invariant under orthogonal transformations and constant on the orbits contained in* $\mathcal{U}_{n,k}(L)/\mathfrak{S}_k$.

Proof. Let $\mathbf{U} = (\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k)$ and $\varphi(\mathbf{U}) = \mathbf{W} = (\boldsymbol{w}_1, \ldots, \boldsymbol{w}_n)$. Since the rows of $\mathbf{U}$ are binary unit vectors, we immediately get $\mathbf{W} \in \{0,1\}^{n\times n}$ and $\operatorname{diag}(\mathbf{W}) = \boldsymbol{e}$. Thus $\mathbf{W} \in \mathrm{PM}_k^n$, since $\operatorname{rank}(\mathbf{W}) = k$ and $\mathbf{W} \geq \mathbf{0}$ follow by definition from $\mathbf{W} = \mathbf{U}\mathbf{U}^\top$ and the pairwise orthogonality of the columns of $\mathbf{U}$. To see that $\mathbf{W} \in \mathrm{PM}_k^n(L)$, note that $\boldsymbol{e}_i^\top \mathbf{U} = \boldsymbol{e}_j^\top$ for some $j \in [k]$ and thus

$$\mathbf{W}\boldsymbol{e}_i = \mathbf{U}\mathbf{U}^\top \boldsymbol{e}_i = \mathbf{U}\boldsymbol{e}_j \in V(L) \tag{5.1}$$

by assumption.

To see the invariance of φ under orthogonal transformations $\mathbf{Q}$, note that

$$\varphi(\mathbf{U}\mathbf{Q}) = \mathbf{U}\mathbf{Q}\mathbf{Q}^\top\mathbf{U}^\top = \mathbf{U}\mathbf{U}^\top,$$

which also shows that φ is constant on the orbits induced by column-permutations.

Injectivity of φ on $\mathcal{U}_{n,k}^{lex}(L)$ follows by noting that the sets

$$\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k\} = \{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_n\}$$

coincide due to (5.1) and $\mathbf{0} \notin \operatorname{col}(\mathbf{U})$.

To see surjectivity of φ, let $\mathbf{W} \in \mathrm{PM}_k^n(L)$ and consider the Gram representation $\mathbf{W} = \mathbf{U}_0\mathbf{U}_0^\top$ with $\mathbf{U}_0 \in \mathbb{R}^{n\times k}$ and $\operatorname{rank}(\mathbf{U}_0) = k$. Then there is an invertible $k \times k$ submatrix $\mathbf{M}$ and we assume without loss of generality that

$$\mathbf{U}_0 = \begin{pmatrix} \mathbf{M} \\ \mathbf{U}_0' \end{pmatrix}.$$

Then $\operatorname{rank}(\mathbf{M}\mathbf{M}^\top) = \operatorname{rank}(\mathbf{M}) = k$, and $\mathbf{M}\mathbf{M}^\top$ is binary with $\operatorname{diag}(\mathbf{M}\mathbf{M}^\top) = \boldsymbol{e}$ as a submatrix of $\mathbf{W}$. Then $\mathbf{M}\mathbf{M}^\top = \mathbf{I}_k = \mathbf{M}^\top\mathbf{M}$ is orthogonal and

$$\mathbf{U}_1 := \begin{pmatrix} \mathbf{I}_k \\ \mathbf{U}_1' \end{pmatrix} = \begin{pmatrix} \mathbf{M}\mathbf{M}^\top \\ \mathbf{U}_0'\mathbf{M}^\top \end{pmatrix} = \mathbf{U}_0\mathbf{M}^\top$$

shows that $\mathbf{U}_1\mathbf{U}_1^\top = \mathbf{U}_0\mathbf{M}\mathbf{M}^\top\mathbf{U}_0^\top = \mathbf{W}$ where $\mathbf{U}_1$ contains $\mathbf{I}_k$ as submatrix where $\mathbf{M}$ used to be. This shows $\mathbf{U}_1$ is binary, since

$$\begin{pmatrix} \mathbf{I}_k & \mathbf{U}_1'^\top \\ \mathbf{U}_1' & \mathbf{U}_1'\mathbf{U}_1'^\top \end{pmatrix} = \begin{pmatrix} \mathbf{I}_k \\ \mathbf{U}_1' \end{pmatrix}\begin{pmatrix} \mathbf{I}_k \\ \mathbf{U}_1' \end{pmatrix}^\top = \mathbf{U}_1\mathbf{U}_1^\top = \mathbf{W} \in \{0,1\}^{n\times n}.$$

Now (5.1) shows both $\operatorname{col}(\mathbf{U}_1) \subseteq V(L)$ as well as the last equation in

$$[n] = \operatorname{supp}(\operatorname{diag}(\mathbf{W})) \subseteq \operatorname{supp}(\mathbf{W}\boldsymbol{e}) = \operatorname{supp}(\mathbf{U}_1\boldsymbol{e}) \subseteq [n],$$

which implies $\mathbf{U}_1\boldsymbol{e} = \boldsymbol{e}$. Then $\mathbf{U}_1 \in \mathcal{U}_{n,k}(L)$ and we are done. □

As a consequence, we get a nice characterization of the set of partitions $\mathcal{P}^n(L)$ as

$$\mathrm{PM}^n(L) = \bigcup_{k\in[n]} \mathrm{PM}_k^n(L) = \left\{\mathbf{W} \in \{0,1\}^{n\times n} \;\middle|\; \operatorname{diag}(\mathbf{W}) = \boldsymbol{e},\ \mathbf{W} \succeq \mathbf{0},\ \operatorname{col}(\mathbf{W}) \subseteq V(L)\right\}.$$

5.2 Connection to Combinatorial Moment Matrices

At this point, the description of PM_k^n is difficult to work with due to both the binary constraint as well as the rank constraint, which both lead to NP-hardness in general. Since Theorem 4.5 showed that Max-Cut is NP-hard, so is optimizing over PM_k^n, since a Max-Cut instance with objective $\mathbf{C}$ is equivalent to solving

$$\max\left\{\langle \mathbf{C}, \mathbf{W}\rangle \;\middle|\; \mathbf{W} \in \mathrm{PM}_2^n\right\}.$$

As such, we can not expect to remove both of these constraints, although it is possible to remove the rank constraint as shown in the following lemma.

Lemma 5.5 ([Ren10, Lemma 18.4])**:**
The set of k-partition matrices has the explicit description

$$\begin{aligned}\mathrm{PM}_k^n &= \{\mathbf{W} \in \{0,1\}^{n\times n} \mid \operatorname{diag}(\mathbf{W}) = \boldsymbol{e},\ (t\mathbf{W} \succeq J \Leftrightarrow t \geq k)\} \\ &= \left\{\mathbf{W} \in \{0,1\}^{n\times n} \,\middle|\, \operatorname{diag}(\mathbf{W}) = \boldsymbol{e},\ \left[\begin{pmatrix} t & \boldsymbol{e}^\top \\ \boldsymbol{e} & \mathbf{W}\end{pmatrix} \succeq \mathbf{0} \Leftrightarrow t \geq k\right]\right\}. \end{aligned} \tag{5.2}$$

This lemma is important, since it highlights a link to the theory of combinatorial moment matrices from Subsection 3.1. Lemma 5.4 shows that starting from an assignment matrix $\mathbf{U}$, we can get a partition matrix by setting

$$\mathbf{W} = \mathbf{U}\mathbf{U}^\top = \sum_{j\in[k]} \boldsymbol{u}_j \boldsymbol{u}_j^\top,$$

where each $\boldsymbol{u}_j$ is the characteristic vector of a part of the partition. By adding a new entry 1, we can consider $\begin{pmatrix}1 & \boldsymbol{u}_j^\top\end{pmatrix}$ as a truncated moment sequence up to degree 1, and the collection of these sequences yields an extended assignment matrix

$$\begin{pmatrix}\boldsymbol{e}^\top \\ \mathbf{U}\end{pmatrix} \in \{0,1\}^{(n+1)\times k}$$

which factors into an extended partition matrix as

$$\begin{pmatrix} k & \boldsymbol{e}^\top \\ \boldsymbol{e} & \mathbf{W}\end{pmatrix} = \begin{pmatrix} k & \boldsymbol{e}^\top\mathbf{U}^\top \\ \mathbf{U}\boldsymbol{e} & \mathbf{U}\mathbf{U}^\top\end{pmatrix} = \begin{pmatrix}\boldsymbol{e}^\top \\ \mathbf{U}\end{pmatrix}\begin{pmatrix}\boldsymbol{e} & \mathbf{U}^\top\end{pmatrix} = \sum_{j\in[k]} \begin{pmatrix}1 & \boldsymbol{u}_j^\top \\ \boldsymbol{u}_j & \boldsymbol{u}_j\boldsymbol{u}_j^\top\end{pmatrix}.$$

The important observation here is that an extended k-partition matrix is equal to the sum of the k first-order combinatorial moment matrices corresponding to the individual sets in its k-partition.

Example 5.6:
Consider again the partition $\{\{1,2\},\{3\}\} \in \mathcal{P}_2^3$ from Example 4.9. Lexicographic order of the corresponding characteristic vectors leads to the assignment and partition matrices

$$\mathbf{U} = \begin{pmatrix}1 & 0\\ 1 & 0\\ 0 & 1\end{pmatrix}, \qquad \mathbf{W} = \begin{pmatrix}1 & 1 & 0\\ 1 & 1 & 0\\ 0 & 0 & 1\end{pmatrix}.$$

The extended partition matrix can be decomposed as sum of first-order combinatorial moment matrices as

$$\left(\begin{array}{c|ccc} 2 & 1 & 1 & 1\\ \hline 1 & 1 & 1 & 0\\ 1 & 1 & 1 & 0\\ 1 & 0 & 0 & 1\end{array}\right) = \left(\begin{array}{c|ccc} 1 & 1 & 1 & 0\\ \hline 1 & 1 & 1 & 0\\ 1 & 1 & 1 & 0\\ 0 & 0 & 0 & 0\end{array}\right) + \left(\begin{array}{c|ccc} 1 & 0 & 0 & 1\\ \hline 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 1 & 0 & 0 & 1\end{array}\right).$$

5.3 Convexification

In the end, we are interested in solving optimization problems over $\mathrm{PM}_k^n(L)$, so we are naturally interested in its convex hull $\mathrm{conv}(\mathrm{PM}_k^n(L))$. However, as we have already seen, Max-k-Cut can be described as

$$\max\left\{\langle \mathbf{C}, \mathbf{X}\rangle \;\middle|\; \mathbf{X} \in \mathrm{PM}_k^n\right\},$$

which is equivalent to

$$\max\left\{\langle \mathbf{C}, \mathbf{X}\rangle \;\middle|\; \mathbf{X} \in \mathrm{conv}(\mathrm{PM}_k^n)\right\},$$

due to the linearity of the objective. Consequently, we should not expect to find a compact description of $\mathrm{conv}(\mathrm{PM}_k^n(L))$, as it is likely out of reach.

In order to tackle this problem anyway, it is easiest to work with convex relaxations of $\mathrm{conv}(\mathrm{PM}_k^n)$. The standard convex relaxation of the set PM_k^n in literature arises from formulation (5.2) by relaxing the integrality constraint $\mathbf{W} \in \{0,1\}^{n\times n}$ away. This leads us to the set

$$\mathrm{RPM}_k^n(L) = \left\{\mathbf{W} \in \mathbb{R}^{n\times n} \;\middle|\; \mathrm{diag}(\mathbf{W}) = \boldsymbol{e},\ \begin{pmatrix} k & \boldsymbol{e}^\top \\ \boldsymbol{e} & \mathbf{W}\end{pmatrix} \succeq \mathbf{0},\ \langle \mathbf{W}, \Omega_L\rangle = 0\right\}, \tag{5.3}$$

which is a convex relaxation of $\mathrm{PM}_k^n(L)$. The constraint $\langle \mathbf{W}, \Omega_L\rangle = 0$ arises as a truncation of $\mathrm{col}(\mathbf{W}) \subseteq V(L)$ and only excludes the sets of size 2 that are forbidden by L.

5.3.1 Applying the Method of Moments to Partition Matrices

A way to interpret Lemma 5.4 is to consider the set $\mathrm{PM}_k^n(L)$ as the image of the second order moments of $\mathcal{U}_{n,k}(L)$ under a linear map. It follows then, since $\mathcal{N}_1^*(\mathcal{U}_{n,k}(L))$ is a convex relaxation of the second order moments of $\mathcal{U}_{n,k}(L)$, that applying the same linear map to $\mathcal{N}_1^*(\mathcal{U}_{n,k}(L))$ will lead us to a convex relaxation of $\mathrm{PM}_k^n(L)$ as well. The goal of this section is to show that surprisingly, this approach yields a relaxation that is equivalent to the straightforward relaxation $\mathrm{RPM}_k^n(L)$ from (5.3).

To see this, first recall the set of assignment matrices

$$\mathcal{U}_{n,k}(L) = \left\{\mathbf{U} \in \{0,1\}^{n\times k} \;\middle|\; \mathbf{U}\boldsymbol{e}_{[k]} = \boldsymbol{e}_{[n]},\ \mathrm{col}(\mathbf{U}) \subseteq V(L)\right\}$$

from (4.3). In order to state the corresponding relaxation of $\mathrm{PM}_k^n(L)$, we will first investigate the structure of $\mathcal{N}_1^*(\mathcal{U}_{n,k}(L))$, where any assignment matrix $\mathbf{U} = (\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k)$ is represented as

$$\begin{pmatrix} 1 \\ \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \\ \vdots \\ \boldsymbol{u}_k \end{pmatrix}\begin{pmatrix} 1 \\ \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \\ \vdots \\ \boldsymbol{u}_k \end{pmatrix}^\top = \begin{pmatrix} 1 & \boldsymbol{u}_1^\top & \boldsymbol{u}_2^\top & \ldots & \boldsymbol{u}_k^\top \\ \boldsymbol{u}_1 & \boldsymbol{u}_1\boldsymbol{u}_1^\top & \boldsymbol{u}_1\boldsymbol{u}_2^\top & \ldots & \boldsymbol{u}_1\boldsymbol{u}_k^\top \\ \boldsymbol{u}_2 & \boldsymbol{u}_2\boldsymbol{u}_1^\top & \boldsymbol{u}_2\boldsymbol{u}_2^\top & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \boldsymbol{u}_{k-1}\boldsymbol{u}_k^\top \\ \boldsymbol{u}_k & \boldsymbol{u}_k\boldsymbol{u}_1^\top & \ldots & \boldsymbol{u}_k\boldsymbol{u}_{k-1}^\top & \boldsymbol{u}_k\boldsymbol{u}_k^\top \end{pmatrix}. \tag{5.4}$$

For the rest of this section, let us denote matrices with this block structure as

$$\mathbf{U}(n,k) := \begin{pmatrix} 1 & \boldsymbol{u}_1^\top & \boldsymbol{u}_2^\top & \dots & \boldsymbol{u}_k^\top \\ \boldsymbol{u}_1 & \mathbf{U}_{11} & \mathbf{U}_{12} & \dots & \mathbf{U}_{1k} \\ \boldsymbol{u}_2 & \mathbf{U}_{21} & \mathbf{U}_{22} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \mathbf{U}_{(k-1)k} \\ \boldsymbol{u}_k & \mathbf{U}_{k1} & \dots & \mathbf{U}_{k(k-1)} & \mathbf{U}_{kk} \end{pmatrix} \in \mathbb{R}^{(1+nk)\times(1+nk)}. \tag{5.5}$$

In order to describe the set $\mathcal{N}_1^*(\mathcal{U}_{n,k}(L))$, we use the representation

$$\mathcal{U}_{n,k}(L) = \mathcal{U}_{n,k} \cap \mathcal{V}_{\mathbb{R}}\left(L^k\right)$$

to work with

$$\mathcal{N}_1^*(\mathcal{U}_{n,k}(L)) = \mathcal{N}_1^*(\mathcal{U}_{n,k}) \cap \mathcal{N}_1^*(\mathcal{I}(L^k)).$$

Then $\mathcal{N}_1^*(\mathcal{U}_{n,k})$ can be explicitly described as

$$\mathcal{N}_1^*(\mathcal{U}_{n,k}) = \left\{ \mathbf{U}(n,k) \succeq \mathbf{0} \;\middle|\; \begin{array}{llll} \sum_{j\in[k]} \boldsymbol{u}_j & = \boldsymbol{e}, & \langle \boldsymbol{u}_i, \boldsymbol{e}\rangle & \geq 1 \\ \sum_{j\in[k]} \mathbf{U}_{ij} & = \boldsymbol{u}_i \boldsymbol{e}^\top, & \operatorname{diag}(\mathbf{U}_{ii}) & = \boldsymbol{u}_i \end{array} \; \forall i \in [k] \right\}. \tag{5.6}$$

Furthermore, it follows from Example 2.13 that

$$\mathcal{N}_1^*(\mathcal{I}(L^k)) = \{\mathbf{U}(n,k) \succeq \mathbf{0} \mid \langle \mathbf{U}_{ii}, \Omega_L \rangle = 0 \ \forall i \in [k]\}, \tag{5.7}$$

whose block structure repeats with k.

Now if $\mathbf{U}(n,k)$ arises from a matrix $\mathbf{U} \in \mathcal{U}_{n,k}(L)$, then the linear map φ_2 given by

$$\varphi_2 : \mathbf{U}(n,k) \mapsto \sum_{j\in[k]} \mathbf{U}_{jj}$$

yields the same result as φ from Lemma 5.4, since under these circumstances,

$$\varphi_2(\mathbf{U}(n,k)) = \sum_{j\in[k]} \mathbf{U}_{jj} = \sum_{j\in[k]} \boldsymbol{u}_j \boldsymbol{u}_j^\top = \mathbf{U}\mathbf{U}^\top = \varphi(\mathbf{U}).$$

As a consequence, we get a convex relaxation of $\mathrm{PM}_k^n(L)$ by noting that

$$\mathrm{PM}_k^n(L) \subseteq \varphi_2(\mathcal{N}_1^*(\mathcal{U}_{n,k}(L))).$$

The rest of this section is dedicated to show Theorem 5.8, which will state that already

$$\varphi_2(\mathcal{N}_1^*(\mathcal{U}_{n,k}(L))) = \mathrm{RPM}_k^n(L).$$

Due to the symmetry, we can construct multiple valid representations of (5.4) by permuting the columns of the assignment matrix $\mathbf{U}$ before the construction. To see this, let the group $\mathfrak{S}_k$ act on $\mathbb{R}^{(1+nk)\times(1+nk)}$ by setting

$$\pi * \mathbf{U}(n,k) = \begin{pmatrix} 1 & \boldsymbol{u}_{\pi(1)}^\top & \dots & \boldsymbol{u}_{\pi(k)}^\top \\ \boldsymbol{u}_{\pi(1)} & \mathbf{U}_{\pi(1)\pi(1)} & \dots & \mathbf{U}_{\pi(1)\pi(k)} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{u}_{\pi(k)} & \mathbf{U}_{\pi(k)\pi(1)} & \dots & \mathbf{U}_{\pi(k)\pi(k)} \end{pmatrix}$$

for each $\pi \in \mathfrak{S}_k$ and each $\mathbf{U}(n,k) \in \mathbb{R}^{(1+nk)\times(1+nk)}$. It then follows from the symmetrical nature of $\mathcal{U}_{n,k}(L)$ that whenever $\mathbf{U}(n,k)$ arises from $\mathbf{U} \in \mathcal{U}_{n,k}(L)$ as in (5.4), the matrix $\pi * \mathbf{U}(n,k)$ arises from $\pi(\mathbf{U})$. This naturally leads to $k!$ distinct representations $\pi * \mathbf{U}(n,k)$ of *each* partition, which makes the set $\mathcal{N}_1^*(\mathcal{U}_{n,k}(L))$ difficult to work with.

However, we can reduce it to its symmetry invariant subspace. To this end, let us introduce the linear *Reynolds operator* $\psi\colon \mathbb{R}^{(1+nk)\times(1+nk)} \to \mathbb{R}^{(1+nk)\times(1+nk)}$ given as

$$\psi(\mathbf{U}(n,k)) := \frac{1}{k!}\sum_{\pi\in\mathfrak{S}_k} (\pi * \mathbf{U}(n,k)) = \begin{pmatrix} 1 & \boldsymbol{a}^\top & \dots & \dots & \boldsymbol{a}^\top \\ \boldsymbol{a} & \mathbf{A} & \mathbf{B} & \dots & \mathbf{B} \\ \vdots & \mathbf{B} & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \mathbf{B} \\ \boldsymbol{a} & \mathbf{B} & \dots & \mathbf{B} & \mathbf{A} \end{pmatrix} =: \mathbf{U}(\mathbf{A},\mathbf{B},\boldsymbol{a}),$$

where

$$\begin{aligned} \boldsymbol{a} &= \frac{1}{k!}\sum_{\pi\in\mathfrak{S}_k} \boldsymbol{u}_{\pi(i)} &&= \frac{1}{k}\sum_{j\in[k]} \boldsymbol{u}_j && \forall i \in [k], \\ \mathbf{A} &= \frac{1}{k!}\sum_{\pi\in\mathfrak{S}_k} \mathbf{U}_{\pi(j)\pi(j)} &&= \frac{1}{k}\sum_{j\in[k]} \mathbf{U}_{jj} && \forall i \in [k], \\ \mathbf{B} &= \frac{1}{k!}\sum_{\pi\in\mathfrak{S}_k} \mathbf{U}_{\pi(i)\pi(j)} &&= \frac{1}{k(k-1)}\sum_{i,j\in[k],i\neq j} \mathbf{U}_{\pi(i)\pi(j)} && \forall i,j \in [k]. \end{aligned}$$

For later, note that we get the identity

$$\varphi_2(\mathbf{U}(n,k)) = \sum_{j\in[k]} \mathbf{U}_{jj} = k\mathbf{A} = \varphi_2(\mathbf{U}(\mathbf{A},\mathbf{B},\boldsymbol{a})), \tag{5.8}$$

which shows that $\varphi_2 \circ \psi = \varphi_2$ as a side result.

Denoting the linear subspace given by the image of ψ by $H_\psi := \varphi\big(\mathbb{R}^{(1+kn)\times(1+kn)}\big)$, it has the explicit description

$$H_\psi = \left\{ \mathbf{U}(\mathbf{A},\mathbf{B},\boldsymbol{a}) \;\middle|\; \mathbf{A},\mathbf{B} \in \mathbb{R}^{n\times n}, \boldsymbol{a} \in \mathbb{R}^n \right\},$$

as

$$\pi * \mathbf{U}(\mathbf{A}, \mathbf{B}, \boldsymbol{a}) = \mathbf{U}(\mathbf{A}, \mathbf{B}, \boldsymbol{a})$$

holds for all $\pi \in \mathfrak{S}_k$ and all $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n\times n}, \boldsymbol{a} \in \mathbb{R}^n$. The intersection $H_\psi \cap \mathcal{S}_+^{(1+nk)}$ is particularly nice, as is shown by the following lemma.

Lemma 5.7 ([GL08, Lemma 2.8])**:**
Let $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n\times n}$ *and*

$$\mathbf{X} = \mathbf{I}_k \otimes \mathbf{A} + \mathbf{J}_k \otimes \mathbf{B} = \begin{pmatrix} \mathbf{A}+\mathbf{B} & \mathbf{B} & \dots & \mathbf{B} \\ \mathbf{B} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{B} \\ \mathbf{B} & \dots & \mathbf{B} & \mathbf{A}+\mathbf{B} \end{pmatrix} \in \mathbb{R}^{kn\times kn}.$$

Then $\mathbf{X} \succeq \mathbf{0}$ *if and only if* $\mathbf{A} \succeq \mathbf{0}$ *and* $\mathbf{A} + k\mathbf{B} \succeq \mathbf{0}$.

We are now ready to show the central result of this section.

Theorem 5.8:
It holds that

$$\varphi_2(\mathcal{N}_1^*(\mathcal{U}_{n,k}(L))) = \mathrm{RPM}_k^n(L).$$

Proof. We first argue that

$$\psi(\mathcal{N}_1^*(\mathcal{U}_{n,k}(L))) = \mathcal{N}_1^*(\mathcal{U}_{n,k}(L)) \cap H_\psi.$$

For " $\subseteq$ ", note that $\mathcal{N}_1^*(\mathcal{U}_{n,k}(L))$ is convex and that by definition, ψ maps matrices $\mathbf{U}(n,k) \in \mathcal{N}_1^*(\mathcal{U}_{n,k}(L))$ into the convex hull of their orbits, so

$$\psi(\mathcal{N}_1^*(\mathcal{U}_{n,k}(L))) \subseteq \mathcal{N}_1^*(\mathcal{U}_{n,k}(L)).$$

For " $\supseteq$ ", note that ψ acts as the identity on H_ψ, and so $\mathbf{U}(n,k) \in \mathcal{N}_1^*(\mathcal{U}_{n,k}(L)) \cap H_\psi$ implies $\mathbf{U}(n,k) = \psi(\mathbf{U}(n,k)) \in \psi(\mathcal{N}_1^*(\mathcal{U}_{n,k}(L)))$.

It follows then from (5.6) that

$$\psi(\mathcal{N}_1^*(\mathcal{U}_{n,k})) = \left\{ \mathbf{U}(\mathbf{A}, \mathbf{B}, \tfrac{1}{k}\boldsymbol{e}) \succeq \mathbf{0} \;\middle|\; \mathbf{A} + (k-1)\mathbf{B} = \tfrac{1}{k}\mathbf{J}_n,\ \mathrm{diag}(\mathbf{A}) = \tfrac{1}{k}\boldsymbol{e} \right\}.$$

Using the Schur complement Lemma 2.5 on $\mathbf{U}(\mathbf{A}, \mathbf{B}, \frac{1}{k}\boldsymbol{e})$ shows

$$\mathbf{U}(\mathbf{A}, \mathbf{B}, \tfrac{1}{k}\boldsymbol{e}) \succeq \mathbf{0} \;\Leftrightarrow\; \mathbf{I}_k \otimes (\mathbf{A} - \mathbf{B}) + \mathbf{J}_k \otimes (\mathbf{B} - \tfrac{1}{k^2}\mathbf{J}_n) \succeq \mathbf{0},$$

which is equivalent to

$$\mathbf{A} - \mathbf{B} \succeq \mathbf{0},\ (\mathbf{A} - \mathbf{B}) + k(\mathbf{B} - \tfrac{1}{k^2}\mathbf{J}_n) \succeq \mathbf{0}$$

by Lemma 5.7. Since $\mathbf{A}+(k-1)\mathbf{B} = \frac{1}{k}\mathbf{J}_n$ holds by assumption, the second psd. condition is satisfied and solving for $\mathbf{B}$ yields

$$\mathbf{B}(\mathbf{A}) := \frac{1}{k-1}\left(\frac{1}{k}\mathbf{J}_n - \mathbf{A}\right),$$

which can be substituted into $\mathbf{A} - \mathbf{B} \succeq \mathbf{0}$ to yield $\mathbf{A} \succeq \frac{1}{k^2}\mathbf{J}_n$ after rescaling.

We can thus simplify

$$\begin{aligned}\psi(\mathcal{N}_1^*(\mathcal{U}_{n,k})) &= \left\{\mathrm{U}\left(\mathbf{A}, \mathbf{B}(\mathbf{A}), \tfrac{1}{k}\boldsymbol{e}\right) \;\middle|\; \begin{pmatrix} 1 & \frac{1}{k}\boldsymbol{e}^\top \\ \frac{1}{k}\boldsymbol{e} & \mathbf{A}\end{pmatrix} \succeq \mathbf{0},\ \operatorname{diag}(\mathbf{A}) = \frac{1}{k}\boldsymbol{e}\right\} \\ &= \left\{\mathrm{U}\left(\mathbf{A}, \mathbf{B}(\mathbf{A}), \tfrac{1}{k}\boldsymbol{e}\right) \;\middle|\; k\mathbf{A} \in \mathrm{RPM}_k^n\right\},\end{aligned}$$

and (5.8) shows the claim for $L = 2^{[n]}$.

To finish the proof, we need to show that this structural result is unchanged by introducing a non-trivial L. But this follows by definition, since adding an independence system L only introduces the constraint $\langle \mathbf{A}, \Omega_L \rangle = 0$ in both sets, as can be seen by comparing (5.3) and (5.7). □

This shows that one way to understand the approximation quality of $\mathrm{RPM}_k^n(L)$ is the fact that it reduces to computing the first stage of MM for $\mathcal{U}_{n,k}(L)$.

5.3.2 Applying the Method of Moments to Orbitopes

As we have seen in the preceding section in Theorem 5.8, the symmetry of $\mathcal{U}_{n,k}(L)$ leads to a compact formulation of the first stage of MM, and in turn to the compact relaxation $\mathrm{RPM}_k^n(L)$ for $\mathrm{PM}_k^n(L)$. While this is good in terms of computation, the question is whether this also leads to a decrease in quality of the solutions, as is often the case with relaxations defined on the symmetry-invariant subspace. To this end, the next logical step is to apply MM to the symmetry-free $\mathcal{U}_{n,k}^{lex}(L)$ instead, and ask if the image under φ_2 leads to tighter relaxations than $\mathrm{RPM}_k^n(L)$.

In general, this would be an ambitious approach, since applying MM to the extended formulation of $\mathcal{U}_{n,k}^{lex}(L)$ does not lead to sets with intuitive descriptions. Fortunately, we can explicitly analyse this construction for the case of $k = 2$, due to the simplified description of $\mathcal{U}_{n,2}^{lex}(L)$ in Theorem 4.13.

Remark 5.9:
Although $k = 2$ may seem limiting at first, it can actually be considered as the hardest setting by noting that good approximation ratios for Max-k-Cut are available for growing k through the following, simple heuristic:

Choosing a partition matrix $\mathbf{W} \in \mathrm{PM}_k^n$ uniformly at random, the probability of $w_{ij} = 1$ equals $\frac{1}{k}$ for any pair $i \neq j \in [n]$. This translates to an expected approximation

ratio of $1 - \frac{1}{k}$ for Max-k-Cut given as

$$\max\left\{\langle \mathbf{C}, \mathbf{W}\rangle \;\middle|\; \mathbf{W} \in \mathrm{PM}_k^n\right\}, \tag{5.9}$$

which is strictly increasing for growing k.

Before we go on, it is important to point out that in the case of $L \neq 2^{[n]}$, we can simplify the problem drastically.

Lemma 5.10:
Let $L \neq 2^{[n]}$, $G_L \in \mathcal{G}_n$ be the graph whose adjacency matrix is given by Ω_L and let $m = |\,\mathrm{CC}(G_L)|$ be the number of its connected components. Then

$$\mathcal{U}_{n,2}(L) \leftrightarrow \mathcal{U}_{m,2}^{\mathbf{0}} := \left\{\mathbf{U} \in \{0,1\}^{m\times 2} \;\middle|\; \mathbf{U}\boldsymbol{e}_{[2]} = \boldsymbol{e}_{[m]}\right\}.$$

Proof. By the central assumption that $\mathcal{P}_2^n(L) \neq \emptyset$, G_L is bipartite and can be decomposed into the union of its bipartite connected components. Then for $j \in [m]$, the j-th connected component has a unique partition $\{T_1^j, T_2^j\}$ with $T_1^j \neq \emptyset$, which can by found by the following well-known algorithm:

Starting with any node i of the component, assume $i \in T_1^j$ to conclude for its neighbourhood $N(i)$ that $N(i) \subseteq T_2^j$. But then $N(N(i)) \subseteq T_1^j$, and iterating this process at most n steps, we cover all nodes of the component and arrive at a full list for $\{T_1^j, T_2^j\}$.

Now any element of $\{T_1, T_2\} \in \mathcal{P}_2^n(L)$ is completely described by an assignment

$$\left\{T_1^j \;\middle|\; j \in [m]\right\} \to \{T_1, T_2\},$$

since for $\{a, b\} = [2]$, we have $T_1^j \subseteq T_a$ if and only if $T_2^j \subseteq T_b$. Thus $\mathcal{U}_{n,2}(L)$ can be represented by all $m \times 2$ assignment matrices, where we allow for $\mathbf{0}$ columns. □

Recall that according to (5.6) and (5.7), the first stage of MM for $\mathcal{U}_{n,2}(L)$ is

$$\mathcal{N}_1^*(\mathcal{U}_{n,2}) = \left\{\mathbf{U}(n,2) \;\middle|\; \begin{array}{rlrl} \boldsymbol{u}_1 + \boldsymbol{u}_2 &= \boldsymbol{e}, & \mathbf{U}_{i1} + \mathbf{U}_{i2} &= \boldsymbol{u}_i\boldsymbol{e}^\top, \\ \mathrm{diag}(\mathbf{U}_{11}) &= \boldsymbol{u}_1, & \mathrm{diag}(\mathbf{U}_{22}) &= \boldsymbol{u}_2, \\ \langle \boldsymbol{u}_1, \boldsymbol{e}\rangle &\geq 1, & \langle \boldsymbol{u}_2, \boldsymbol{e}\rangle &\geq 1, \\ & & \mathbf{U}(n,2) &\geq \mathbf{0} \end{array}\right\},$$

where the inequalities $\langle \boldsymbol{u}_i, \boldsymbol{e}\rangle \geq 1$ are the only addition to $\mathcal{N}_1^*(\mathcal{U}_{n,2})$ over $\mathcal{N}_1^*(\mathcal{U}_{n,2}^{\mathbf{0}})$.

Theorem 5.11:
For all $n \in \mathbb{N}$, it holds that

$$\mathcal{N}_1^*(\mathcal{U}_{n,2}^{\mathbf{0},lex}) \cong \mathcal{N}_1^*(\mathcal{U}_{n-1,2}^{\mathbf{0}}),$$

where

$$\mathcal{U}_{n,2}^{\mathbf{0},lex} = \left\{\mathbf{U} \in \mathcal{U}_{n,2}^{\mathbf{0}} \;\middle|\; u_{11} = 1\right\}.$$

Proof. Let $\mathbf{U} \in \mathcal{N}_1^*(\mathcal{U}_{n,2}^{\mathbf{0},lex})$. Then $\mathbf{U} \in \mathcal{N}_1^*(\mathcal{U}_{n,2}^{\mathbf{0}})$ and by definition of MM, we have the additional constraints $(\boldsymbol{u}_1)_1 = 1$ and $\mathbf{U}_{i1}\boldsymbol{e}_1 = \boldsymbol{u}_i$ for $i = 1, 2$. In particular,

$$\mathbf{U} = \left(\begin{array}{c|c|c} 1 & \boldsymbol{u}_1^\top & \boldsymbol{u}_2\top \\ \hline \boldsymbol{u}_1 & \mathbf{U}_{11} & \mathbf{U}_{12} \\ \hline \boldsymbol{u}_2 & \mathbf{U}_{21} & \mathbf{U}_{22} \end{array}\right) = \left(\begin{array}{c|c|cc|cc} 1 & 1 & \boldsymbol{u}_1'^\top & 0 & \boldsymbol{u}_2'\top \\ \hline 1 & 1 & \boldsymbol{u}_1'^\top & 0 & \boldsymbol{u}_2'\top \\ \boldsymbol{u}_1' & \boldsymbol{u}_1' & \mathbf{U}_{11}' & \mathbf{0} & \mathbf{U}_{12}' \\ \hline 0 & 0 & \mathbf{0} & 0 & \mathbf{0} \\ \boldsymbol{u}_2' & \boldsymbol{u}_2' & \mathbf{U}_{21}' & \mathbf{0} & \mathbf{U}_{22}' \end{array}\right), \tag{5.10}$$

where the blocks of both matrices correspond to each other. Then we can reduce $\mathbf{U} \succeq \mathbf{0}$ to $\mathbf{U}' \succeq \mathbf{0}$, where

$$\mathbf{U}' = \begin{pmatrix} 1 & \boldsymbol{u}_1'^\top & \boldsymbol{u}_2'\top \\ \boldsymbol{u}_1' & \mathbf{U}_{11}' & \mathbf{U}_{12}' \\ \boldsymbol{u}_2' & \mathbf{U}_{21}' & \mathbf{U}_{22}' \end{pmatrix},$$

due to Corollary 3.3.

But then $\mathbf{U}' \in \mathcal{N}_1^*(\mathcal{U}_{n-1,2}^{\mathbf{0}})$, which can be verified by applying the equations from $\mathcal{N}_1^*(\mathcal{U}_{n,2}^{\mathbf{0}})$ on the blocks given by the right hand side of (5.10). □

While it is true that from a structural point of view, we could now use Theorem 5.8 on the reformulation given by Lemma 5.10 to end up with RPM_2^{m-1}, the reduction breaks down when we consider the objective function as well. When formulating (5.9) in the setting of $\mathcal{N}_1^*(\mathcal{U}_{n,2})$, we get the objective matrix

$$\left(\begin{array}{c|c|c} 0 & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{W} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{W} \end{array}\right) = \left(\begin{array}{c|cc|cc} 0 & 0 & \mathbf{0} & 0 & \mathbf{0} \\ \hline 0 & w_{11} & \boldsymbol{w}_1'^\top & 0 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{w}_1' & \mathbf{W}_{11}' & \mathbf{0} & \mathbf{0} \\ \hline 0 & 0 & \mathbf{0} & w_{11} & \boldsymbol{w}_1'^\top \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{w}_1'^\top & \mathbf{W}_{11}' \end{array}\right) \mapsto \mathbf{W}' = \begin{pmatrix} w_{11} & \boldsymbol{w}_1'^\top & \mathbf{0} \\ \boldsymbol{w}_1' & \mathbf{W}_{11}' & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{W}_{11}' \end{pmatrix},$$

for which we cannot find a new objective $\mathbf{W}''$ such that

$$\langle \mathbf{U}', \mathbf{W}' \rangle = \langle \varphi_2(\mathbf{U}'), \mathbf{W}'' \rangle.$$

This shows that applying MM to orbitopes does indeed break the symmetry and prevents a compact formulation similar to RPM_2^{m-1}, though at the cost of an SDP of twice the size. As a corollary, the same is true for the case when $L = 2^{[n]}$, with the minor addition of the inequality $\langle \boldsymbol{u}_2, \boldsymbol{e} \rangle \geq 1$, since $\langle \boldsymbol{u}_1, \boldsymbol{e} \rangle \geq 1$ is satisfied by assumption.

Corollary 5.12:
For all $n \in \mathbb{N}$, it holds that

$$\mathcal{N}_1^*(\mathcal{U}_{n,2}^{lex}) \cong \left\{ \mathbf{U}(n-1,2) \;\middle|\; \begin{array}{rlrl} \boldsymbol{u}_1 + \boldsymbol{u}_2 &= \boldsymbol{e}, & \mathbf{U}_{i1} + \mathbf{U}_{i2} &= \boldsymbol{u}_i \boldsymbol{e}^\top, \\ \mathrm{diag}(\mathbf{U}_{11}) &= \boldsymbol{u}_1, & \mathrm{diag}(\mathbf{U}_{22}) &= \boldsymbol{u}_2, \\ \langle \boldsymbol{u}_2, \boldsymbol{e} \rangle &\geq 1, & \mathbf{U}(n-1,2) &\succeq \mathbf{0} \end{array} \right\}.$$

Chapter 6

Projection Matrices

So far, we have seen that the symmetrical nature of assignment matrices $\mathcal{U}_{n,k}(L)$ has both led to meaningless solutions as well as more compact formulations of otherwise large problems. The reason for this is the 'global' representation of the partition:

Given the i-th row of an assignment matrix, we only know the label of the part that contains i, but not how this set actually looks like. Since we need the complete assignment matrix in order to reconstruct the part, in a sense, this information is locally unknown.

The idea of this chapter is to make this information locally available - if we assign an actual description of the part containing i to i instead of a label, symmetry will cease to be an issue, although at the price of a redundant description.

To this end, we will express the partitions as projection matrices in Section 6.1, which will have nice properties concerning the hierarchy constructed by MM in Section 6.2. In Section 6.3, we will apply this representation to both the graph-colouring and Euclidean k-clustering problems. As a main result of the thesis, we give a new formulation of the graph colouring number $\chi(G)$ and relate its relaxations to well-studied ones from literature.

6.1 Overview

Throughout this chapter, let $\mathbf{R} = (\boldsymbol{r}_1, \dots, \boldsymbol{r}_n) = \{r_{ij}\}_{i,j\in[n]} \in \mathbb{R}^{n\times n}$. We first recall some basic properties about projection matrices arising from orthogonal projections onto subspaces.

Definition 6.1:
Let $\mathbf{X} \in \mathbb{R}^{n\times k}$ be a matrix with full column rank to define its corresponding *projection matrix* as

$$\rho(\mathbf{X}) := \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top \in \mathbb{R}^{n\times n}.$$

The set of symmetric projection matrices of size n is defined as

$$\mathrm{SProM}^n := \left\{\mathbf{R} \in \mathbb{R}^{n\times n} \;\middle|\; \mathbf{R}^2 = \mathbf{R},\, \mathbf{R} = \mathbf{R}^\top\right\} \subseteq \mathcal{S}_+^n.$$

Note that by assumption, $\mathbf{X}^\top\mathbf{X}$ is invertible and $\rho(\mathbf{X}) \in \mathrm{SProM}^n$. We proceed by recalling some fundamental properties of projection matrices.

Lemma 6.2:
Let $\mathbf{R} \in \mathrm{SProM}^n$ *and* $\mathbf{X} \in \mathbb{R}^{n\times k}$ *with full column rank. Then the following holds:*

(i) Binary eigenvalues: $\mathrm{spct}(\mathbf{R}) \subseteq \{0, 1\}$,

(ii) Rank equation: $\mathrm{tr}(\mathbf{R}) = \mathrm{rank}(\mathbf{R})$,

(iii) Projection property: $\rho(\mathbf{X})\mathbf{X} = \mathbf{X}$, $\rho(\mathbf{X}) \cdot \mathbb{R}^n = \mathbf{X} \cdot \mathbb{R}^k$,

(iv) Isometry invariance: $\rho(\mathbf{X}) = \rho(\mathbf{XQ})$ *for all orthogonal* $\mathbf{Q} \in \mathbb{R}^{n\times n}$.

Proof. (i) follows from the Cayley-Hamilton theorem applied to the matrix polynomial $\mathbf{R}^2 = \mathbf{R}$. For (ii), note that $\mathrm{tr}(\mathbf{R})$ is equivalent to the sum of its eigenvalues, which count the rank due to (i). (iii) and (iv) follow directly from the definition. □

Remark 6.3:
The set of projection matrices can be bijectively mapped to the set of all linear subspaces of $\mathbb{R}^n$ by assigning a projection matrix to its image. In particular, the matrix $\rho(\mathbf{X})$ maps to the column space spanned by $\mathbf{X}$ as can be seen by Lemma 6.2.

More specifically, we can recover the *Grassmannian* $\mathrm{Gr}(k, \mathbb{R}^n)$ as a real variety by intersecting SProM^n with the affine hyperplane

$$\left\{\mathbf{R} \in \mathbb{R}^{n\times n} \;\middle|\; \mathrm{tr}(\mathbf{R}) = k\right\}$$

and evoking the rank equation in Lemma 6.2.

Example 6.4:
Recall Example 5.6, where we looked at the assignment matrix $\mathbf{U}$ and corresponding partition matrix $\mathbf{UU}^\top$ of the partition $\{\{1, 2\}, \{3\}\}$ given by

$$\mathbf{U} = \begin{pmatrix} 0 & 1 \\ 0 & 1 \\ 1 & 0 \end{pmatrix} \text{ and } \mathbf{UU}^\top = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Applying ρ to $\mathbf{U}$ leads us to the corresponding projection matrix

$$\rho(\mathbf{U}) = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Denoting the columns as $\rho(\mathbf{U}) = \left(\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_3\right)$, we observe for the following that

- $\mathbf{U}^\top\mathbf{U}$ is a diagonal matrix which contains the size of parts of the partition,
- $\rho(\mathbf{U})$ is a rescaled version of the partition matrix $\mathbf{U}\mathbf{U}^\top$,
- the columns $\{\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_3\}$ consist of all eigenvectors of $\rho(\mathbf{U})$ with repetition,
- $\{\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_3\} \subseteq \Delta^3$,
- $\|\boldsymbol{r}_i\|_0 \cdot \|\boldsymbol{r}_i\|_\infty = 1$ holds for each column $i \in [3]$,
- $\boldsymbol{r}_i = \boldsymbol{r}_j$ if and only if $(\boldsymbol{r}_i)_j = (\boldsymbol{r}_j)_i > 0$.

The preceding example motivates the following definition of a special class of doubly stochastic matrices.

Definition 6.5:
The set of *combinatorial projection matrices* $\mathrm{CProM}_k^n(L)$ is given as

$$\mathrm{CProM}_k^n(L) = \left\{ \mathbf{R} \in \mathbb{R}^{n\times n} \;\middle|\; \begin{array}{rll} \mathrm{col}(\mathbf{R}) & \subseteq \Delta_L^n & \\ r_{ij} \cdot (\boldsymbol{r}_i - \boldsymbol{r}_j) & = \mathbf{0} & \forall i,j \in [n] \\ \mathbf{R} & = \mathbf{R}^\top & \\ \mathrm{tr}(\mathbf{R}) & = k & \end{array} \right\},$$

where

$$\Delta_L^n := \{\boldsymbol{r} \in \Delta^n \mid \mathrm{supp}(\boldsymbol{r}) \in L\}\,.$$

The quadratic equations

$$r_{ij} \cdot (\boldsymbol{r}_i - \boldsymbol{r}_j) = \mathbf{0} \quad \forall i,j \in [n]$$

will be called *block-inducing* in the following.

Unsurprisingly, this set turns out to be exactly the projection matrices corresponding to assignment matrices, as shown by the following two lemmas.

Lemma 6.6:
Combinatorial projection matrices are projection matrices

$$\mathrm{CProM}_k^n(L) \subseteq \mathrm{SProM}^n$$

and have strictly positive diagonal $\mathrm{diag}(\mathbf{R}) > 0$. *In particular, each column* $\boldsymbol{r}_i$ *is uniquely determined by its support via*

$$r_{ji} = \delta_{j\in\mathrm{supp}(\boldsymbol{r}_i)} \frac{1}{\|\boldsymbol{r}_i\|_0}$$

and satisfies the equation

$$\|\boldsymbol{r}_i\|_0 \cdot \|\boldsymbol{r}_i\|_\infty = 1.$$

Proof. Let $\mathbf{R} \in \mathrm{CProM}_k^n(L)$ and choose any $i, j, l \in [n]$. By assumption, the block-inducing equations $r_{il} \cdot (\boldsymbol{r}_i - \boldsymbol{r}_l) = \mathbf{0}$ and $r_{ij} \cdot (\boldsymbol{r}_i - \boldsymbol{r}_j) = \mathbf{0}$ show the identity

$$r_{il} r_{jl} = r_{il} r_{ij} = r_{ij} r_{il} = r_{ij} r_{jl},$$

using the symmetry of $\mathbf{R}$. Then $\mathbf{R}^2 = \mathbf{R}$ follows from

$$(\mathbf{R}^2)_{ij} = \sum_{l \in [n]} r_{il} r_{jl} = \sum_{l \in [n]} r_{ij} r_{jl} = r_{ij} \Big(\sum_{l \in [n]} r_{jl} \Big) = r_{ij} = (\mathbf{R})_{ij} \quad \forall i, j \in [n].$$

Furthermore, the block-inducing equations $r_{ij} \cdot (\boldsymbol{r}_i - \boldsymbol{r}_j) = \mathbf{0}$ show $r_{ii}(r_{ij} - r_{ii}) = 0$ for all $j \in [n]$, which implies $r_{ij} \in \{0, r_{ii}\}$. Since $\boldsymbol{r}_i \in \Delta^n$, we necessarily have $r_{ii} \cdot \|\boldsymbol{r}_i\|_0 = 1$, which finishes the proof. □

Theorem 6.7:
The combinatorial projection matrices are precisely the projection matrices corresponding to assignment matrices:

$$\rho(\mathcal{U}_{n,k}(L)) = \mathrm{CProM}_k^n(L) \subseteq \mathrm{SProM}^n .$$

In particular, ρ is a bijection that shows

$$\mathcal{U}_{n,k}^{lex}(L) \leftrightarrow \mathrm{CProM}_k^n(L).$$

Proof. We first show $\rho(\mathcal{U}_{n,k}(L)) \subseteq \mathrm{CProM}_k^n(L)$, so let $\mathbf{U} = (\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k) \in \mathcal{U}_{n,k}(L)$. We already know $\rho(\mathbf{U}) \in \mathrm{SProM}^n$ and the corresponding properties from Lemma 6.2, so we only need to check the first two properties outlined in Definition 6.5.

For the first, assume $(\boldsymbol{u}_j)_i = 1$ to see

$$\rho(\mathbf{U})\boldsymbol{e}_i = \mathbf{U}(\mathbf{U}^\top \mathbf{U})^{-1} \mathbf{U}^\top \boldsymbol{e}_i = \mathbf{U}(\mathbf{U}^\top \mathbf{U})^{-1} \boldsymbol{e}_j = \mathbf{U} \frac{\boldsymbol{e}_j}{\langle \boldsymbol{u}_j, \boldsymbol{u}_j \rangle} = \frac{\boldsymbol{u}_j}{\langle \boldsymbol{e}, \boldsymbol{u}_j \rangle} \in \Delta_L^n,$$

where we used that $\boldsymbol{u}_j$ is binary in the second to last equation. For later, note the implication that

$$\{\mathrm{supp}(\rho(\mathbf{U})\boldsymbol{e}_i) \mid i \in [n]\} = \left\{ \mathrm{supp}(\mathbf{U}\boldsymbol{e}_j) \,\middle|\, j \in [k] \right\}. \tag{6.1}$$

For the second, assume $\boldsymbol{e}_i^\top \rho(\mathbf{U}) \boldsymbol{e}_j > 0$, since $\rho(\mathbf{U}) \geq 0$ and there is nothing to show otherwise. Assuming $(\boldsymbol{u}_{j'})_j = 1$ and $(\boldsymbol{u}_{i'})_i = 1$, we have

$$0 < \boldsymbol{e}_i^\top \rho(\mathbf{U}) \boldsymbol{e}_j = \boldsymbol{e}_i^\top \mathbf{U}(\mathbf{U}^\top \mathbf{U})^{-1} \mathbf{U}^\top \boldsymbol{e}_j = e_{i'}^\top (\mathbf{U}^\top \mathbf{U})^{-1} e_{j'} = \delta_{i',j'} \cdot (\mathbf{U}^\top \mathbf{U})^{-1}_{i',i'},$$

since $(\mathbf{U}^\top \mathbf{U})^{-1}$ is diagonal. It follows that $i' = j'$, which in turn means

$$\mathbf{U}^\top \boldsymbol{e}_i = \boldsymbol{e}_{i'} = \boldsymbol{e}_{j'} = \mathbf{U}^\top \boldsymbol{e}_j.$$

In particular,

$$\rho(\mathbf{U})(\boldsymbol{e}_i - \boldsymbol{e}_j) = \mathbf{U}(\mathbf{U}^\top\mathbf{U})^{-1}\left(\mathbf{U}^\top(\boldsymbol{e}_i - \boldsymbol{e}_j)\right) = \mathbf{0},$$

which we wanted to show.

The reverse inclusion $\rho(\mathcal{U}_{n,k}(L)) \supseteq \mathrm{CProM}_k^n(L)$ follows if we can show that ρ is a bijection from $\mathcal{U}_{n,k}^{lex}(L)$ to $\mathrm{CProM}_k^n(L)$, since $\rho(\mathcal{U}_{n,k}(L)) = \rho(\mathcal{U}_{n,k}^{lex}(L))$ by the isometry invariance of Lemma 6.2.

To see that ρ is injective, note that the invariant (6.1) immediately shows that whenever $\mathbf{U}, \mathbf{U}' \in \mathcal{U}_{n,k}^{lex}(L)$ are assignment matrices of distinct partitions $\mathcal{T} \neq \mathcal{T}'$, then also $\rho(\mathbf{U}) \neq \rho(\mathbf{U}')$.

To show surjectivity, we argue that for $\mathbf{R} \in \mathrm{CProM}_k^n(L)$, the map

$$\mathbf{R} \to \mathcal{T}(\mathbf{R}) =: \{\mathrm{supp}(\boldsymbol{r}_i) \mid i \in [n]\} \in \mathcal{P}_k^n(L)$$

is well-defined and injective, since then the corresponding assignment-matrices define a proper inverse $\rho^{-1}(\mathbf{R})$.

Since $\boldsymbol{r}_i \in \Delta_L^n$, we have $\mathcal{T}(\mathbf{R}) \subseteq L_*$. Disjointness of the sets in $\mathcal{T}(\mathbf{R})$ follows from the implications of the block-inducing equations, and coverage of $[n]$ follows due to $\mathbf{R}$ being doubly stochastic and so $\mathcal{T}(\mathbf{R}) \in \mathcal{P}_k^n(L)$ is well-defined.

Finally, the map $\mathbf{R} \to \mathcal{T}(\mathbf{R})$ is injective since $\mathbf{R}$ can be uniquely reconstructed from $\mathcal{T}(\mathbf{R})$ by starting with $i \in \mathrm{supp}(\boldsymbol{r}_i)$ and applying Lemma 6.6. □

6.2 Convexification

As in the case of partition matrices, we are interested in the complexity of optimizing a linear function over $\mathrm{CProM}_k^n(L)$. Unfortunately, Section 6.3 shows examples where this is NP-hard, so we can not expect a algorithmically exploitable description. For this reason, we again revert to studying relaxations of the convex hull.

The easiest convex relaxation for $\mathrm{CProM}_k^n(L)$ is given by replacing the non-linear constraints

$$r_{ij} \cdot (\boldsymbol{r}_i - \boldsymbol{r}_j) = \mathbf{0} \quad \forall i, j \in [n]$$

with a psd. constraint $\mathbf{R} \succeq \mathbf{0}$ emerging from the fact that since $\mathbf{R}$ is a projection matrix, we have $\mathbf{R}^2 = \mathbf{R}$. We thus get the set

$$\mathrm{RProM}_k^n(L)^0 = \left\{\mathbf{R} \in \mathbb{R}_+^{n\times n} \;\middle|\; \mathrm{tr}(\mathbf{R}) = k,\ \mathbf{R}\boldsymbol{e} = \boldsymbol{e},\ \langle \mathbf{R}, \Omega_L\rangle = 0,\ \mathbf{R} \succeq \mathbf{0}\right\}.$$

Just like in the preceding chapters, the idea now is to take the variety $\mathrm{CProM}_k^n(L)$ and apply MM to it. For this reason, we will need to investigate the monomial structure of the underlying ideal of $\mathrm{CProM}_k^n(L)$, which we will call $\mathcal{I}_\rho$ and is defined as

$$\mathcal{I}_\rho = \left\langle \left\{ r_{ij} - r_{ji},\ \mathrm{tr}(\mathbf{R}) - k,\ \langle \boldsymbol{r}_i, \boldsymbol{e}\rangle - 1,\ r_{ij}(r_{il} - r_{jl})\ \forall i, j, l \in [n] \right\} \right\rangle \subseteq \mathbb{R}[\mathbf{R}].$$

Unfortunately, the ideal $\mathcal{I}_\rho$ is an example where stating a reduced Gröbner basis and a corresponding basis for $\mathbb{R}[\mathbf{R}]/\mathcal{I}_\rho$ explicitly becomes difficult and would distract from the underlying problem structure. For this reason, we will only state a generating system instead of a basis.

Lemma 6.8:
A monomial generating system of $\mathbb{R}_t[\mathbf{R}]/\mathcal{I}_\rho$ can be indexed by the set

$$\mathrm{ProMo}_{n,t} = \left\{ (G,\psi) \in \mathcal{G}_n \times \mathbb{N}^{\mathrm{CC}(G)} \;\middle|\; \frac{1}{2}|E(G)| + \sum_{T\in\mathrm{CC}(G)} \psi(T) = t \right\}.$$

Proof. Let $\mathbf{R}^{\mathbf{\Psi}} \in \mathbb{R}_t[\mathbf{R}]$, so in particular $\mathbf{\Psi} \in \mathbb{N}_t^{n\times n}$. We will show that each equivalence class $[\mathbf{R}^{\mathbf{\Psi}}] \in \mathbb{R}_t[\mathbf{R}]/\mathcal{I}_\rho$ can be identified with a pair (G,ψ) as described in the set above. To simplify notation, we will note that the congruence $\mathbf{R}^{\mathbf{\Psi}} \equiv \mathbf{R}^{\mathbf{\Psi}'} \mod \mathcal{I}_\rho$ can equivalently be expressed as a congruence on the exponent matrices $\mathbf{\Psi} \cong \mathbf{\Psi}'$, whose integer entries describe the powers of the corresponding variables.

First, since the columns of $\mathbf{R}$ are normalized, we have the inclusion

$$\mathbb{R}[\mathbf{R}]/\mathcal{I}_\rho \subseteq \prod_{i\in[n]} \mathbb{R}[\boldsymbol{r}_i]/\mathcal{I}(\Delta^n)$$

and we can use homogenization (2.5) on any column $\boldsymbol{r}_i$ to see that $\mathbf{R}^{\mathbf{\Psi}}$ can be written as a sum of several $\mathbf{R}^{\mathbf{\Psi}'}$, where $\mathbf{\Psi}' \in \mathbb{N}^{n\times n}$ and $\langle \mathbf{\Psi}', \mathbf{J}_n\rangle = t$.

Second, we can find $\mathbf{\Psi}' \cong \mathbf{\Psi}''$ where $\mathbf{\Psi}''$ is upper triangular, since $r_{ij} = r_{ji}$, and so everything can be expressed in terms of r_{ij} with $i \le j$.

Third, we can use the identities $r_{ij}(r_{il} - r_{jl}) = 0$ with $l = j$ to see that

$$r_{jj} \cdot r_{ij} = r_{ij}^2 = r_{ii} \cdot r_{ij}, \tag{6.2}$$

which shows that we can decrease the off-diagonal entries of $\mathbf{\Psi}$ exceeding 1 to increase the main-diagonal of the corresponding column or row. We thus can find upper-triangular $\mathbf{\Psi}''' \cong \mathbf{\Psi}''$ such that

$$\mathbf{\Psi}''' \in \{0,1\}^{n\times n} + \mathrm{Diag}(\mathbb{N}^n), \quad \langle \mathbf{\Psi}'', \mathbf{J}_n\rangle = t,$$

e.g. all off-diagonal exponents are binary.

In this form, the off-diagonal entries of $\mathbf{\Psi}'''$ define the adjacency matrix of a directed graph, which we can identify with its underlying undirected graph G. We define a map $\psi \in \mathbb{N}^{\mathrm{CC}(G)}$ by setting

$$\psi(T) = \sum_{i\in T} \mathbf{\Psi}_{ii} \quad \forall T \in \mathrm{CC}(G).$$

Then (6.2) shows that the construction of ψ is independent on how we choose to construct $\mathbf{\Psi}'''$ from $\mathbf{\Psi}''$, since we can decrease any $\mathbf{\Psi}'''_{ii}$ by an integer to increase $\mathbf{\Psi}'''_{jj}$ by the

same amount whenever the edge (i, j) is contained in G without leaving its equivalence class. In particular, this extends to any connecting path between i and j, such that the sum of the diagonal entries over a connected component are invariant. We can thus identify Ψ' with the couple (G, ψ) as claimed, allowing us to write any $\mathbf{R}^{\Psi}$ as a linear combination of such monomials. □

One can verify that $\mathrm{ProMo}_{n,1} \cong \left\{(i,j) \in [n]^2 \;\middle|\; i \leq j\right\}$, as graphs containing a single edge (i, j) can be identified with that edge, and the connected components of edgeless graphs can be identified with their vertices i to yield (i, i). We thus recover the monomials $\left\{r_{ij} \;\middle|\; i \leq j\right\}$ we started with.

Remark 6.9:
The monomial generating system indexed by $\mathrm{ProMo}_{n,t}$ in Lemma 6.8 is not isomorphic to a basis for $\mathbb{R}_t[\mathbf{R}]/\mathcal{I}_\rho$, since further reductions via the unused equation $\mathrm{tr}(\mathbf{R}) - k$ are possible. We neglect any such simplifications for the sake of clarity.

While considering $\mathbb{R}_t[\mathbf{R}]$ with symmetry equations would lead to $z_{\binom{n}{2}}(t)$ monomials, the reduced system indexed by $\mathrm{ProMo}_{n,t}$ grows much slower, as can be seen in Table 6.1.

	1	2	3	4	5
5	15	100	390	1000	1863
6	21	201	1156	4451	12230
7	28	364	2905	15876	62972

	1	2	3	4	5
5	16	136	816	3876	15504
6	22	253	2024	12650	65780
7	29	435	4495	35960	237336

Table 6.1: Left: $|\mathrm{ProMo}_{n,t}|$ indexed by pairs (n, t). Right: Values $\dim(\mathbb{R}_t[\mathbf{R}])$ indexed by pairs (n, t).

Table 6.1 shows that the number of monomials for approximating $\mathrm{CProM}_k^n(L)$ grows much too fast. Unfortunately, even the very first stage of MM needs second order moments, which still grow too fast even for small inputs. To remedy this, we can relax the first stage of MM by discarding some of the off-diagonal entries. One way to do this is by starting with all monomials in $\mathbb{R}[\mathbf{R}]$, discarding off-diagonal entries to end up with a block-diagonal structure and then reducing the remaining matrices by using a smaller monomial base suited for $\mathbb{R}[\mathbf{R}]/\mathcal{I}_\rho$.

The main matrix in $\mathcal{N}_1^*(\mathrm{CProM}_k^n(L))$ is given by

$$\mathbf{M}_1(\mathbf{R}) = \begin{pmatrix} 1 & \boldsymbol{r}_1^\top & \dots & \boldsymbol{r}_n^\top \\ \boldsymbol{r}_1 & \mathbf{R}_{11} & \dots & \mathbf{R}_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{r}_n & \mathbf{R}_{n1} & \dots & \mathbf{R}_{nn} \end{pmatrix} \succeq \mathbf{0}.$$

The easiest way to relax this condition is to discard all matrices $\mathbf{R}_{ij}$ with $i \neq j$, which relaxes $\mathbf{M}_1(\mathbf{R}) \succeq \mathbf{0}$ to the system

$$\mathbf{M}_1(\mathbf{r}_i) = \begin{pmatrix} 1 & \boldsymbol{r}_i^\top \\ \boldsymbol{r}_i & \mathbf{R}_{ii} \end{pmatrix} \succeq \mathbf{0} \quad \forall i \in [n].$$

Invoking the results of Lemma 6.8, we see that we can discard the first row and column from this condition. Applying the remaining constraints for each $\mathbf{R}_{ii}$, we end up with feasible sets

$$R_i = \left\{ \mathbf{R}_{ii} \in \mathbb{R}_+^{n \times n} \;\middle|\; \langle \mathbf{R}_{ii}, \mathbf{J}_n \rangle = 1,\ \mathbf{R}_{ii}\boldsymbol{e}_i = \operatorname{diag}(\mathbf{R}_{ii}),\ \mathbf{R}_{ii} \succeq \mathbf{0} \right\},$$

where the condition $\mathbf{R}_{ii}\boldsymbol{e}_i = \operatorname{diag}(\mathbf{R}_{ii})$ breaks the symmetry between different $i \in [n]$. By invoking the block inducing equations, one can show that additionally, the matrices $\mathbf{R}_{ii}$ are linked through the constraints

$$\sum_{i \in [n]} \mathbf{R}_{ii} = \mathbf{R}$$

and

$$(\mathbf{R}_{ii})_{jl} = (\mathbf{R}_{jj})_{il} = (\mathbf{R}_{ll})_{ij},$$

which implies that we can consider each matrix $\mathbf{R}_{ii}$ as a slice of a symmetrical third-order tensor, as shown in Figure 6.1.

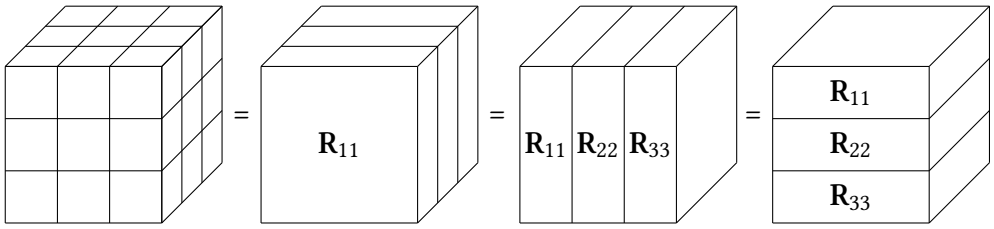

Figure 6.1: Different ways to align the variables of the various $\mathbf{R}_{ii}$ into the same symmetrical third order tensor, each cube representing one variable.

Denoting this structure with the set

$$R = \left\{ \mathbf{R} \in \mathbb{R}^{n \times n} \;\middle|\; \mathbf{R} = \sum_{i \in [n]} \mathbf{R}_{ii},\ (\mathbf{R}_{ii})_{jl} = (\mathbf{R}_{jj})_{il},\ \mathbf{R}_{ii} \in R_i\ \forall i, j, l \in [n] \right\},$$

a relaxation of the first stage of MM is given by

$$\operatorname{RProM}_k^n(L) = \{ \mathbf{R} \in R \mid \operatorname{tr}(\mathbf{R}) = k,\ \mathbf{R}\boldsymbol{e} = \boldsymbol{e},\ \langle \mathbf{R}, \Omega_L \rangle = 0 \},$$

where $\operatorname{RProM}_k^n(L)^0$ relaxes this set by using $R \subseteq \mathbb{R}_+^{n \times n} \cap \mathcal{S}_+^n$.

Remark 6.10:
The approach of constructing a symmetrical third-order tensor to relax a moment matrix of order 1 was already proposed in [GL08] to relax a combinatorial moment matrix of order 1 for the stable set problem. In particular, each entry of our tensor can be identified with a set $\{i, j, l\}$ via its indices, which can in turn be understood as a rescaling of combinatorial moments up to the third order.

While this process can be generalized to turn higher order combinatorial moment matrices of rank 1 into symmetrical tensors of appropriate order, it does not work in our setting. The main reason is the assumption $\operatorname{rank}(\mathbf{R}) = k > 1$, which destroys the symmetry; in general, the equation $(\mathbf{R}_{ij})_{lm} = (\mathbf{R}_{il})_{jm}$ will not hold, so we can not derive unique entries for the entries associated with $\{i, j, l, m\}$.

6.3 Applications

6.3.1 Graph Colouring

We will shortly recall the basics about graph colouring before we show how to use projection matrices for a new formulation of this problem. For an extensive survey on convex relaxations for graph colouring, consider [GL08].

Stable Sets

Definition 6.11:
A *stable set* of a graph $G \in \mathcal{G}_n$ is a subset $S \subseteq [n]$ such that the subgraph induced by S does not contain any edges. The set of stable sets in G is denoted by $S_G \subseteq 2^{[n]}$. The *stable set number* $\alpha(G)$ is the biggest size $|S|$ of any stable set in G and defines a function $\alpha : \mathcal{G}_n \to \mathbb{N}$. The *clique number* $\omega(G)$ of G is the biggest size $|S|$ of a stable set in the complement graph $\overline{G}$, so $\omega(G) := \alpha(\overline{G})$.

Formally, we can express $\alpha(G)$ as the solution to an integer problem by working with characteristic vectors $\boldsymbol{x} \in \{0, 1\}^n$ via

$$\begin{aligned}\alpha(G) &= \max\left\{\langle \boldsymbol{x}, \boldsymbol{e}\rangle \;\middle|\; x_i x_j = 0 \quad \forall (i,j) \in E,\ \boldsymbol{x} \in \{0,1\}^n\right\} \\ &= \max\left\{\langle \boldsymbol{x}, \boldsymbol{e}\rangle \;\middle|\; \boldsymbol{x}^\top \mathbf{A}_G \boldsymbol{x} = 0,\ \boldsymbol{x} \in \{0,1\}^n\right\},\end{aligned}$$

where $\mathbf{A}_G$ denotes the adjacency matrix of G. For later, we will note the following crucial property of S_G.

Lemma 6.12:
The stable sets S_G form an independence system.

Proof. Follows directly by checking the properties in the definition (2.2). □

Graph Colourings

Definition 6.13:
A *k-colouring* of a graph $G = ([n], E)$ is a k-partition $\mathcal{T} = (T_1, \dots, T_k)$ of the node set $[n]$, such that each T_i is a stable set in G. The *chromatic number* $\chi(G)$ is the smallest value k such that G has a k-colouring.

As shown back in Example 4.7, computing $\chi(G)$ is a minimum cover problem, but it can not be modelled with assignment matrices alone. While there is a model involving assignment matrices, we will consider another standard formulation in terms of extended partition matrices instead.

$$\begin{aligned}\chi(G) &= \min\left\{k \in \mathbb{N} \;\middle|\; \mathbf{X} \in \mathrm{PM}_k^n(S_G)\right\} \\ &= \min\left\{k \in \mathbb{R} \;\middle|\; \begin{pmatrix} k & \boldsymbol{e}^\top \\ \boldsymbol{e} & \mathbf{X} \end{pmatrix} \succeq \mathbf{0},\ \langle \mathbf{X}, \mathbf{A}_G\rangle = 0,\ \mathrm{diag}(\mathbf{X}) = \boldsymbol{e},\ \mathbf{X} \in \{0,1\}^{n\times n}\right\}\end{aligned} \tag{6.3}$$

Unfortunately, we have the following hardness result.

Theorem 6.14 ([Sch03, BS94])**:**
Computing $\alpha(G)$ or $\chi(G)$ is NP-hard. Furthermore, it is NP-hard to approximate $\chi(G)$ within $n^{1/14-\epsilon}$ for any $\varepsilon > 0$.

In spite of being NP-hard to compute in general, $\chi(G)$ behaves much nicer restricted to special classes of graphs with regards to the following construction.

Definition 6.15 (Lovász theta number [Lov79])**:**
The *Lovász theta number* $\vartheta(G)$ is the optimal value of the following primal/dual SDP problems

$$\max\left\{\langle \mathbf{X}, \mathbf{J}_n\rangle \mid \langle \mathbf{X}, \mathbf{I}_n\rangle = 1,\ \langle \mathbf{X}, \mathbf{A}_G\rangle = 0,\ \mathbf{X} \succeq \mathbf{0}\right\}, \tag{ϑ-P}$$

$$\min\left\{k \;\middle|\; \mathrm{diag}(\mathbf{Y}) = \boldsymbol{e},\ \langle \mathbf{Y}, \mathbf{A}_{\overline{G}}\rangle = 0,\ \begin{pmatrix} k & \boldsymbol{e}^\top \\ \boldsymbol{e} & \mathbf{Y} \end{pmatrix} \succeq \mathbf{0}\right\}. \tag{ϑ-D}$$

Theorem 6.16 (Lovász sandwich Theorem [Lov87])**:**
For all $G \in \mathcal{G}_n$, we have

$$\omega(G) \leq \vartheta(\overline{G}) \leq \chi(G).$$

This result shines in the context of *perfect graphs*, where $\omega(H) = \chi(H)$ for all induced subgraphs $H \subseteq G$, and $\chi(G)$ can be computed in polynomial time. For more on the theory of perfect graphs, consider the recent survey [Tro15].

In spite of the hardness results in Theorem 6.14, a lot of work has been done to improve $\vartheta(G)$ to get better bounds for imperfect graphs. We exemplarily show the following variants taken from [DR07] and [GL08].

Nonnegativity

Adding $\mathbf{X} \geq \mathbf{0}$ to (ϑ-P) and $\mathbf{Y} \geq \mathbf{0}$ to (ϑ-D) results in the Schrijver number $\vartheta^-(G)$ [Sch79] and the Szegedy number $\vartheta^+(G)$ [Sze94] respectively.

Applying MM

Following Section 5.3, we can interpret $\mathrm{PM}_k^n(S_G)$ in (6.3) as variety in terms of $\mathcal{U}_{n,k}(S_G)$, and MM can be applied as shown before. Denoting the corresponding optimal value of the t-th stage as $\psi^{(t)}(G)$, we get a converging hierarchy of lower bounds for $\chi(G)$.

Note that we actually have $\psi^{(1)}(G) = \vartheta(\overline{G})$, since the feasible set of formulation (ϑ-D) is precisely the relaxation $\mathrm{RPM}_k^n(S_{\overline{G}})$ in this case. In particular, the Lovász theta number arises from the underlying MM hierarchy, and so MM can be seen as a natural generalization.

Summarizing the results so far, [GL08] shows

$$\psi^{(1)}(G) = \vartheta(\overline{G}) \leq \vartheta^+(\overline{G}) \leq \psi^{(2)}(G) \leq \chi(G).$$

There are several other relaxations for $\chi(G)$ available in [GL08], where among others the computationally tractable $\psi(G)$ is introduced.

The projection ϑ-number

We now propose a new formulation in terms of projection matrices and relate it to $\vartheta(G)$.

Theorem 6.17:
We have the following characterization of the chromatic number:

$$\begin{aligned}\chi(G) &= \min\left\{k \;\middle|\; \mathrm{CProM}_k^n(S_G) \neq \emptyset\right\} \\ &= \min\left\{\mathrm{tr}(\mathbf{R}) \;\middle|\; \begin{aligned} \mathrm{col}(\mathbf{R}) &\subseteq \Delta^n \\ r_{ij} &= 0 \qquad \forall (i,j) \in E(G) \\ r_{ij} \cdot (\boldsymbol{r}_i - \boldsymbol{r}_j) &= \mathbf{0} \qquad \forall i,j \in [n] \\ \mathbf{R} &= \mathbf{R}^\top \end{aligned} \right\}.\end{aligned}$$

Proof. The first equation follows by definition, so we will show the second. Let $\chi'(G)$ denote the optimal value of the second optimization problem and let $\mathcal{T}$ be a minimal colouring with $\chi(G)$ colours. Then the projection matrix $\mathbf{R}(\mathcal{T})$ corresponding to $\mathcal{T}$ is feasible for the second optimization problem, since $\mathrm{supp}(\mathbf{R}(\mathcal{T})\boldsymbol{e}_j) \in S_G$ and $\mathrm{diag}(\mathbf{R}(\mathcal{T})) > \mathbf{0}$ imply $\boldsymbol{e}_i^\top \mathbf{R}(\mathcal{T})\boldsymbol{e}_j = 0$ whenever $(i,j) \in E(G)$, and so $\chi'(G) \leq \chi(G)$.

To see the other inequality, first note that $\chi'(G) \geq 0$ is integral, since whenever $\mathbf{R}$ is feasible, $\mathbf{R} \in \mathrm{SProM}^n$ as shown in the proof of Lemma 6.6, and so the eigenvalues of

$\mathbf{R}$ are binary. It suffices to show that $\mathbf{R} \in \mathrm{CProM}^n_{\mathrm{tr}(\mathbf{R})}(S_G)$ then, and the only thing that is left to prove is $\mathrm{supp}(\boldsymbol{r}_i) \in S_G$.

Suppose that $\mathrm{supp}(\boldsymbol{r}_i) \notin S_G$, so there is $(j,l) \in E(G)$ such that $(\boldsymbol{r}_i)_j, (\boldsymbol{r}_i)_l > 0$. It follows from $r_{ij} \cdot (\boldsymbol{r}_i - \boldsymbol{r}_j) = \mathbf{0}$ that $(\boldsymbol{r}_j)_l = (\boldsymbol{r}_i)_l > 0$, which contradicts $(\boldsymbol{r}_j)_l = 0$. □

The importance of the preceding theorem lies in the fact that we made the constraint $\mathrm{supp}(\boldsymbol{r}_i) \in S_G$ tractable, so that we can use the hierarchy given by MM to approximate $\chi(G)$. Following Section 6.2, we can immediately give the 0-th stage of our relaxations.

Definition 6.18:
The projection ϑ-number is given as

$$\begin{aligned}\hat{\vartheta}(G) &:= \min\left\{k \in \mathbb{R}^n_+ \;\middle|\; \mathrm{RProM}^n_k(S_G)^0 \neq \emptyset\right\} \\ &= \min\left\{\mathrm{tr}(\mathbf{R}) \;\middle|\; \mathbf{R} \geq \mathbf{0},\, \mathbf{R}\boldsymbol{e} = \boldsymbol{e},\, \langle \mathbf{R}, \mathbf{A}_{\overline{G}}\rangle = 0,\, \mathbf{R} \geq \mathbf{0}\right\}. \end{aligned} \tag{6.4}$$

We should note that by non-negativity of $\mathbf{R}$, we have

$$\langle \mathbf{R}, \mathbf{A}_G\rangle = 0 \quad \Leftrightarrow \quad r_{ij} = 0 \quad \forall (i,j) \in E(G),$$

and we will prefer this notation. For the following, it will be important to explicitly write down the Szegedy number mentioned before, which is given as

$$\vartheta^+(G) := \min\left\{x_0 \;\middle|\; \begin{pmatrix} x_0 & \boldsymbol{e}^\top \\ \boldsymbol{e} & \mathbf{X}\end{pmatrix} \geq \mathbf{0},\, \mathrm{diag}(\mathbf{X}) = \boldsymbol{e},\, \langle \mathbf{X}, \mathbf{A}_{\overline{G}}\rangle = 0,\, \mathbf{X} \geq \mathbf{0}\right\}. \tag{6.5}$$

Since both bounds have similar computational complexity, it makes sense to compare their quality. Unfortunately, it turns out that the classical approach is always at least as good as using projection matrices, as shown in the following theorem.

Theorem 6.19:
Let $G \in \mathcal{G}_n$, then $\vartheta^+(G) \geq \hat{\vartheta}(G)$.

Proof. Let $x_0 = \vartheta^+(G)$ and $\mathbf{X}$ be an optimal solution to (6.5). Since $\mathbf{X}$ is symmetrical and non-negative, it follows from [BPS66] that there exists a diagonal matrix $\mathbf{D} = \mathrm{Diag}(\boldsymbol{d})$ with $\boldsymbol{d} > \mathbf{0}$ such that

$$\begin{pmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{D}\end{pmatrix}\begin{pmatrix} \vartheta^+(G) & \boldsymbol{e}^\top \\ \boldsymbol{e} & \mathbf{X}\end{pmatrix}\begin{pmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{D}\end{pmatrix} = \begin{pmatrix} \vartheta^+(G) & \boldsymbol{d}^\top \\ \boldsymbol{d} & \mathbf{DXD}\end{pmatrix} \geq \mathbf{0},$$

where $\mathbf{R} := \mathbf{DXD}$ is doubly stochastic, so $\mathbf{R}\boldsymbol{e} = \boldsymbol{e}$. In particular, $\mathbf{R}$ is feasible for (6.4), and we have $\mathrm{tr}(\mathbf{R}) = \|\boldsymbol{d}\|_2^2$. Using positive semidefiniteness, for any $\mu \in \mathbb{R}$, we have

$$0 \leq \begin{pmatrix} -1 \\ \mu\boldsymbol{d}\end{pmatrix}^\top \begin{pmatrix} \vartheta^+(G) & \boldsymbol{d}^\top \\ \boldsymbol{d} & \mathbf{R}\end{pmatrix}\begin{pmatrix} -1 \\ \mu\boldsymbol{d}\end{pmatrix} = \vartheta^+(G) - 2\mu\|\boldsymbol{d}\|_2^2 + \mu^2(\boldsymbol{d}^\top\mathbf{R}\boldsymbol{d})$$

which becomes

$$\begin{aligned}\vartheta^+(G) &\geq 2\mu\|\boldsymbol{d}\|_2^2 - \mu^2(\boldsymbol{d}^\top \mathbf{R}\boldsymbol{d}) = \|\boldsymbol{d}\|_2^2\left(2\mu - \mu^2(\tfrac{\boldsymbol{d}^\top}{\|\boldsymbol{d}\|_2}\mathbf{R}\tfrac{\boldsymbol{d}}{\|\boldsymbol{d}\|_2})\right)\\ &\geq \|\boldsymbol{d}\|_2^2\left(2\mu - \mu^2\right) = \operatorname{tr}(\mathbf{R})\left(2\mu - \mu^2\right).\end{aligned}$$

Since $\max\left\{2\mu - \mu^2 \;\middle|\; \mu \in \mathbb{R}\right\} = 1$, it follows that $\mathbf{R}$ is a feasible solution to (6.4) that satisfies $\vartheta^+(G) \geq \operatorname{tr}(\mathbf{R})$ and the theorem follows. □

A crucial question in understanding the projection-model is the magnitude of the gap $\vartheta^+(G) - \hat{\vartheta}(G)$. It turns out that we can state an explicit asymptotic lowerbound.

Theorem 6.20:
The worst case gap $\vartheta^+(G) - \hat{\vartheta}(G)$ has asymptotic behaviour

$$\max\left\{\vartheta^+(G) - \hat{\vartheta}(G) \;\middle|\; G \in \mathcal{G}_n\right\} = \Omega(n).$$

In particular,

$$\limsup_{n\to\infty} \max\left\{\frac{\vartheta^+(G) - \hat{\vartheta}(G)}{n} \;\middle|\; G \in \mathcal{G}_n\right\} \geq \left(\frac{3}{\sqrt{2}} - 2\right) \approx 0,1213.$$

Proof. We will explicitly construct a graph family for which the bound on the gap holds true asymptotically. To this end, consider for any two integers $n_1, n_2 \in \mathbb{N}$, the graph

$$G(n_1, n_2) := K_{n_1} \cup K_{n_2},$$

which we define as the union of two complete graphs with n_1 and n_2 nodes respectively. In particular, the corresponding adjacency matrix is given as

$$\mathbf{A}_{G(n_1,n_2)} = \begin{pmatrix} \mathbf{J}_{n_1} - \mathbf{I}_{n_1} & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_{n_2} - \mathbf{I}_{n_2} \end{pmatrix},$$

and since these graphs are perfect, we explicitly have

$$\chi(G(n_1, n_2)) = \vartheta^+(\overline{G(n_1, n_2)}) = \omega(G(n_1, n_2)) = \max(n_1, n_2).$$

For $\hat{\vartheta}(\overline{G(n_1, n_2)})$, we will construct the optimal solution $\mathbf{R}$ in closed form. Since this graph has multiple symmetries, we can assume $\mathbf{R}$ to be symmetry invariant and parametrize the feasible set with only three parameters α, β, γ by setting

$$\mathbf{R} = \begin{pmatrix} \alpha\mathbf{I}_{n_1} & \beta\mathbf{J}_{n_1,n_2} \\ \beta\mathbf{J}_{n_2,n_1} & \gamma\mathbf{I}_{n_2} \end{pmatrix}.$$

Now that the constraint $\langle \mathbf{R}, \mathbf{A}_{G(n_1,n_2)} \rangle = 0$ is satisfied, the remaining affine constraints turn into

$$\begin{aligned} \mathbf{R} \succeq \mathbf{0} \quad &\Leftrightarrow \quad \alpha, \beta, \gamma \geq 0, \\ \mathbf{R}\boldsymbol{e} = \boldsymbol{e} \quad &\Leftrightarrow \quad \alpha + n_2\beta = 1, \quad \gamma + n_1\beta = 1. \end{aligned}$$

Assuming $\alpha, \gamma \neq 0$ to make the condition $\mathbf{R} \succeq \mathbf{0}$ non-trivial, we can use the Schur complement Lemma 2.5 to rewrite

$$\begin{aligned} \mathbf{R} \succeq \mathbf{0} \quad &\Leftrightarrow \quad \gamma \mathbf{I}_{n_2} - \frac{\beta^2}{\alpha} \mathbf{J}_{n_2,n_1} \cdot \mathbf{J}_{n_1,n_2} \succeq \mathbf{0} \quad \Leftrightarrow \quad \gamma \mathbf{I}_{n_2} - \frac{\beta^2}{\alpha} n_1 \mathbf{J}_{n_2} \succeq \mathbf{0} \\ &\Leftrightarrow \quad \gamma \geq \frac{\beta^2}{\alpha} n_1 n_2, \end{aligned}$$

where we used the fact that $\mathbf{J}_n$ only has one non-zero eigenvalue given by n. Lastly, we can explicitly express the objective function as

$$\operatorname{tr}(\mathbf{R}) = \alpha n_1 + \gamma n_2 = n_1 + n_2 - 2n_1 n_2 \beta,$$

using the affine constraints. Ignoring the constants, the resulting problem of computing the number $\hat{\vartheta}(\overline{G(n_1, n_2)})$ is equivalent to

$$\max \left\{ \beta \;\middle|\; \alpha + n_2\beta = 1, \quad \gamma + n_1\beta = 1, \quad \alpha\gamma \geq \beta^2 n_1 n_2, \quad \alpha, \beta, \gamma \geq 0 \right\}.$$

Using the equations, one can show that the unique solution is

$$\begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} = \frac{1}{n_1 + n_2} \begin{pmatrix} n_1 \\ 1 \\ n_2 \end{pmatrix}$$

and

$$\hat{\vartheta}(\overline{G(n_1, n_2)}) = \frac{n_1^2 + n_2^2}{n_1 + n_2}.$$

In particular, we now have the gap

$$\Delta(n_1, n_2) := \vartheta^+(\overline{G(n_1, n_2)}) - \hat{\vartheta}(\overline{G(n_1, n_2)}) = \max(n_1, n_2) - \frac{n_1^2 + n_2^2}{n_1 + n_2}. \tag{6.6}$$

W.l.o.g., let $n_1 = m$, $n_2 = \mu m \in \mathbb{N}$ for some $\mu \in [0, 1]$. Then (6.6) reads

$$\Delta(m, \mu m) = m - \frac{(1 + \mu^2)m^2}{(1 + \mu)m} = \mu \left(\frac{1 - \mu}{1 + \mu} \right) m.$$

Finally, optimizing the choice $\mu \in [0, 1]$ yields the biggest theoretical gap for $\mu = \sqrt{2} - 1$ and

$$\max_{\mu \in [0,1]} \Delta(m, \mu m) = (3 - 2\sqrt{2})m.$$

For growing m, we can approximate this gap arbitrarily well by choosing $n_1 = m$ and $n_2 = \lceil(\sqrt{2}-1)m\rceil$ to get a graph of size $\lceil\sqrt{2}m\rceil$ with asymptotic relative gap

$$\limsup_{m\to\infty} \frac{\Delta(m, \lceil(\sqrt{2}-1)m\rceil)}{\lceil\sqrt{2}m\rceil} = \frac{3-2\sqrt{2}}{\sqrt{2}} \approx 0,1213.$$

□

As a side effect of the preceding theorem, we get the following corollary, which might explain the discrepancy by relating $\hat{\vartheta}(G)$ to Theorem 6.16.

Corollary 6.21:
The inequation $\omega(G) \leq \hat{\vartheta}(\overline{G})$ does not hold in general. In particular, $\hat{\vartheta}(G)$ is not necessarily exact for perfect graphs G.

Even though the preceding results show severe disadvantages over the classical formulation, we can recover some useful properties for the class of vertex-transitive graphs. We first cite the property in question.

Theorem 6.22 ([Sze94])**:**
For all $G \in \mathcal{G}_n$, the inequality

$$\vartheta^+(G) \cdot \vartheta^-(\overline{G}) \geq n$$

holds, with equality if G is vertex-transitive.

Surprisingly, we can use $\hat{\vartheta}(G)$ to sharpen this inequality.

Lemma 6.23:
For all $G \in \mathcal{G}_n$, the inequality $\hat{\vartheta}(G) \cdot \vartheta^-(\overline{G}) \geq n$ holds.

Proof. Let $\mathbf{R}$ be an optimal solution to (6.4). Recalling the definition

$$\vartheta^-(\overline{G}) = \max\left\{\langle \mathbf{X}, \mathbf{J}\rangle \;\middle|\; \operatorname{tr}(\mathbf{X}) = 1,\ \langle \mathbf{X}, \mathbf{A}_{\overline{G}}\rangle = 0,\ \mathbf{X} \succeq \mathbf{0},\ \mathbf{X} \geq \mathbf{0}\right\}$$

as (ϑ-P) with non-negativity constraints, we see that $\mathbf{X} := \frac{1}{\operatorname{tr}(\mathbf{R})}\mathbf{R}$ is a feasible solution and as desired,

$$\vartheta^-(\overline{G}) \geq \langle \mathbf{X}, \mathbf{J}\rangle = \frac{\langle \mathbf{R}, \mathbf{J}\rangle}{\operatorname{tr}(\mathbf{R})} = \frac{n}{\hat{\vartheta}(G)}.$$

□

Theorem 6.24:
For vertex-transitive $G \in \mathcal{G}_n$, we have

$$\vartheta^+(G) = \hat{\vartheta}(G).$$

In particular, $\omega(G) = \hat{\vartheta}(\overline{G}) = \chi(G)$ for vertex-transitive perfect $G \in \mathcal{G}_n$.

Proof. Using Theorem 6.19, Lemma 6.23 and Theorem 6.22, we see that

$$\vartheta^+(G) \geq \hat{\vartheta}(G) \geq \frac{n}{\vartheta^-(\overline{G})} = \vartheta^+(G)$$

whenever $G \in \mathcal{G}_n$ is vertex-transitive. □

Since vertex-transitive graphs are known to be examples for which the relaxations $\vartheta(G)$ and $\vartheta^+(G)$ perform badly, it is surprising to see that we can recover an analogue of Theorem 6.16 for $\hat{\vartheta}(G)$ in this case. The implication is that the advantage of $\vartheta(G)$ over $\hat{\vartheta}(G)$ is related to exploiting the lacking symmetries of a given graph.

Moving up in the hierarchy

We close this section by noting that even when we refine $\hat{\vartheta}(G)$ by using the computationally expensive $\mathrm{RProM}_k^n(S_G)$ instead of $\mathrm{RProM}_k^n(S_G)^0$, we still cannot guarantee a lower-bound of $\vartheta(G)$. To this end, let

$$\hat{\vartheta}'(G) := \min\left\{k \in \mathbb{R}_+^n \;\middle|\; \mathrm{RProM}_k^n(S_G) \neq \emptyset\right\}$$

be the corresponding strengthening of $\hat{\vartheta}(G)$.

Empirical evidence points to the fact that $\hat{\vartheta}'(G)$ yields the correct answer for the graph class $G(n_1, n_2)$ used as counter example in Theorem 6.20. However, we can easily generalize this counter example to find another class of perfect graphs for which we get results where $\vartheta(G) > \hat{\vartheta}'(G)$. Let $n_1, n_2, n_3 \in \mathbb{N}$ such that

$$G(n_1, n_2, n_3) = K_{n_1} \cup K_{n_2} \cup K_{n_3}$$

is the union of three complete graphs. Again, this class of graphs is perfect, and as such we get

$$\vartheta(\overline{G(n_1, n_2, n_3)}) = \omega(G(n_1, n_2, n_3)) = \max\{n_1, n_2, n_3\}.$$

Continuing in this fashion, we also define the class of perfect graphs given by the union of a complete graph on n_1 nodes together with m isolated nodes, or more formally

$$G(n_1, \boldsymbol{e}_m) = K_{n_1} \cup \left(\bigcup_{i \in [m]} K_1\right),$$

where

$$\vartheta(\overline{G(n_1, \boldsymbol{e}_m)}) = \omega(G(n_1, \boldsymbol{e}_m)) = n_1.$$

Fixing the number of total nodes to 9, the tables in Figure 6.2 and 6.3 show how the various relaxations behave. While there is a definite increase in quality by going from

n_1	n_2	n_3	$\hat{\vartheta}$	$\hat{\vartheta}'$	$\vartheta = \chi$
3	3	3	3	3	3
4	3	2	3.222	3.968	4
4	4	1	3.666	4	4
5	2	2	3.666	4.972	5
5	3	1	3.888	4.983	5
6	2	1	4.555	5.983	6
7	1	1	5.666	6.985	7

Table 6.2: Relaxations for $G(n_1, n_2, n_3)$.

n_1	m	$\hat{\vartheta}$	$\hat{\vartheta}'$	$\vartheta = \chi$
2	7	1.222	1.772	2
3	6	1.666	2.792	3
4	5	2.333	3.851	4
5	4	3.222	4.905	5
6	3	4.333	5.951	6
7	2	5.666	6.986	7
8	1	7.222	8	8

Table 6.3: Relaxations for $G(n_1, \boldsymbol{e}_m)$.

$\hat{\vartheta}(G)$ to $\hat{\vartheta}'(G)$, the difference between $\hat{\vartheta}'(G)$ and $\vartheta(G)$ grows roughly linear with the number of connected components.

The main reason for this bad performance of relaxations based on projection matrices is the fact that the relaxations are *not monotone* in terms of subgraphs. In particular, if H is a subgraph of G, then we have the implication

$$H \leq G \quad \Rightarrow \quad \vartheta(H) \leq \vartheta(G),$$

which is not true for either $\hat{\vartheta}(G)$ or $\hat{\vartheta}'(G)$. Unfortunately, it is not straightforward to add this property to the functions in question, leaving them at a inherent disadvantage compared to the original ϑ-function. In particular, this begs the question if in general, approximating matrices with binary eigenvalues is harder than approximating matrices with binary entries.

6.3.2 Euclidean k-Clustering

Surprisingly, projection matrices also occur naturally in *Euclidean clustering*, where the goal is to cluster points in euclidean space into sets such that their deviation from a central point, or centroid, is minimal. More formally, we have the following definition.

Definition 6.25:
Let $\mathbf{B} = (\boldsymbol{b}_1, \dots, \boldsymbol{b}_n) \in \mathbb{R}^{d \times n}$ and define the corresponding *squared error* set-function $\mathrm{SE}_{\mathbf{B}} : 2^{[n]} \to \mathbb{R}$ as

$$\mathrm{SE}_{\mathbf{B}}(T) = \min \left\{ \sum_{i \in T} \|\boldsymbol{b}_i - \boldsymbol{x}\|_2^2 \;\middle|\; \boldsymbol{x} \in \mathbb{R}^d \right\}.$$

Then Euclidean k-clustering is the separable partition problem given by

$$\min \left\{ \sum_{T \in \mathcal{T}} \mathrm{SE}_{\mathbf{B}}(T) \;\middle|\; \mathcal{T} \in \mathcal{P}_k^n \right\}. \tag{6.7}$$

Unfortunately, Euclidean k-clustering is known to be NP-hard [ADHP09]. We cite the following, stronger result regarding its approximation.

Theorem 6.26 ([ACKS15])**:**
There exists a constant $\varepsilon > 0$ such that it is NP-hard to approximate the Euclidean k-clustering problem (6.7) to a factor better than $(1+\varepsilon)$.

In alternative to (6.7), we can restate the problem by using assignment matrices via

$$\begin{aligned}\min\left\{\sum_{T\in\mathcal{T}}\mathrm{SE}_{\mathbf{B}}(T)\,\middle|\,\mathcal{T}\in\mathcal{P}_k^n\right\} &= \min\left\{\sum_{T\in\mathcal{T}}\min\left\{\sum_{i\in T}\|\boldsymbol{b}_i-\boldsymbol{x}\|_2^2\,\middle|\,\boldsymbol{x}\in\mathbb{R}^d\right\}\,\middle|\,\mathcal{T}\in\mathcal{P}_k^n\right\}\\ &= \min\left\{\sum_{T\in\mathcal{T}}\sum_{i\in T}\|\boldsymbol{b}_i-\boldsymbol{x}_T\|_2^2\,\middle|\,\mathcal{T}\in\mathcal{P}_k^n,\ \{\boldsymbol{x}_T\}_{T\in\mathcal{T}}\subseteq\mathbb{R}^d\right\}\\ &= \min\left\{\sum_{j\in[k]}\sum_{i\in[n]}u_{ij}\|\boldsymbol{b}_i-\boldsymbol{x}_j\|_2^2\,\middle|\,\mathbf{U}\in\mathcal{U}_{n,k},\ \mathbf{X}\in\mathbb{R}^{d\times k}\right\}.\end{aligned}$$

It is important to note that due to symmetry, the minimizers of this formulation are not unique. While we could remedy this by working with $\mathcal{U}_{n,k}^{lex}$, it would be difficult to apply MM due to the newly introduced centroid variables x_j. Instead, we will reformulate the problem based on the following lemma.

Lemma 6.27:
The function $\mathrm{SE}_{\mathbf{B}}$ has the following explicit form:

$$\mathrm{SE}_{\mathbf{B}}(T)=\sum_{i\in T}\|\boldsymbol{b}_i\|_2^2-\frac{1}{|T|}\Big\|\sum_{i\in T}\boldsymbol{b}_i\Big\|_2^2$$

In particular, the unique minimizer of $\mathrm{SE}_{\mathbf{B}}(T)$ is given by

$$\boldsymbol{x}_T^*=\frac{\sum_{i\in T}\boldsymbol{b}_i}{|T|}.$$

Proof. For a fixed $T\in 2^{[n]}$, let $g(\boldsymbol{x})=\sum_{i\in T}\|\boldsymbol{b}_i-\boldsymbol{x}\|_2^2$. Since g is convex, it suffices to find a local minimizer by finding a point where the gradient vanishes. This leads to

$$\nabla g(\boldsymbol{x})=2\sum_{i\in T}(\boldsymbol{x}-\boldsymbol{b}_i),$$

which vanishes if and only if $|T| \cdot \boldsymbol{x} = \sum_{i \in T} \boldsymbol{b}_i$. Consequently,

$$\begin{aligned} \mathrm{SE}_{\mathbf{B}}(T) = g(\boldsymbol{x}_T^*) &= \sum_{i \in T} \left\| \boldsymbol{b}_i - \frac{\sum_{i \in T} \boldsymbol{b}_i}{|T|} \right\|_2^2 \\ &= \sum_{i \in T} \left(\|\boldsymbol{b}_i\|_2^2 - \frac{2}{|T|} \langle \boldsymbol{b}_i, \sum_{j \in T} \boldsymbol{b}_j \rangle + \left\| \frac{\sum_{i \in T} \boldsymbol{b}_i}{|T|} \right\|_2^2 \right) \\ &= \sum_{i \in T} \|\boldsymbol{b}_i\|_2^2 - \frac{1}{|T|} \left\| \sum_{i \in T} \boldsymbol{b}_i \right\|_2^2. \end{aligned}$$

□

Corollary 6.28:
For each $T \in 2^{[n]}$, the unique minimizer of $\mathrm{SE}_{\mathbf{B}}(T)$ is contained in $\mathrm{conv}(\{\boldsymbol{b}_i \mid i \in [n]\})$, the convex hull of the data points. Since each vertex $\boldsymbol{b}_i$ is a minimizer of $\mathrm{SE}_{\mathbf{B}}(\{i\})$, this set coincides with the convex hull of all minimizers of $\mathrm{SE}_{\mathbf{B}}$, which shows that it has a compact description as a polytope on n vertices.

Lemma 6.27 allows us to formulate Euclidean k-clustering as a linear optimization problem over the set of combinatorial projection matrices as follows.

Theorem 6.29 ([PX05])**:**
The Euclidean k-clustering problem (6.7) is equivalent to

$$\min \left\{ \langle \mathbf{B}^\top \mathbf{B}, \mathbf{I}_n - \mathbf{R} \rangle \;\middle|\; \mathbf{R} \in \mathrm{CProM}_k^n \right\}.$$

Proof. Let $\mathcal{T} \in \mathcal{P}_k^n$ and let $\mathbf{U} = (\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k) \in \mathcal{U}_{n,k}$ be the corresponding assignment matrix. Applying Lemma 6.27 to (6.7) shows

$$\begin{aligned} \sum_{T \in \mathcal{T}} \mathrm{SE}_{\mathbf{B}}(T) &= \sum_{T \in \mathcal{T}} \sum_{i \in T} \|\boldsymbol{b}_i\|_2^2 - \sum_{T \in \mathcal{T}} \frac{1}{|T|} \left\| \sum_{i \in T} \boldsymbol{b}_i \right\|_2^2 \\ &= \langle \mathbf{B}^\top \mathbf{B}, \mathbf{I}_n \rangle - \sum_{j \in [k]} \frac{1}{\|\boldsymbol{u}_j\|_2^2} \langle \mathbf{B}^\top \mathbf{B}, \boldsymbol{u}_j \boldsymbol{u}_j^\top \rangle \\ &= \langle \mathbf{B}^\top \mathbf{B}, \mathbf{I}_n - \rho(\mathbf{U}) \rangle. \end{aligned} \qquad \begin{aligned} &= \sum_{i \in [n]} \|\boldsymbol{b}_i\|_2^2 - \sum_{j \in [k]} \frac{1}{\|\boldsymbol{u}_j\|_2^2} \|\mathbf{B}\boldsymbol{u}_j\|_2^2 \\ &= \left\langle \mathbf{B}^\top \mathbf{B}, \mathbf{I}_n - \sum_{j \in [k]} \frac{\boldsymbol{u}_j \boldsymbol{u}_j^\top}{\|\boldsymbol{u}_j\|_2^2} \right\rangle \end{aligned}$$

Since $\rho(\mathcal{U}_{n,k}) = \mathrm{CProM}_k^n$ by Theorem 6.7, the result follows. □

This formulation was already proposed in [PX05], where it was shown that the same is true for a slightly relaxed formulation of CProM_k^n, where the block inducing equations are replaced with the more general projection constraint $\mathbf{R}^2 = \mathbf{R}$.

Convexification

In [PW07], it was shown that $\mathrm{RProM}_k^{n,0}$ is a suitable relaxation for CProM_k^n and that using techniques similar to spectral clustering, one can use the solution of

$$\min \left\{ \langle \mathbf{B}^\top \mathbf{B}, \mathbf{I}_n - \mathbf{R} \rangle \;\middle|\; \mathbf{R} \in \mathrm{RProM}_k^{n,0} \right\}. \tag{6.8}$$

to arrive at a Euclidean k-clustering problem in dimension $\mathbb{R}^{k-1}$. Solving the new problem is, for the common situation where $d \gg k$, significantly easier, and it is shown that such a solution can be used to reconstruct a 2-approximation of the original problem. In particular, for the special case of $k = 2$, the resulting clustering problem can be solved in time $O(n \log n)$.

While it is possible to improve the results of (6.8) by going from $\mathrm{RProM}_k^{n,0}$ to the sharper relaxation RProM_k^n, the runtimes of other approximation algorithms give little incentives to do so. For a proper overview over viable alternatives, we direct the reader to the paper [PW07].

Chapter 7

Affine Euclidean Clustering

This chapter covers the treatment of what we call the affine Euclidean k-clustering problem by using a simplicial cover of the feasible set. This is the largest chapter of this thesis, since this problem is very general and can express many different kinds of partition problems.

In Section 7.1, we introduce the problem and show how it is an overall much harder generalization of Euclidean k-clustering. In order to tackle this problem anyway, we introduce the concept of simplicial cover in Section 7.2 and show how to use this to reformulate the problem into a much more structured form, ready to be exploited in Section 7.3 to formulate convex relaxations. At this point, we are able to relate our relaxation to existing work in Section 7.4 before we go on by showing how to extract feasible solutions in Section 7.5. After that, we show how our approach can be generalized to various variants of the problem in Section 7.6 before we finish the chapter with a discussion of applications in Section 7.7.

7.1 Overview

This section outlines the problem of affine Euclidean k-clustering and the associated difficulties in solving it.

7.1.1 Problem Formulation

In Section 6.3.2, we considered set functions $\mathrm{SE}_{\mathbf{B}}$, which mapped a subset of given points to their least-square centroid as partition criterion. In this chapter, we will examine a generalization of these functions with increased expressiveness and start by introducing them.

Definition 7.1:
A set function $\mathrm{SE}_{\mathbf{B}}^{\mathcal{A}} : 2^{[n]} \to \mathbb{R}$ is called an *affine squared error function*, if there are

matrices $\mathcal{A} := \{\mathbf{A}_i\}_{i\in[n]} \subseteq \mathbb{R}^{l\times d}$ and $\mathbf{B} = (\boldsymbol{b}_1, \dots, \boldsymbol{b}_n) \in \mathbb{R}^{l\times n}$ such that

$$\mathrm{SE}_{\mathbf{B}}^{\mathcal{A}}(T) = \min\left\{ \sum_{i\in T} \|\mathbf{A}_i\boldsymbol{x} - \boldsymbol{b}_i\|_2^2 \;\middle|\; \boldsymbol{x} \in \mathbb{R}^d \right\}. \tag{7.1}$$

Setting $d = l$ and using $\mathbf{A}_i = \mathbf{I}_d$ for all $i \in [n]$, we recover $\mathrm{SE}_{\mathbf{B}}$. However, compared to $\mathrm{SE}_{\mathbf{B}}$, the map $\mathrm{SE}_{\mathbf{B}}^{\mathcal{A}}$ is more complex, as it can in general not be geometrically understood as computing a centroid of the point set $\{\boldsymbol{b}_i\}_{i\in T}$. Following Section 6.3.2, we define the *affine Euclidean k-clustering problem* as the separable partition problem given by

$$\min\left\{ \sum_{T\in\mathcal{T}} \mathrm{SE}_{\mathbf{B}}^{\mathcal{A}}(T) \;\middle|\; \mathcal{T} \in \mathcal{P}_k^n \right\}. \tag{7.2}$$

Alternatively, we can restate the problem by using assignment matrices via

$$\begin{aligned}
\min\left\{ \sum_{T\in\mathcal{T}} \mathrm{SE}_{\mathbf{B}}^{\mathcal{A}}(T) \;\middle|\; \mathcal{T} \in \mathcal{P}_k^n \right\} &= \min\left\{ \sum_{T\in\mathcal{T}} \min\left\{ \sum_{i\in T} \|\mathbf{A}_i\boldsymbol{x} - \boldsymbol{b}_i\|_2^2 \;\middle|\; \boldsymbol{x} \in \mathbb{R}^d \right\} \;\middle|\; \mathcal{T} \in \mathcal{P}_k^n \right\} \\
&= \min\left\{ \sum_{T\in\mathcal{T}} \sum_{i\in T} \|\mathbf{A}_i\boldsymbol{x}_T - \boldsymbol{b}_i\|_2^2 \;\middle|\; \mathcal{T} \in \mathcal{P}_k^n,\ \{\boldsymbol{x}_T\}_{T\in\mathcal{T}} \subseteq \mathbb{R}^d \right\} \\
&= \min\left\{ \sum_{j\in[k]} \sum_{i\in[n]} u_{ij}\|\mathbf{A}_i\boldsymbol{x}_j - \boldsymbol{b}_i\|_2^2 \;\middle|\; \mathbf{U} \in \mathcal{U}_{n,k},\ \mathbf{X} \in \mathbb{R}^{d\times k} \right\}.
\end{aligned}$$

For later reference, the formulation

$$\min\left\{ \sum_{j\in[k]} \sum_{i\in[n]} u_{ij}\|\mathbf{A}_i\boldsymbol{x}_j - \boldsymbol{b}_i\|_2^2 \;\middle|\; \mathbf{U} \in \mathcal{U}_{n,k}, \mathbf{X} \in \mathbb{R}^{d\times k} \right\}. \tag{7.3}$$

will be referred to as *affine Euclidean k-clustering* from now on and will be the focus of this chapter. We start with a basic observation.

Lemma 7.2:
Problem (7.3) is a polynomial optimization problem.

Proof. Since

$$\|\mathbf{A}_i\boldsymbol{x}_j - \boldsymbol{b}_i\|^2 = \boldsymbol{x}_j^\top(\mathbf{A}_i^\top\mathbf{A}_i)\boldsymbol{x}_j - 2(\boldsymbol{b}_i^\top\mathbf{A}_i)\boldsymbol{x}_j + \|\boldsymbol{b}_i\|_2^2$$

and

$$u_{ij} \in \{0,1\} \quad \Leftrightarrow \quad u_{ij}(1-u_{ij}) = 0$$

for all $i \in [n]$ and $j \in [k]$, we need to optimize a polynomial over a real variety. □

We point out that again due to symmetry, the minimizers of (7.3) are not unique, since any permutation acting on the columns of both $\mathbf{U}$ and $\mathbf{X}$ simultaneously will leave the objective value invariant.

While we could hope on using $\mathcal{U}_{n,k}^{lex}$ to remedy this, it would be difficult to apply MM due to the newly introduced centroid variables $\mathbf{X}$. In particular, the total degree of the objective function is 3 due to $\mathbf{X}$, so that we would need the prohibitive big second stage of MM, which is not available for practical computations. Furthermore, it is hard to extract feasible solutions from lower levels of MM, and the symmetrical structure of the problem makes this even harder.

Instead, a better strategy would be to first reduce the problem to a more compact reformulation. To this end, it makes sense to follow Section 6.3.2 again and to try finding a closed form for $\mathrm{SE}_{\mathbf{B}}^{\mathcal{A}}$. This leads to the following lemma.

Lemma 7.3:
The minimizers of $\mathrm{SE}_{\mathbf{B}}^{\mathcal{A}}(T)$ *are given as the solution space of the linear system*

$$\sum_{i\in T} \mathbf{A}_i^\top \boldsymbol{b}_i = \left(\sum_{i\in T} \mathbf{A}_i^\top \mathbf{A}_i\right) \boldsymbol{x}.$$

Proof. For a fixed set $T \in 2^{[n]}$, let $g(x) = \sum_{i\in T} \|\mathbf{A}_i\boldsymbol{x} - \boldsymbol{b}_i\|_2^2$. Since g is continuous and convex, it suffices to find a local minimizer by finding a point where the gradient vanishes. This leads to

$$\nabla g(\boldsymbol{x}) = 2\sum_{i\in T} \mathbf{A}_i^\top(\mathbf{A}_i\boldsymbol{x} - \boldsymbol{b}_i),$$

which vanishes if and only if x solves the linear system $\sum_{i\in T} \mathbf{A}_i^\top \boldsymbol{b}_i = \left(\sum_{i\in T} \mathbf{A}_i^\top \mathbf{A}_i\right) \boldsymbol{x}$. □

Unfortunately, the introduction of the linear maps $\mathbf{A}_i$ makes it very difficult to give an explicit formula for the minimizer of $\mathrm{SE}_{\mathbf{B}}^{\mathcal{A}}$, so that we can not follow the approach from Section 6.3.2. Before pursuing a different line of thought in the following section, we finish with a theorem that highlights the potential complexity of $\mathrm{SE}_{\mathbf{B}}^{\mathcal{A}}$.

Theorem 7.4:
We can find matrices $\{\mathbf{A}_i\}_{i\in[n]}$ *and vectors* $\{\boldsymbol{b}_i\}_{i\in[n]}$ *such that the convex hull of the minimizers of* $\mathrm{SE}_{\mathbf{B}}^{\mathcal{A}}(T)$ *is a polytope with an exponential number of vertices and facets in the input size.*

Proof. We will explicitly construct such an instance for a given number n of pairs $(\mathbf{A}, \boldsymbol{b})$. Before we start, first note that if we partition

$$\mathbf{A}_i = \begin{pmatrix} \mathbf{A}_{i,1} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{i,2} \end{pmatrix}, \quad \boldsymbol{b}_i = \begin{pmatrix} \boldsymbol{b}_{i,1} \\ \boldsymbol{b}_{i,2} \end{pmatrix}, \quad \boldsymbol{x} = \begin{pmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{pmatrix}$$

into blocks of consistent dimension, we can separate

$$\mathrm{SE}_{\mathbf{B}}^{\mathcal{A}}(T) = \mathrm{SE}_{\mathbf{B}_1}^{\mathcal{A}_1}(T) + \mathrm{SE}_{\mathbf{B}_2}^{\mathcal{A}_2}(T)$$

where $\mathcal{A}_j = \{\mathbf{A}_{i,j}\}_{i\in[n]} \subseteq \mathbb{R}^{l_j \times d_j}$ and $\mathbf{B}_j = \left(\boldsymbol{b}_{1,j}, \ldots, \boldsymbol{b}_{n,j}\right) \in \mathbb{R}^{l_j \times n}$ for $j = 1, 2$ to compute $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ separately. Furthermore, if we can partition $[n] = T_1 \cup T_2$ such that

$$\begin{aligned} i \in T_1 \quad &\Rightarrow \quad (\mathbf{A}_{i,2}, \boldsymbol{b}_{i,2}) = (\mathbf{0}, \mathbf{0}), \\ i \in T_2 \quad &\Rightarrow \quad (\mathbf{A}_{i,1}, \boldsymbol{b}_{i,1}) = (\mathbf{0}, \mathbf{0}), \end{aligned}$$

then

$$\arg\min\left(\mathrm{SE}_{\mathbf{B}}^{\mathcal{A}}(T)\right) = \arg\min\left(\mathrm{SE}_{\mathbf{B}_1}^{\mathcal{A}_1}(T \cap T_1)\right) \times \arg\min\left(\mathrm{SE}_{\mathbf{B}_2}^{\mathcal{A}_2}(T \cap T_2)\right).$$

It thus suffices to construct one instance that results in an exponential number of vertices and one that results in an exponential number of facets to show the result, since vertices and facets are preserved under taking the convex hull of their Cartesian product.

For the first instance, we create the n-dimensional unit-cube C^n by setting $d = n$, $l = 1$ and

$$(\mathbf{A}_i, \boldsymbol{b}_i) = (\boldsymbol{e}_i^\top, 1) \in \mathbb{R}^{1\times n} \times \mathbb{R} \quad \forall i \in [n].$$

Then for any set $T \in 2^{[n]}$, we can set $\boldsymbol{x}_T = \boldsymbol{e}_T$ to see that since

$$\sum_{i\in T} \mathbf{A}_i^\top \boldsymbol{b}_i = \sum_{i\in T} \boldsymbol{e}_i \cdot 1 = \sum_{i\in T} \boldsymbol{e}_i \langle \boldsymbol{e}_i, \boldsymbol{e}_T\rangle = \left(\sum_{i\in T} \boldsymbol{e}_i \boldsymbol{e}_i^\top\right) \boldsymbol{e}_T = \left(\sum_{i\in T} \mathbf{A}_i^\top \mathbf{A}_i\right) \boldsymbol{x}_T,$$

the solution $\boldsymbol{x}_T$ is optimal for $\mathrm{SE}_{\mathbf{B}}^{\mathcal{A}}(T)$ according to Lemma 7.3. This takes care of the exponential number of vertices in n, since we get a unique solution for each $T \in 2^{[n]}$, resulting in 2^n vertices.

For the second instance, we construct the n-dimensional cross polytope given by

$$(C^n)^* = \mathrm{conv}(\{\pm\boldsymbol{e}_i \mid i \in [n]\})$$

by setting $d = l = n$ and

$$(\mathbf{A}_{i,\pm}, \boldsymbol{b}_{i,\pm}) = (\mathbf{I}_n, \pm\boldsymbol{e}_i) \in \mathbb{R}^{n\times n} \times \mathbb{R}^n \quad \forall i \in [n].$$

This way, we model the special case of $\mathrm{SE}_{\mathbf{B}}$, where

$$\mathbf{B} = (+\boldsymbol{e}_1, -\boldsymbol{e}_1, \ldots, +\boldsymbol{e}_n, -\boldsymbol{e}_n) \in \mathbb{R}^{n\times 2n}.$$

By Corollary 6.28, it follows immediately that the convex hull of the minimizers of $\mathrm{SE}_{\mathbf{B}}$ is equal to $(C^n)^*$. This takes care of the exponential number of facets in n, since $(C^n)^*$ is the dual-polytope of C^n, and thus has one facet for each of the 2^n vertices of C^n.

Combining both instances through their Cartesian product results in an instance with $3n$ pairs $(\mathbf{A}, \boldsymbol{b})$ and at least 2^n vertices and facets, thus completing the proof. □

7.2 Simplicial Covers

In its general form, it seems very hard to tackle problem (7.3). Instead, we will introduce some assumptions that make the problem easier to handle, starting with a bound on the minimizers of $\mathrm{SE}_{\mathbf{B}}^{\mathcal{A}}$. We will need a definition first.

Definition 7.5 (Simplicial cover)**:**
A finite set of d-dimensional simplices $\{P_s\}_{s\in[q]}$ with disjoint interior is called a *simplicial cover* of $\mathrm{SE}_{\mathbf{B}}^{\mathcal{A}}$ if the inclusion

$$\{\boldsymbol{x}_T\}_{T\in 2^{[n]}} \subseteq \mathcal{P} = \bigcup_{s\in[q]} P_s$$

holds for the minimizers $\boldsymbol{x}_T$ of $\mathrm{SE}_{\mathbf{B}}^{\mathcal{A}}(T)$.

In the following, we will abuse notation by using $\mathcal{P}$ to denote both the set of simplices $\{P_s \mid s \in [q]\}$ as well as their union.

Remark 7.6:
The concept of simplicial cover is motivated by what is known in literature as a *simplicial complex*. A simplicial complex is a collection of simplices $\mathcal{P} = \bigcup_{s\in[q]} P_s$ with the property that whenever F_s is a face of P_s and $F_{s'}$ is a face of $P_{s'}$, their intersection $F_s \cap F_{s'}$ is a face of both P_s and $P_{s'}$, independent of the choice of $s, s' \in [q]$. In particular, such a simplicial complex defines an independence system on the set of vertices of a simplicial cover $\mathcal{P}$, where the independent sets are precisely the sets of vertices that lie on a common face of any simplex P_s. By our definition, each simplicial complex gives rise to a simplicial cover, but the reverse is not true in general, as shown by Figure 7.1.

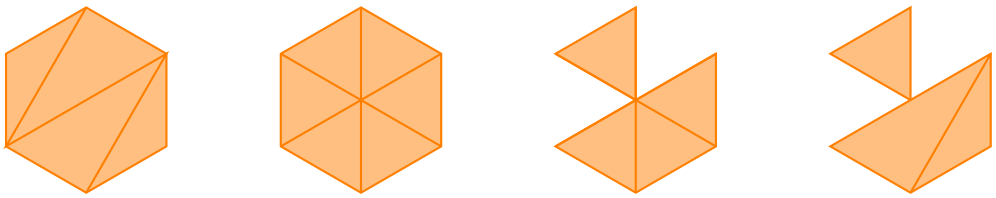

Figure 7.1: Four examples of simplicial covers. All simplicial covers but the last one are also simplicial complexes.

A natural question is to ask about the availability of such a simplicial cover. Since our goal is to use this concept to solve the optimization problem (7.3), the construction of such an simplicial cover should be much cheaper than computing each of the exponentially many minimizers $\{\boldsymbol{x}_T\}_{T\in 2^{[n]}}$ first. To this end, we will use the following underlying assumption for now.

Assumption 7.7 (Simplicial cover assumption)**:**
Given $\{(\mathbf{A}_i, \boldsymbol{b}_i)\}_{i\in[n]}$*, we can construct a simplicial cover of* $\mathrm{SE}_{\mathbf{B}}^{\mathcal{A}}$ *in polynomial time.*

The crucial property of Assumption 7.7 is not the existence but rather the efficiency of the underlying construction. In particular, a simplicial cover always exists, since each point $\boldsymbol{x}_T$ can be computed by Lemma 7.3, and the convex hull of these points is a polytope that can be turned into a simplicial complex via various methods [BEF00].

Alternatively, if a bound on the norm of all minimizers is available, then a single simplex containing all minimizers can easily be constructed. Such a bound is not unlikely to be available in applications [XWI05].

Example 7.8:
Setting each matrix $\mathbf{A}_i$ to the identity matrix, we recover the original Euclidean clustering problem (6.7). Then Assumption 7.7 is satisfied by Corollary 6.28, since

$$\max \{\|\boldsymbol{b}_i\|_2 \mid i \in [n]\}$$

is a bound for the norm of all points in the convex hull of the minimizers.

We will skip the realization of Assumption 7.7 for now and postpone a closer investigation to Section 7.7 where we give examples that satisfy the assumption naturally. For now, we proceed by introducing notation based on a fixed simplicial cover $\mathcal{P}$ for $\mathrm{SE}_{\mathbf{B}}^{\mathcal{A}}$. Let $\mathbf{V}_s \in \mathbb{R}^{d\times(d+1)}$ be the matrix whose columns denote the vertices of P_s. Then $\mathrm{conv}(\mathbf{V}_s) = P_s$ and we gather all vertices in $\mathbf{V} = (\mathbf{V}_1, \dots, \mathbf{V}_q) \in \mathbb{R}^{d\times m}$, where

$$m := \sum_{s\in[q]} |\mathbf{V}_s| = q(d+1).$$

Furthermore, we implicitly define index sets $v(s)$ for each P_s by partitioning

$$[m] = \dot{\bigcup_{s\in[q]}} v(s).$$

Then membership $\boldsymbol{x} \in \mathcal{P}$ naturally implies the existence of

$$\boldsymbol{\lambda}^\top := (\boldsymbol{\lambda}_{v(1)}^\top, \dots, \boldsymbol{\lambda}_{v(q)}^\top) \in \Delta^m$$

where

$$\boldsymbol{x} = \mathbf{V}\boldsymbol{\lambda} = \sum_{s\in[q]} \mathbf{V}_s \boldsymbol{\lambda}_{v(s)}.$$

Remark 7.9:
If the simplices P_s share common vertices, then $\mathbf{V}$ has multiple identical columns across different $\mathbf{V}_s$. This is intentional, and the potential benefit of removing multiples of these columns will be explored in Section 7.6.

Now $\mathcal{P}$ can be expressed as the linear image of a constrained standard simplex through

$$\mathcal{P} = \left\{ \boldsymbol{x} \in \mathbb{R}^d \;\middle|\; \exists \boldsymbol{\lambda} \in \Delta^m \colon \boldsymbol{x} = \mathbf{V}\boldsymbol{\lambda}, \quad \boldsymbol{\lambda}_{v(r)}\boldsymbol{\lambda}_{v(s)}^\top = \mathbf{0} \quad \forall\, r, s \in [q], r \neq s \right\}. \tag{7.4}$$

The nonlinear orthogonality constraint

$$\boldsymbol{\lambda}_{v(r)}\boldsymbol{\lambda}_{v(s)}^{\top} = \mathbf{0} \quad \forall\, r, s \in [q], r \neq s \tag{7.5}$$

ensures that exactly one $\boldsymbol{\lambda}_{v(s)}$ is nonzero, which implies $\boldsymbol{x} = \mathbf{V}\boldsymbol{\lambda} = \mathbf{V}_s\boldsymbol{\lambda}_{v(s)} \in P_s$ for some $s \in [q]$. Since $\boldsymbol{\lambda} \geq \mathbf{0}$, we can see that (7.5) is equivalent to the quadratic constraint

$$\boldsymbol{\lambda}^{\top}\Omega\boldsymbol{\lambda} = 0,$$

where

$$\Omega := (\mathbf{J}_q - \mathbf{I}_q) \otimes \mathbf{J}_{d+1} \in \{0,1\}^{m \times m}$$

induces a block structure with zero block-diagonal. This suggests to set

$$\Delta_{\Omega}^{m} := \left\{ \boldsymbol{\lambda} \in \Delta^m \;\middle|\; \boldsymbol{\lambda}^{\top}\Omega\boldsymbol{\lambda} = 0 \right\}$$

so that in particular, we can write

$$\mathcal{P} = \mathbf{V} \cdot \Delta_{\Omega}^{m}$$

as a shorthand for (7.4).

Remark 7.10:
The index sets $\{v(s) \mid s \in [q]\} \subseteq 2^{[m]}$ induce a relatively simple independence set

$$L = \bigcup_{s \in [q]} 2^{v(s)} \subseteq 2^{[m]}$$

due to their disjointness, and our choice of Ω coincides with the Ω_L matrix defined for this independence system L, which makes this problem another instance of problems constrained by independence systems.

Remark 7.11:
The parametrization (7.4) of points in $\mathcal{P}$ is unique almost everywhere. This is implicit in Theorem 2.8, which guarantees uniqueness for all interior points $\boldsymbol{x} \in \dot{\bigcup}_{s \in [q]} \operatorname{int}(P_s)$. However, since simplicial covers allow intersections of boundaries $\operatorname{bd}(P_r) \cap \operatorname{bd}(P_s)$ to be nonempty, we can not assume unique representations for boundary points once $q > 1$.

Given the new notation, we can proceed by parametrizing (7.1) in terms of (7.4). Using $\boldsymbol{x} = \mathbf{V}\boldsymbol{\lambda}$, we can homogenize the objective by using $1 = \langle \boldsymbol{\lambda}, \boldsymbol{e} \rangle$ to end up with

$$\|\mathbf{A}_i\boldsymbol{x} - \boldsymbol{b}_i\|_2^2 = \langle \boldsymbol{\lambda}^j, \mathbf{C}_i\boldsymbol{\lambda}^j \rangle,$$

where

$$\mathbf{C}_i := \mathbf{V}^{\top}\mathbf{A}_i^{\top}\mathbf{A}_i\mathbf{V} - (\boldsymbol{e}\boldsymbol{b}_i^{\top}\mathbf{A}_i\mathbf{V} + \mathbf{V}^{\top}\mathbf{A}_i^{\top}\boldsymbol{b}_i\boldsymbol{e}^{\top}) + \|\boldsymbol{b}_i\|_2^2 \cdot \mathbf{J}_m \quad \forall i \in [n].$$

This leads to the reformulations

$$\mathrm{SE}_{\mathbf{B}}^{\mathcal{A}}(T) = \min\left\{ \sum_{i\in T} \langle \boldsymbol{\lambda}, \mathbf{C}_i \boldsymbol{\lambda} \rangle \;\middle|\; \boldsymbol{\lambda} \in \Delta_\Omega^m \right\}$$

of (7.1) and

$$\min\left\{ \sum_{i\in[n]} \sum_{j\in[k]} u_{ij} \langle \boldsymbol{\lambda}^j, \mathbf{C}_i \boldsymbol{\lambda}^j \rangle \;\middle|\; \mathbf{U} \in \mathcal{U}_{n,k},\ \boldsymbol{\lambda}^j \in \Delta_\Omega^m \quad \forall j \in [k] \right\} \tag{R1}$$

of (7.3).

7.2.1 Separating Simplicial Covers

So far, the goal of every chapter has been to reformulate the partition problems by replacing the assignment matrices with a class of matrices that is better suited to deal with the underlying structure of the problem at hand.

We will continue to follow this paradigm and start by noting that in the context of (R1), the assignment matrix $\mathbf{U}$ can be interpreted as assigning an optimal minimizer of $\mathrm{SE}_{\mathbf{B}}^{\mathcal{A}}$ to each data point. To see this, consider the equation

$$\sum_{j\in[k]} u_{ij} \langle \boldsymbol{\lambda}^j, \mathbf{C}_i \boldsymbol{\lambda}^j \rangle = \left\langle \Big(\sum_{j\in[k]} u_{ij} \boldsymbol{\lambda}^j \Big), \mathbf{C}_i \Big(\sum_{j\in[k]} u_{ij} \boldsymbol{\lambda}^j \Big) \right\rangle,$$

which holds since each row of $\mathbf{U}$ is a binary unit vector. Letting $\boldsymbol{\lambda}_i(\mathbf{U}) := \sum_{j\in[k]} u_{ij} \boldsymbol{\lambda}^j$ for each $i \in [n]$ then shows that

$$\mathbf{U} \in \mathcal{U}_{n,k} \;\Leftrightarrow\; \{\boldsymbol{\lambda}_i(\mathbf{U}) \mid i \in [n]\} = \{\boldsymbol{\lambda}^j \mid j \in [k]\}.$$

In particular, (R1) can be restated as

$$\min\left\{ \sum_{i\in[n]} \langle \boldsymbol{\lambda}_i, \mathbf{C}_i \boldsymbol{\lambda}_i \rangle \;\middle|\; \boldsymbol{\lambda}_i \in \{\boldsymbol{\lambda}^j\}_{j\in[k]} \subseteq \Delta_\Omega^m \quad \forall i \in [n] \right\}, \tag{7.6}$$

where $\boldsymbol{\lambda}_i(\mathbf{U})$ is replaced by a new vector $\boldsymbol{\lambda}_i$ subject to the membership constraint $\boldsymbol{\lambda}_i \in \{\boldsymbol{\lambda}^j\}_{j\in[k]}$.

At this point, we want to recall again that while it is important to model this membership constraint in a more tractable formulation, using the set $\mathcal{U}_{n,k}$ introduced the inherent problematic symmetries mentioned in Section 4.2.1 by inducing an arbitrary order on the set $\{\boldsymbol{\lambda}^j\}_{j\in[k]}$.

For this reason, our goal is to propose a different, symmetry-free way to formalize this membership, and we will use the following property as a starting point.

Definition 7.12 (Separating simplicial cover)**:**
Let $\mathcal{P} = \bigcup_{s\in[q]} P_s$ be a simplicial cover of $\mathrm{SE}_{\mathbf{B}}^{\mathcal{A}}$. Then a set of points $X \subseteq \mathcal{P}$ is called *separated* by $\mathcal{P}$ if

$$|X \cap P_s| \leq 1 \quad \forall s \in [q].$$

Likewise, a set of parameters $\Lambda \subseteq \Delta_\Omega^m$ is *separated* by $\mathcal{P}$ if their image under $\mathbf{V}$ is contained in $\mathcal{P}$ and separated by $\mathcal{P}$.

In addition, let $\boldsymbol{x}_T$ be the minimizer of $\mathrm{SE}_{\mathbf{B}}^{\mathcal{A}}(T)$ for all $T \in 2^{[n]}$. Then we say $\mathcal{P}$ *separates* $\mathrm{SE}_{\mathbf{B}}^{\mathcal{A}}$ if there is an optimal solution $\mathcal{T}$ of (7.2) such that $\{\boldsymbol{x}_T \mid T \in \mathcal{T}\}$ is separated by $\mathcal{P}$.

The idea behind separating simplicial covers is to guarantee pairwise orthogonality of the vectors $\boldsymbol{\lambda}^j$ in (7.6), as can be seen in the following lemma.

Lemma 7.13:
Let $\mathcal{P} = \bigcup_{s\in[q]} P_s$ be a simplicial cover of $\mathrm{SE}_{\mathbf{B}}^{\mathcal{A}}$. For a set $X \subseteq \mathcal{P}$, let $\Lambda \subseteq \Delta_\Omega^m$ be its representation in (7.4). Then X is separated by $\mathcal{P}$ if and only if for each pair $\boldsymbol{\lambda} \neq \boldsymbol{\lambda}' \in \Lambda$,

$$\boldsymbol{\lambda}'_{v(s)}\boldsymbol{\lambda}_{v(s)}^\top = \mathbf{0} \quad \forall s \in [q]. \tag{7.7}$$

In particular, each pair in $\boldsymbol{\lambda} \neq \boldsymbol{\lambda}' \in \Lambda$ is coordinatewise orthogonal and has disjoint support.

Proof. Let $\boldsymbol{x} \neq \boldsymbol{x}' \in \mathcal{P}$ with $\boldsymbol{x} = \mathbf{V}\boldsymbol{\lambda}$ and $\boldsymbol{x}' = \mathbf{V}\boldsymbol{\lambda}'$. Then for some fixed $t, t' \in [q]$,

$$\boldsymbol{x} \in P_t \;\Leftrightarrow\; \langle \boldsymbol{\lambda}_{v(s)}, \boldsymbol{e}\rangle = \delta_{st} \quad \text{and} \quad \boldsymbol{x}' \in P_{t'} \;\Leftrightarrow\; \langle \boldsymbol{\lambda}'_{v(s)}, \boldsymbol{e}\rangle = \delta_{st'}$$

for all $s \in [q]$. Now due to non-negativity, each equation of (7.7) can be equivalently expressed as

$$\begin{aligned}\boldsymbol{\lambda}'_{v(s)}\boldsymbol{\lambda}_{v(s)}^\top = \mathbf{0} \;&\Leftrightarrow\; 0 = \langle \boldsymbol{\lambda}'_{v(s)}\boldsymbol{\lambda}_{v(s)}^\top, \mathbf{J}_{d+1}\rangle = \langle \boldsymbol{\lambda}'_{v(s)}, \boldsymbol{e}\rangle\langle \boldsymbol{\lambda}_{v(s)}, \boldsymbol{e}\rangle = \delta_{st'}\delta_{st} \\ &\Leftrightarrow\; (s \neq t) \vee (s \neq t').\end{aligned}$$

Then letting s run through $[q]$ shows that (7.7) is equivalent to $t \neq t'$, which means that $\boldsymbol{x}$ and $\boldsymbol{x}'$ are separated by $\mathcal{P}$. □

A direct consequence of Lemma 7.13 is that if $\mathcal{P}$ separates $\mathrm{SE}_{\mathbf{B}}^{\mathcal{A}}$, then (7.7) necessarily holds for an optimal solution of (R1). Furthermore, its simple observation implies that we can express the membership constraint in (7.6) with linear inequalities and quadratic constraints.

Lemma 7.14:
Let $\mathcal{P}$ be a simplicial cover for $\mathrm{SE}_{\mathbf{B}}^{\mathcal{A}}$ and let

$$\boldsymbol{\lambda}_* := \sum_{j\in[k]} \boldsymbol{\lambda}^j \tag{7.8}$$

for a separated set $\{\boldsymbol{\lambda}^j\}_{j\in[k]}$ *feasible for* (R1)*. Then the equivalence*

$$\boldsymbol{\lambda} \in \{\boldsymbol{\lambda}^j\}_{j\in[k]} \quad \Leftrightarrow \quad \boldsymbol{\lambda} \leq \boldsymbol{\lambda}_* \tag{7.9}$$

holds for $\boldsymbol{\lambda} \in \Delta_\Omega^m$.

Proof. Implication " $\Rightarrow$ " is straightforward. Consider the reverse direction for $\boldsymbol{\lambda} \in \Delta_\Omega^m$ and let $s_j \in [q]$ for $j \in [k]$ such that $\operatorname{supp}(\boldsymbol{\lambda}^j) \subseteq v(s_j)$. Then it follows from $\boldsymbol{\lambda}^\top \Omega \boldsymbol{\lambda} = 0$, $\mathbf{0} \leq \boldsymbol{\lambda} \leq \boldsymbol{\lambda}_*$ and Lemma 7.13 that

$$\operatorname{supp}(\boldsymbol{\lambda}) \subseteq v(s^*) \subseteq \operatorname{supp}(\boldsymbol{\lambda}_*) = \dot{\bigcup_{j\in[k]}} \operatorname{supp}(\boldsymbol{\lambda}^j) = \dot{\bigcup_{j\in[k]}} v(s_j),$$

which implies $s^* = s_{j'}$ for some $j' \in [k]$. Therefore,

$$\mathbf{0} \leq \boldsymbol{\lambda} \leq \boldsymbol{\lambda}^{j'} \quad \text{and} \quad \langle \boldsymbol{\lambda}, \boldsymbol{e} \rangle = 1 = \langle \boldsymbol{\lambda}^{j'}, \boldsymbol{e} \rangle,$$

which is only possible if $\boldsymbol{\lambda} = \boldsymbol{\lambda}^{j'}$. □

Lemma 7.14 reduces the membership in $\{\boldsymbol{\lambda}^j\}_{j\in k}$ to a more tractable relation involving its sum $\boldsymbol{\lambda}_*$. The idea is that if we can also give a tractable characterization of those $\boldsymbol{\lambda}_*$ that correspond to a sum of $\{\boldsymbol{\lambda}^j\}_{j\in[k]}$ satisfying the orthogonality constraint (7.7), then we can restate (R1) without the assignment matrices when $\mathrm{SE}_{\mathbf{B}}^{\mathcal{A}}$ is separated by $\mathcal{P}$. Fortunately, we can give such a tractable characterization, as is shown in the following lemma.

Lemma 7.15:
Let

$$\mathcal{L} := \left\{ \{\boldsymbol{\lambda}^j\}_{j\in[k]} \subseteq \Delta_\Omega^m \;\middle|\; (7.7) \text{ holds} \right\}$$

and

$$\mathcal{L}' := \left\{ \boldsymbol{\lambda} \in k \cdot \Delta^m \;\middle|\; \langle \boldsymbol{\lambda}_{v(s)}, \boldsymbol{e} \rangle \boldsymbol{\lambda}_{v(s)} = \boldsymbol{\lambda}_{v(s)} \quad \forall s \in [q] \right\}. \tag{7.10}$$

Then $\mathcal{L} \leftrightarrow \mathcal{L}'$ *through the bijection* $\phi : \mathcal{L} \to \mathcal{L}'$ *given by*

$$\phi(\{\boldsymbol{\lambda}^j\}_{j\in[k]}) := \sum_{j\in[k]} \boldsymbol{\lambda}^j.$$

Proof. To verify that ϕ is well-defined, we recall from the proof of Lemma 7.13 that

$$\langle \boldsymbol{\lambda}^j_{v(s)}, \boldsymbol{e} \rangle = \delta_{s,s_j}$$

for some $s_j \in [q]$, and that the s_j are pairwise distinct. In particular, the identity

$$\delta_{s,s_j} \boldsymbol{\lambda}^j_{v(s)} = \boldsymbol{\lambda}^j_{v(s)}$$

holds for all $j \in [k]$ and all $s \in [q]$. Then

$$\left\langle \sum_{j\in[k]} \boldsymbol{\lambda}^j_{v(s)}, \boldsymbol{e} \right\rangle \left(\sum_{j\in[k]} \boldsymbol{\lambda}^j_{v(s)} \right) = \left(\sum_{j\in[k]} \delta_{s,s_j} \right) \left(\sum_{j\in[k]} \delta_{s,s_j} \boldsymbol{\lambda}^j_{v(s)} \right) = \sum_{j\in[k]} \delta_{s,s_j} \boldsymbol{\lambda}^j_{v(s)} = \sum_{j\in[k]} \boldsymbol{\lambda}^j_{v(s)}$$

since $\delta_{s,s_j}\delta_{s,s_{j'}} = \delta_{j,j'}\delta_{s,s_j}$, and so ϕ is well-defined.

We proceed by constructing an inverse-function $\psi : \mathcal{L}' \to \mathcal{L}$, so let $\boldsymbol{\lambda} \in \mathcal{L}'$. By taking the scalar product with $\boldsymbol{e}$ on each of the defining equations in (7.10), we see the membership $\langle \boldsymbol{\lambda}_{v(s)}, \boldsymbol{e} \rangle \in \{0, 1\}$ for all $s \in [q]$. By definition, there is a set $\{s_j\}_{j\in[k]} \subseteq [q]$ such that $\langle \boldsymbol{\lambda}_{v(s_j)}, \boldsymbol{e} \rangle = 1$ for all $j \in [k]$. Now define vectors $\{\boldsymbol{\lambda}^j\}_{j\in[k]}$ according to

$$\boldsymbol{\lambda}^j_{v(s)} := \begin{cases} \boldsymbol{\lambda}_{v(s)} & \text{if } s = s_j, \\ \mathbf{0} & \text{else} \end{cases} \qquad \forall s \in [q], \quad \forall j \in [k]$$

and set $\psi(\boldsymbol{\lambda}) := \{\boldsymbol{\lambda}^j\}_{j\in[k]}$. It is easy to check that ψ is well-defined. Now for $\boldsymbol{\lambda} \in \mathcal{L}'$ one has

$$\phi(\psi(\boldsymbol{\lambda}))_{v(s)} = \sum_{j\in[k]} \boldsymbol{\lambda}^j_{v(s)} = \begin{cases} \boldsymbol{\lambda}_{v(s)} & \text{if } s \in \{s_j\}_{j\in[k]}, \\ \mathbf{0} = \boldsymbol{\lambda}_{v(s)} & \text{else,} \end{cases}$$

which shows $\phi \circ \psi = \mathrm{Id}_{\mathcal{L}'}$.

For $\{\boldsymbol{\lambda}^j\}_{j\in[k]} \in \mathcal{L}$, let $\boldsymbol{\lambda} = \phi(\{\boldsymbol{\lambda}^j\}_{j\in[k]})$. Then we can choose $\{s_j\}_{j\in[k]}$ such that

$$1 = \langle \boldsymbol{\lambda}_{v(s_j)}, \boldsymbol{e} \rangle = \sum_{j'\in[k]} \langle \boldsymbol{\lambda}^{j'}_{v(s_j)}, \boldsymbol{e} \rangle = \langle \boldsymbol{\lambda}^{j}_{v(s_j)}, \boldsymbol{e} \rangle$$

by Lemma 7.13, which shows $\psi \circ \phi = \mathrm{Id}_{\mathcal{L}}$. □

With all the tools available from the preceding results, we can restate the variant (7.6) of (R1) as the following, symmetry-free polynomial optimization problem

$$\min \left\{ \sum_{i\in[n]} \langle \boldsymbol{\lambda}_i, \mathbf{C}_i \boldsymbol{\lambda}_i \rangle \;\middle|\; \begin{array}{rl} \boldsymbol{\lambda}_* \in k \cdot \Delta^m, & \\ \langle (\boldsymbol{\lambda}_*)_{v(s)}, \boldsymbol{e} \rangle (\boldsymbol{\lambda}_*)_{v(s)} = (\boldsymbol{\lambda}_*)_{v(s)} & \forall s \in [q], \\ \boldsymbol{\lambda}_i \in \Delta^m_\Omega, & \forall i \in [n], \\ \boldsymbol{\lambda}_i \leq \boldsymbol{\lambda}_* & \forall i \in [n] \end{array} \right\}. \tag{R2}$$

Formally, we have the following connection between (R1) and (R2).

Theorem 7.16:
Problem (R2) computes the optimal separated solution of (R1). In particular, both problems are equivalent if and only if $\mathcal{P}$ separates $\mathrm{SE}^{\mathcal{A}}_{\mathbf{B}}$.

Proof. Starting from (7.6), the optimal separated solution of (R1) is given as

$$\min\left\{\sum_{i\in[n]}\langle\boldsymbol{\lambda}_i,\mathbf{C}_i\boldsymbol{\lambda}_i\rangle \;\middle|\; \boldsymbol{\lambda}_i\in\{\boldsymbol{\lambda}^j\}_{j\in[k]}\ \ \forall i\in[n],\ \ \{\boldsymbol{\lambda}^j\}_{j\in[k]}\in\mathcal{L}\right\}$$

by Lemma 7.13. Applying Lemma 7.14, this turns into

$$\min\left\{\sum_{i\in[n]}\langle\boldsymbol{\lambda}_i,\mathbf{C}_i\boldsymbol{\lambda}_i\rangle \;\middle|\; \boldsymbol{\lambda}_i\leq\sum_{j\in[k]}\boldsymbol{\lambda}^j\ \ \forall i\in[n],\ \{\boldsymbol{\lambda}^j\}_{j\in[k]}\in\mathcal{L}\right\},$$

which in turn is equal to (R2) given as

$$\min\left\{\sum_{i\in[n]}\langle\boldsymbol{\lambda}_i,\mathbf{C}_i\boldsymbol{\lambda}_i\rangle \;\middle|\; \boldsymbol{\lambda}_i\leq\boldsymbol{\lambda}_*\ \ \forall i\in[n],\ \boldsymbol{\lambda}_*\in\mathcal{L}'\right\}$$

by Lemma 7.15. By definition, there is an optimal separated solution of (R1) if and only if $\mathcal{P}$ separates $\mathrm{SE}_{\mathbf{B}}^{\mathcal{A}}$, so the rest of the theorem follows. □

Remark 7.17:
While the separation assumption in Theorem 7.16 makes the choice of $\mathcal{P}$ harder when we want to find the global optimum of (7.2), it can also be used as a powerful modelling paradigm. In particular, we can construct $\mathcal{P}$ to search for the optimal solution in predetermined regions, which may be desirable in practice. For example, various facility location problems may be modelled this way.

7.3 Convexification

The goal of this section is to construct convex relaxations of (R2).

Description of (R2) as semi-algebraic set

Before we attempt to relax anything, it is necessary to fix an explicit list of polynomial inequalities to describe (R2). Given the preceding discussion, we will use the following system, which is taken directly from (R2).

$$\min_{\boldsymbol{\lambda}} \sum_{i\in[n]} \langle \boldsymbol{\lambda}_i, \mathbf{C}_i \boldsymbol{\lambda}_i \rangle \quad s.t. \tag{7.11a}$$

$$\langle (\boldsymbol{\lambda}_*)_{v(s)}, \boldsymbol{e} \rangle (\boldsymbol{\lambda}_*)_{v(s)} = (\boldsymbol{\lambda}_*)_{v(s)} \qquad \forall s \in [q] \tag{7.11b}$$

$$\langle \boldsymbol{\lambda}_*, \boldsymbol{e} \rangle = k \tag{7.11c}$$

$$\langle \boldsymbol{\lambda}_i, \boldsymbol{e} \rangle = 1 \qquad \forall i \in [n] \tag{7.11d}$$

$$(\boldsymbol{\lambda}_*)_{v(s)} (\boldsymbol{\lambda}_i)_{v(s)}^\top = (\boldsymbol{\lambda}_i)_{v(s)} (\boldsymbol{\lambda}_i)_{v(s)}^\top \qquad \forall s \in [q] \quad \forall i \in [n] \tag{7.11e}$$

$$(\boldsymbol{\lambda}_i)_{v(s)} (\boldsymbol{\lambda}_i)_{v(t)}^\top = \mathbf{0} \qquad \forall s \neq t \in [q] \quad \forall i \in [n] \tag{7.11f}$$

$$\boldsymbol{\lambda}_* \geq \boldsymbol{\lambda}_i \geq \mathbf{0} \qquad \forall i \in [n] \tag{7.11g}$$

It is not hard to verify that excluding (7.11e), this is a reformulation of (R2). The additional equations (7.11e) are implied by Lemma 7.13 and actually redundant. However, since their degree is low and they directly reduce the number of available monomials, we use them to decrease the size of MM in what follows.

A big advantage of this description is the following property, which will be invaluable for actual computations.

Remark 7.18:
Each polynomial of (7.11) belongs to a polynomial ring of the form $\mathbb{R}[\boldsymbol{\lambda}_*, \boldsymbol{\lambda}_i]$ for some $i \in [n]$. In particular, this sparsity pattern implies that once the values for $\boldsymbol{\lambda}_*$ are fixed, the problem decomposes into multiple independent problems. This is a crucial property to speed up computations and will be exploited algorithmically in the next section.

Let (R2)$[t]$ denote the t-th stage of MM for the system (7.11). Since the problems are quickly expanding in size with growing t, it will only make sense to consider $t = 1$ in the following.

Simplifying (R2)[1]

While (R2)[1] is still prohibitively large on its own, we can simplify the formulation to end up with a much more tractable formulation that can be solved by current software. To this end, we will first explicitly write down an SDP-representation of (R2)[1] where we use the notation

$$\mathbf{M}_1(\boldsymbol{\lambda}) = \begin{pmatrix} 1 & \boldsymbol{\lambda}_1^\top & \cdots & \boldsymbol{\lambda}_n^\top & \boldsymbol{\lambda}_*^\top \\ \boldsymbol{\lambda}_1 & \boldsymbol{\Lambda}_{11} & \cdots & \boldsymbol{\Lambda}_{1n} & \boldsymbol{\Lambda}_{1*} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{\lambda}_n & \boldsymbol{\Lambda}_{n1} & \cdots & \boldsymbol{\Lambda}_{nn} & \boldsymbol{\Lambda}_{n*} \\ \boldsymbol{\lambda}^* & \boldsymbol{\Lambda}_{*1} & \cdots & \boldsymbol{\Lambda}_{*n} & \boldsymbol{\Lambda}_{**} \end{pmatrix} \succeq \mathbf{0}$$

for the moment matrix involved. We proceed by explicitly writing the first stage of MM of (R2) given by (7.11).

$$\min_{\boldsymbol{\lambda},\boldsymbol{\Lambda}} \sum_{i\in[n]} \langle \mathbf{C}_i, \boldsymbol{\Lambda}_{ii}\rangle \;\; s.t. \tag{7.12a}$$

$$(\boldsymbol{\Lambda}_{**})_{v(s)}\boldsymbol{e} = (\boldsymbol{\lambda}_*)_{v(s)} \qquad \forall s \in [q] \tag{7.12b}$$

$$\langle \boldsymbol{\lambda}_*, \boldsymbol{e}\rangle = k, \qquad \boldsymbol{\Lambda}_{j*}\boldsymbol{e} = k\boldsymbol{\lambda}_j \qquad \forall j \in [n]\cup\{*\} \tag{7.12c}$$

$$\langle \boldsymbol{\lambda}_i, \boldsymbol{e}\rangle = 1, \qquad \boldsymbol{\Lambda}_{ji}\boldsymbol{e} = \boldsymbol{\lambda}_j \qquad \forall j \in [n]\cup\{*\} \quad \forall i \in [n] \tag{7.12d}$$

$$(\boldsymbol{\Lambda}_{i*})_{v(s)} = (\boldsymbol{\Lambda}_{ii})_{v(s)} \qquad \forall s \in [q] \quad \forall i \in [n] \tag{7.12e}$$

$$\langle \boldsymbol{\Lambda}_{ii}, \Omega\rangle = 0 \qquad \forall i \in [n] \tag{7.12f}$$

$$\boldsymbol{\lambda}_* \geq \boldsymbol{\lambda}_i \geq \mathbf{0}, \qquad \boldsymbol{\Lambda}_{**} \geq \boldsymbol{\Lambda}_{*i} \geq \boldsymbol{\Lambda}_{ii} \geq \mathbf{0} \qquad \forall i \in [n] \tag{7.12g}$$

$$\mathbf{M}_1(\boldsymbol{\lambda}) \succeq \mathbf{0} \tag{7.12h}$$

This can be verified by checking that each line of (7.12) is the consequence from the corresponding line in (7.11), with the addition of (7.12h).

Unfortunately, (7.12) is not tractable in practice, since the dimension of $\mathbf{M}_1(\boldsymbol{\lambda})$ is given by $O(nm)$, which is prohibitive for SDP-solvers in general. However, (7.12) is not optimized for efficient computation at this point, and we will be able to make the formulation much more compact using the following lemmas.

Lemma 7.19:
Let $\mathbf{M}_1(\boldsymbol{\lambda}|i)$ denote the submatrix of $\mathbf{M}_1(\boldsymbol{\lambda})$ in (7.12) given by

$$\mathbf{M}_1(\boldsymbol{\lambda}|i) := \begin{pmatrix} 1 & \boldsymbol{\lambda}_i^\top & \boldsymbol{\lambda}_*^\top \\ \boldsymbol{\lambda}_i & \boldsymbol{\Lambda}_{ii} & \boldsymbol{\Lambda}_{i*} \\ \boldsymbol{\lambda}^* & \boldsymbol{\Lambda}_{*i} & \boldsymbol{\Lambda}_{**} \end{pmatrix}.$$

Then we can replace $\mathbf{M}_1(\boldsymbol{\lambda}) \succeq \mathbf{0}$ in (7.12) with

$$\mathbf{M}_1(\boldsymbol{\lambda}|i) \succeq \mathbf{0} \;\; \forall i \in [n].$$

Proof. Follows from the application of [Las15, Section 8.1] to Remark 7.18. The idea is that the underlying variables of the polynomial rings $\mathbb{R}[\boldsymbol{\lambda}_i, \boldsymbol{\lambda}_*]$ satisfy the *running intersection property*, which allows to separate them for computations. □

While this is already a significant reduction of the SDP constraint, the following lemma gives an alternative proof for the fact that we can discard the constant and linear monomials due to Theorem 2.15 or Corollary 3.3.

Lemma 7.20:
Consider a matrix $\boldsymbol{\Lambda} \succeq \mathbf{0}$ and a vector $\boldsymbol{a}$. Let $\boldsymbol{a}^\top\boldsymbol{\Lambda}\boldsymbol{a} = v$ and define $\boldsymbol{\lambda} := \boldsymbol{\Lambda}\boldsymbol{a}$. Then

$$v\boldsymbol{\Lambda} \succeq \boldsymbol{\lambda}\boldsymbol{\lambda}^\top \text{ or equivalently } \begin{pmatrix} v & \boldsymbol{\lambda}^\top \\ \boldsymbol{\lambda} & \boldsymbol{\Lambda} \end{pmatrix} \succeq \mathbf{0}. \tag{7.13}$$

Proof. Since $\mathbf{\Lambda} \succeq \mathbf{0}$, there is $\mathbf{M}$ such that $\mathbf{\Lambda} = \mathbf{M}^\top \mathbf{M}$ and consequently,

$$v = \boldsymbol{a}^\top \mathbf{\Lambda} \boldsymbol{a} = \|\mathbf{M}\boldsymbol{a}\|_2^2.$$

Then for arbitrary $\boldsymbol{x}$ we have

$$\begin{aligned}\boldsymbol{x}^\top (v\mathbf{\Lambda} - (\mathbf{\Lambda}\boldsymbol{a})(\mathbf{\Lambda}\boldsymbol{a})^\top)\boldsymbol{x} &= v \cdot \langle \mathbf{M}\boldsymbol{x}, \mathbf{M}\boldsymbol{x}\rangle - \langle \mathbf{M}\boldsymbol{x}, \mathbf{M}\boldsymbol{a}\rangle^2 \\ &= \|\mathbf{M}\boldsymbol{a}\|_2^2 \cdot \|\mathbf{M}\boldsymbol{x}\|_2^2 - |\langle \mathbf{M}\boldsymbol{x}, \mathbf{M}\boldsymbol{a}\rangle|^2 \geq 0\end{aligned}$$

where the last inequality is the Cauchy-Schwarz inequality. The equivalent second formulation of (7.13) follows from the Schur complement Lemma 2.5. □

We are now ready to simplify (7.12) in the following form.

$$\min_{\boldsymbol{\lambda}, \mathbf{\Lambda}} \sum_{i\in[n]} \langle \mathbf{C}_i, \mathbf{\Lambda}_{ii}\rangle \quad s.t. \tag{7.14a}$$

$$(\mathbf{\Lambda}_{**})_{v(s)}\boldsymbol{e} = (\mathbf{\Lambda}_{*i}\boldsymbol{e})_{v(s)} \qquad \forall s \in [q] \qquad \forall i \in [n] \tag{7.14b}$$

$$\mathbf{\Lambda}_{j*}\boldsymbol{e} = k\mathbf{\Lambda}_{ji}\boldsymbol{e} \qquad \forall j \in [n] \cup \{*\} \qquad \forall i \in [n] \tag{7.14c}$$

$$(\mathbf{\Lambda}_{i*})_{v(s)} = (\mathbf{\Lambda}_{ii})_{v(s)} \qquad \forall s \in [q] \qquad \forall i \in [n] \tag{7.14d}$$

$$\langle \mathbf{\Lambda}_{ii}, \mathbf{\Omega}\rangle = 0 \qquad \forall i \in [n] \tag{7.14e}$$

$$\langle \mathbf{\Lambda}_{ii}, \mathbf{J}\rangle = 1 \qquad \forall i \in [n] \tag{7.14f}$$

$$\mathbf{\Lambda}_{**} \succeq \mathbf{\Lambda}_{*i} \succeq \mathbf{\Lambda}_{ii} \succeq \mathbf{0} \qquad \forall i \in [n] \tag{7.14g}$$

$$\begin{pmatrix} \mathbf{\Lambda}_{ii} & \mathbf{\Lambda}_{i*} \\ \mathbf{\Lambda}_{*i} & \mathbf{\Lambda}_{**} \end{pmatrix} \succeq \mathbf{0} \qquad \forall i \in [n] \tag{7.14h}$$

Theorem 7.21:
Problem (7.14) is equivalent to (7.12).

Proof. Problem (7.14) arises from (7.12) by first applying Lemma 7.19 to break down the moment matrix and then applying Theorem 2.15 to reformulate the problem in terms of monomials of degree 2. □

Now reformulation (7.14) uses n psd. matrices of dimension $2m = 2q(d+1)$, which is still very limiting. However, we don't really exploit $\langle \mathbf{\Lambda}_{ii}, \mathbf{\Omega}\rangle = 0$, which results in a huge sparsity pattern in the block-diagonal $\mathbf{\Lambda}_{ii}$.

Based on this observation, a good approach is to drop all variables in $\mathbf{\Lambda}_{i*}$ and $\mathbf{\Lambda}_{**}$ from (7.14) that do not belong to the blockdiagonal structure induced by $\mathbf{\Omega}$. Conceptually, the intention is that we only lose information of entries in $\mathbf{\Lambda}_{**}$ that have a minor, indirect impact on $\mathbf{\Lambda}_{ii}$. On the computational side, this turns each constraint in (7.14h) into q separate psd. constraints of the form

$$\begin{pmatrix} (\mathbf{\Lambda}_{ii})_{v(s)} & (\mathbf{\Lambda}_{i*})_{v(s)} \\ (\mathbf{\Lambda}_{*i})_{v(s)} & (\mathbf{\Lambda}_{**})_{v(s)} \end{pmatrix} \succeq \mathbf{0} \quad \forall s \in [q]$$

of size $2(d+1)$. This is already much smaller, but using the fact that

$$(\mathbf{\Lambda}_{i*})_{v(s)} = (\mathbf{\Lambda}_{ii})_{v(s)},$$

we can additionally compress this to

$$(\mathbf{\Lambda}_{**})_{v(s)} \succeq (\mathbf{\Lambda}_{ii})_{v(s)} \quad \forall s \in [q]$$

as a consequence of the Schur complement Lemma 2.5.

Formally, we end up with the following smaller relaxation of (7.12).

$$\min_{\boldsymbol{\lambda},\Lambda} \sum_{i\in[n]} \sum_{s\in[q]} \langle (\mathbf{C}_i)_{v(s)}, (\mathbf{\Lambda}_{ii})_{v(s)} \rangle \quad s.t. \tag{7.15a}$$

$$\sum_{s\in[q]} \langle (\mathbf{\Lambda}_{**})_{v(s)}, \mathbf{J} \rangle = k \tag{7.15b}$$

$$\sum_{s\in[q]} \langle (\mathbf{\Lambda}_{ii})_{v(s)}, \mathbf{J} \rangle = 1 \qquad \forall i \in [n] \tag{7.15c}$$

$$(\mathbf{\Lambda}_{**})_{v(s)} \succcurlyeq (\mathbf{\Lambda}_{ii})_{v(s)} \succcurlyeq \mathbf{0} \qquad \forall s \in [q] \qquad \forall i \in [n] \tag{7.15d}$$

Theorem 7.22:
Problem (7.15) *is a relaxation of* (7.14).

Proof. The objectives of both problems are equal, since only the blocks $(\mathbf{\Lambda}_{ii})_{v(s)}$ can be nonzero due to $\langle \mathbf{\Lambda}_{ii}, \Omega \rangle = 0$. With the same reasoning, (7.15c) is just a reformulation of $\langle \mathbf{\Lambda}_{ii}, \mathbf{J} \rangle = 1$. Equation (7.15b) then follows from (7.14) as

$$k = k\langle \mathbf{\Lambda}_{ii}, \mathbf{J} \rangle = (k\boldsymbol{e}^\top \mathbf{\Lambda}_{ii})\boldsymbol{e} = (\boldsymbol{e}^\top \mathbf{\Lambda}_{*i})\boldsymbol{e} = \sum_{s\in[q]} \boldsymbol{e}_{v(s)}^\top (\mathbf{\Lambda}_{*i} e)_{v(s)} = \sum_{s\in[q]} \boldsymbol{e}^\top (\mathbf{\Lambda}_{**})_{v(s)} \boldsymbol{e}.$$

□

While relaxation (7.15) may be slightly weaker than (7.14), it is much more tractable for computation. In fact, we only need to care about $2qn$ psd. constraints for matrices of size $d+1$, which is much better than the n psd. constraints for matrices of size $2q(d+1)$ in (7.14) from a computational point of view. In particular, since the system is only weakly coupled, parallel computing schemes can be efficiently used to solve the relaxation for problems of moderate parameters (n, q, d).

7.4 Related Approaches

Now that we laid down our approach, we are able to compare it to similar approaches found in literature.

7.4.1 Moment Sequences

In the preceding sections, one main idea was (7.8), where we aggregated the optimal parameters $\boldsymbol{\lambda}^j$ into their sum $\boldsymbol{\lambda}_*$, and continued to treat $\boldsymbol{\lambda}_*$ as a separate, new variable.

One way to interpret this is as aggregation of monomials of the first degree. Then, a possible way to extend this idea is to aggregate whole moment sequences instead and work with the resulting new object. In particular, letting $\boldsymbol{y}^j \in \mathcal{N}^*(\Delta_\Omega^d)$ denote the moment sequence of $\boldsymbol{\lambda}^j$, we can define

$$\boldsymbol{y}_* = \sum_{j\in[k]} \boldsymbol{y}^j \in \mathcal{N}^*(\Delta_\Omega^d), \quad (\boldsymbol{y}_*)_0 = k,$$

to get the implication

$$\boldsymbol{y} \in \{\boldsymbol{y}^j\}_{j\in[k]} \quad \Rightarrow \quad \boldsymbol{y}, \boldsymbol{y}^* - \boldsymbol{y} \in \mathcal{N}^*(\Delta_\Omega^d), \quad y_0 = 1$$

as a weaker alternative to (7.9). Then, applying MM on the sets $\mathcal{N}^*(\Delta_\Omega^d)$ *individually* leads to the hierarchy

$$\min_{\boldsymbol{y}} \sum_{i\in[n]} L_{\boldsymbol{y}_i}(\mathbf{W}_i) \tag{7.16a}$$

$$\text{s.t.} \quad \begin{matrix} \boldsymbol{y}_i & \in \mathcal{N}_t^*(\Delta_\Omega^m), & (\boldsymbol{y}_i)_0 & = 1, & \forall i \in [n], \\ \boldsymbol{y}^* - \boldsymbol{y}_i & \in \mathcal{N}_t^*(\Delta_\Omega^m), & y_0^* - (\boldsymbol{y}_i)_0 & = k-1 & \forall i \in [n], \end{matrix} \tag{7.16b}$$

where it can be shown that for $t = 1$, this coincides with (7.15) after properly reformulating.

Remark 7.23:
(7.16) coincides with problem (21) given in [SRS15] in the respective setting.

7.4.2 Mixed Linear Regression

Conceptionally, our approach has the following properties:

(i) global variables are replaced with local estimates for each data point,

(ii) each local estimate is optimized against its data term,

(iii) constraints enforce global consistency among the estimates.

In [HJ16], a similar approach is considered, where objective and constraints are exchanged to yield the following properties:

(i) global variables are replaced with local estimates for each data point,

(ii) global consistency among the estimates is optimized,

(iii) constraints enforce optimality of each local estimate against its data term.

We will now formally describe their approach, starting with a reformulation of (7.3) given by

$$\min \left\{ \sum_{i\in[n]} \left\| \mathbf{A}_i \Big(\sum_{j\in[k]} u_{ij} \cdot \boldsymbol{x}^j \Big) - \boldsymbol{b}_i \right\|_2^2 \;\middle|\; \mathbf{U} \in \mathcal{U}_{n,k},\ \mathbf{X} \in \mathbb{R}^{d\times k} \right\}.$$

Assuming that the linear equation systems $\mathbf{A}_i\boldsymbol{x} = \boldsymbol{b}_i$ are solvable, we can set

$$\boldsymbol{x}_i(\mathbf{U}) := \sum_{j\in[k]} u_{ij} \cdot \boldsymbol{x}^j$$

and, following our approach in Subsection 7.2.1, relax it to a local estimate $\boldsymbol{x}_i$ to yield

$$\min \left\{ \sum_{i,j\in[n]} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2 \;\middle|\; \mathbf{A}_i\boldsymbol{x}_i = \boldsymbol{b}_i \ \forall i \in [n] \right\},$$

where the objective describes global coherence of the estimates. As with our approach, solving this relaxations amounts to solving a conic program, where the resulting estimates need to be clustered afterwards. In particular, this optimization problem can be formulated using the Lorentz cone $\mathcal{L}_m$.

In [HJ16], only matrices $\mathbf{A}_i = \boldsymbol{a}_i^\top$ with one column are treated, which is known in the literature as *mixed linear regression*. In this setting, the assumption of the solvability of the underlying linear equations $\langle \boldsymbol{x}, \boldsymbol{a}_i \rangle = b_i$ is always satisfied, and under certain separability assumptions on the underlying data, they show that recovery of the optimal solution is possible.

7.5 Rounding

Given a solution $(\Lambda_{**}, \Lambda_{ii})$ of (7.15), we still need to determine a proper partition of $[n]$ through a rounding procedure. Ideally, such an algorithm would aim at a high approximation *guarantee*, but both achieving and proving this can become arbitrarily hard, or may even be impossible, as shown in Theorem 6.14 for the colouring problem.

In the following, we will motivate our rounding procedure for (7.15), leaving the question about provable approximation guarantees open. Before we start with a structural lemma, we will introduce some additional definitions. Let $(\Lambda_{**}, \Lambda_{ii})$ be a solution of (7.15) and define the vectors

$$\boldsymbol{\lambda}_* := \Lambda_{**}\boldsymbol{e}, \quad \boldsymbol{\lambda}_i := \Lambda_{ii}\boldsymbol{e} \quad \forall i \in [n].$$

Furthermore, we define the set $\Lambda := \{\boldsymbol{\lambda}_i \mid i \in [n]\}$.

Lemma 7.24:
The inequality $\langle \boldsymbol{\lambda}_i, \mathbf{C}_i \boldsymbol{\lambda}_i \rangle \leq \langle \mathbf{C}_i, \boldsymbol{\Lambda}_{ii} \rangle$ *holds for all* $i \in [n]$. *Furthermore, strict inequality implies that* $(\boldsymbol{\lambda}_* \boldsymbol{\lambda}_*^\top, \boldsymbol{\lambda}_i \boldsymbol{\lambda}_i^\top)$ *is infeasible for* (7.15).

Proof. By Lemma 7.20 and the Schur complement Lemma 2.5, $\boldsymbol{\Lambda}_{ii} \succeq \boldsymbol{\lambda}_i \boldsymbol{\lambda}_i^\top$, and the inequality follows from $\mathbf{C}_i \succeq \mathbf{0}$ due to self-duality of $\mathcal{S}_+^n$. By our optimality assumption, the set $(\boldsymbol{\lambda}_* \boldsymbol{\lambda}_*^\top, \boldsymbol{\lambda}_i \boldsymbol{\lambda}_i^\top)$ cannot be feasible and have strictly smaller objective value. □

The set $(\boldsymbol{\lambda}_* \boldsymbol{\lambda}_*^\top, \boldsymbol{\lambda}_i \boldsymbol{\lambda}_i^\top)$ already satisfies several properties of (7.12). In particular, the only constraints that can be violated are

$$\langle (\boldsymbol{\lambda}_*)_{v(s)}, \boldsymbol{e} \rangle (\boldsymbol{\lambda}_*)_{v(s)} = (\boldsymbol{\lambda}_*)_{v(s)} \qquad \forall s \in [q],\ \forall i \in [n], \tag{7.17a}$$

$$(\boldsymbol{\lambda}_*)_{v(s)} (\boldsymbol{\lambda}_i)_{v(s)}^\top = (\boldsymbol{\lambda}_i)_{v(s)} (\boldsymbol{\lambda}_i)_{v(s)}^\top \qquad \forall s \in [q],\ \forall i \in [n], \tag{7.17b}$$

$$(\boldsymbol{\lambda}_i)_{v(s)} (\boldsymbol{\lambda}_i)_{v(t)}^\top = \mathbf{0} \qquad \forall s \neq t \in [q],\ \forall i \in [n], \tag{7.17c}$$

which all enforce the combinatorial structure of the problem through restricting the support of the vectors.

Due to this fact and the nonlinear nature of the problem, our goal is now to use the set Λ in order to construct a good partition $\mathcal{T}$ for the original problem (7.2) instead of deriving a feasible solution for (R2). In particular, we use the vectors $\boldsymbol{\lambda}_i$ over their matrices $\boldsymbol{\Lambda}_{ii}$, since rounding to their native space Δ^m is much more tractable than rounding to the space of doubly stochastic matrices of rank 1.

To motivate how we construct the partitions, we need to introduce another concept. Given Λ, we define the family $\{\boldsymbol{\lambda}_T \mid T \subseteq [n]\}$ by defining

$$\boldsymbol{\lambda}_T := \bigvee_{i \in T} \boldsymbol{\lambda}_i$$

as the meet of $\{\boldsymbol{\lambda}_i \mid i \in T\}$ in the lattice $(\mathbb{R}_+^m, \leq)$. In other words, $\boldsymbol{\lambda}_T$ is maximal with regards to $\leq$ such that $\boldsymbol{\lambda}_T \leq \boldsymbol{\lambda}_i$ for all $i \in T$, and as a byproduct, we also have $\boldsymbol{\lambda}_T \geq \mathbf{0}$. The idea behind this concept is the following observation.

Lemma 7.25:
Let $\mathcal{T} \in \mathcal{P}_k^n$, $\boldsymbol{\lambda}_{T_j} \neq \mathbf{0}$ *for all* $j \in [k]$ *and* $\mathbf{C}_i \geq \mathbf{0}$ *for all* $i \in [n]$. *Then*

$$\sum_{j \in [k]} \sum_{i \in T_j} \langle \boldsymbol{\lambda}_{T_j}, \mathbf{C}_i \boldsymbol{\lambda}_{T_j} \rangle \leq \sum_{i \in [n]} \langle \mathbf{C}_i, \boldsymbol{\Lambda}_{ii} \rangle \leq \max \left\{ \langle \boldsymbol{\lambda}_{T_j}, \boldsymbol{e} \rangle^{-2} \,\middle|\, j \in [k] \right\} \cdot \sum_{j \in [k]} \sum_{i \in T_j} \langle \boldsymbol{\lambda}_{T_j}, \mathbf{C}_i \boldsymbol{\lambda}_{T_j} \rangle.$$

Proof. The first inequality follows from the assumption $\mathbf{C}_i \in \mathcal{S}_+^m \cap \mathbb{R}_+^{m \times m}$, since

$$(\boldsymbol{\Lambda}_{ii} - \boldsymbol{\lambda}_i \boldsymbol{\lambda}_i^\top) + (\boldsymbol{\lambda}_i \boldsymbol{\lambda}_i^\top - \boldsymbol{\lambda}_{T_j} \boldsymbol{\lambda}_{T_j}^\top) \in \mathcal{S}_+^m + \mathbb{R}_+^{m \times m} = (\mathcal{S}_+^m \cap \mathbb{R}_+^{m \times m})^*.$$

For the second inequality, first note that $\frac{\boldsymbol{\lambda}_T}{\langle \boldsymbol{\lambda}_T, \boldsymbol{e}\rangle} \in \Delta^m$ holds for all $T \subseteq [n]$. We now argue that the set $\left\{ \frac{\boldsymbol{\lambda}_{T_j}}{\langle \boldsymbol{\lambda}_{T_j}, \boldsymbol{e}\rangle} \;\middle|\; j \in [k] \right\}$ together with $\mathbf{U}(\mathcal{T})$ can be considered as a feasible point for (R1), showing

$$\sum_{i\in[n]} \langle \mathbf{C}_i, \boldsymbol{\Lambda}_{ii} \rangle \leq \sum_{j\in[k]} \sum_{i\in T_j} \langle \boldsymbol{\lambda}_{T_j}, \boldsymbol{e}\rangle^{-2} \cdot \langle \boldsymbol{\lambda}_{T_j}, \mathbf{C}_i \boldsymbol{\lambda}_{T_j} \rangle$$

since (7.15) is a relaxation of (R1).

To see this, first note that Definition 7.5 states that the constraint $\boldsymbol{\lambda}^j \in \Delta_\Omega^m$ in (R1) is redundant. While this normally means the constraint can be ignored completely without changing the optimal value of (R1), we still need to make sure that the constraint $\langle \boldsymbol{\lambda}^j, \boldsymbol{e}\rangle = 1$ remains in place, since it was used in the construction of the objective matrices $\mathbf{C}_i$ for homogenization. Since this is the case for $\frac{\boldsymbol{\lambda}_{T_j}}{\langle \boldsymbol{\lambda}_{T_j}, \boldsymbol{e}\rangle}$, we can set $\boldsymbol{\lambda}^j = \frac{\boldsymbol{\lambda}_{T_j}}{\langle \boldsymbol{\lambda}_{T_j}, \boldsymbol{e}\rangle}$ and use the assignment matrix $\mathbf{U}(\mathcal{T})$ as feasible point of a slight variant of (R1) with the same optimal value. Finally, the second inequality follows by bounding the terms $\langle \boldsymbol{\lambda}_{T_j}, \boldsymbol{e}\rangle^{-2}$ from above. □

The preceding lemma shows that in the case of nonnegative objective $\mathbf{C}_i$, we can actually give an upperbound for the approximation *quality* in the form of

$$\rho(\mathcal{T}) := \max \left\{ \langle \boldsymbol{\lambda}_{T_j}, \boldsymbol{e}\rangle^{-2} \;\middle|\; j \in [k] \right\}.$$

Maybe unsurprisingly, there is an inverse relation between the governing parameter $\langle \boldsymbol{\lambda}_T, \boldsymbol{e}\rangle = \|\boldsymbol{\lambda}_T\|_1$ and the diameter of the underlying set. In particular, we define the diameter $\operatorname{diam}(C)$ of a compact set $C \subseteq \mathbb{R}^d$ as

$$\operatorname{diam}(C) := \max \{ \|\boldsymbol{x} - \boldsymbol{y}\|_2 \;\mid \boldsymbol{x}, \boldsymbol{y} \in C \}.$$

However, in order to show this, we first need some additional geometric insights. For this, we first recall that the d-dimensional unit ball with radius r and center $\boldsymbol{\mu}$ is given as

$$B_r^d(\boldsymbol{\mu}) := \left\{ \boldsymbol{x} \in \mathbb{R}^d \;\middle|\; \|\boldsymbol{x} - \boldsymbol{\mu}\|_2 \leq r \right\}.$$

We can now formulate the following result.

Theorem 7.26 (Jung's Theorem, [DGK63, Thm. 2.6])**:**
Let $C \subseteq \mathbb{R}^d$ be compact. Then there is $\boldsymbol{\mu} \in \mathbb{R}^d$ and

$$\tfrac{1}{2} \operatorname{diam}(C) \leq r \leq \sqrt{\tfrac{d}{2(d+1)}} \operatorname{diam}(C)$$

such that $C \subseteq B_r^d(\boldsymbol{\mu})$.

In what follows, let $H = \left\{ \boldsymbol{x} \in \mathbb{R}^d \;\middle|\; \langle \boldsymbol{x}, \boldsymbol{e} \rangle = 1 \right\}$ be the unique hyperplane satisfying the inclusion $\Delta^d \subseteq H \subseteq \mathbb{R}^d$. Furthermore, we parametrize the $(d-1)$-dimensional balls in $H \cong \mathbb{R}^{d-1}$ through their centers by setting

$$B_r^H(\boldsymbol{\mu}) := B_r^d(\boldsymbol{\mu}) \cap H \subseteq H.$$

for all $\boldsymbol{\mu} \in H$.

Lemma 7.27:
The inclusion $B_r^H(\boldsymbol{\mu}) \subseteq \Delta^d$ implies $r \leq \big(d(d-1)\big)^{-\frac{1}{2}}$.

Proof. Let γ be maximal such that $B_\gamma^H(\boldsymbol{\mu}) \subseteq \Delta^d$. Then by symmetry, we can assume that the center is given by $\boldsymbol{\mu} = \frac{1}{d}\boldsymbol{e} \in \mathbb{R}^d$, and the d facet-defining inequalities of Δ^d are given by $x_i \geq 0$. It follows then from symmetry that

$$\begin{aligned} \gamma^2 &= \min \left\{ \|\boldsymbol{\mu} - \boldsymbol{x}\|_2^2 \;\middle|\; \boldsymbol{x} \in \Delta^d,\ x_i = 0 \right\} \\ &= \tfrac{1}{d^2} + \min \left\{ \|\boldsymbol{\mu}' - \boldsymbol{x}\|_2^2 \;\middle|\; \boldsymbol{x} \in \Delta^{d-1} \right\} \\ &= \tfrac{1}{d^2} + \operatorname{dist}\left(\boldsymbol{\mu}', \Delta^{d-1}\right)^2, \end{aligned}$$

where $\boldsymbol{\mu}' = \frac{1}{d}\boldsymbol{e} \in \mathbb{R}^{d-1}$. Since $\boldsymbol{e}$ is orthogonal to H, the orthogonal projection of $\boldsymbol{\mu}'$ onto Δ^{d-1} is $\pi_{\Delta^{d-1}}(\boldsymbol{\mu}') = \frac{1}{d-1}\boldsymbol{e}$, and the latter minimum is

$$\left\| \left(\frac{1}{d} - \frac{1}{d-1} \right) \boldsymbol{e} \right\|_2^2 = \frac{1}{(d-1)^2 d^2} \cdot (d-1) = \frac{1}{(d-1)d^2}.$$

As a consequence, $\gamma = \sqrt{\frac{1}{d^2} + \frac{1}{(d-1)d^2}} = \big(d(d-1)\big)^{-\frac{1}{2}}$. □

Theorem 7.28 (Lower Bound Construction)**:**
Let $\Lambda \subseteq \Delta^d \subseteq \mathbb{R}^d_+$ and denote by $\boldsymbol{\lambda}_\vee$ the meet of Λ in $(\mathbb{R}^d_+, \leq)$. Then there is a lower bound $\mathbf{0} \leq \boldsymbol{\mu}_+ \leq \boldsymbol{\lambda}_\vee$ such that

$$1 - \tfrac{d-1}{\sqrt{2}} \cdot \operatorname{diam}(\Lambda) \leq \|\boldsymbol{\mu}_+\|_1 \leq \|\boldsymbol{\lambda}_\vee\|_1.$$

Proof. Applying Theorem 7.26 on $\Lambda \subseteq H \cong \mathbb{R}^{d-1}$ yields a translation $\boldsymbol{\mu}' \in H$ such that

$$\Lambda \subseteq B_r^H(\boldsymbol{\mu}') \subseteq H$$

where

$$r \leq \sqrt{\tfrac{d-1}{2d}} \cdot \operatorname{diam}(\Lambda).$$

Using translation and scaling operations, we have the equivalence

$$\begin{aligned} B_\gamma^H(\tfrac{1}{d}\boldsymbol{e}) \subseteq \Delta^d \;&\Leftrightarrow\; (\tfrac{1}{d}\boldsymbol{e} - \boldsymbol{\mu}') + B_\gamma^H(\boldsymbol{\mu}') \subseteq \Delta^d \;&&\Leftrightarrow\; B_\gamma^H(\boldsymbol{\mu}') \subseteq (\boldsymbol{\mu}' - \tfrac{1}{d}\boldsymbol{e}) + \Delta^d \\ &\Leftrightarrow\; \tfrac{\gamma}{r} \cdot B_r^H(\boldsymbol{\mu}') \subseteq (\boldsymbol{\mu}' - \tfrac{1}{d}\boldsymbol{e}) + \Delta^d \;&&\Leftrightarrow\; B_r^H(\boldsymbol{\mu}') \subseteq \tfrac{r}{\gamma}(\boldsymbol{\mu}' - \tfrac{1}{d}\boldsymbol{e}) + \tfrac{r}{\gamma} \cdot \Delta^d, \end{aligned}$$

and it follows by Lemma 7.27 that all statements are true for $\gamma = \big(d(d-1)\big)^{-\frac{1}{2}}$. Denoting $\boldsymbol{\mu} := \frac{r}{\gamma}(\boldsymbol{\mu}' - \frac{1}{d}\boldsymbol{e})$ and $\tau := \frac{r}{\gamma}$, we thus have the inclusion

$$B_r^H(\boldsymbol{\mu}') \subseteq \boldsymbol{\mu} + \tau \cdot \Delta^d =: D \subseteq H. \tag{7.18}$$

Now let $\boldsymbol{\mu} = \boldsymbol{\mu}_+ + \boldsymbol{\mu}_-$ be the Moreau decomposition of $\boldsymbol{\mu}$ into non-negative and non-positive vectors. Since $\Lambda \subseteq D_+ := D \cap \mathbb{R}^d_+$, we have $\mathbf{0} \leq \boldsymbol{\mu}_+ \leq \boldsymbol{\lambda}$ for all $\boldsymbol{\lambda} \in \Lambda$ and in particular $\mathbf{0} \leq \boldsymbol{\mu}_+ \leq \boldsymbol{\lambda}_\vee$. As a direct consequence,

$$\|\boldsymbol{\lambda}_\vee\|_1 = \langle \boldsymbol{\lambda}_\vee, \boldsymbol{e}\rangle \geq \langle \boldsymbol{\mu}_+, \boldsymbol{e}\rangle \geq \langle \boldsymbol{\mu}, \boldsymbol{e}\rangle = 1 - \tau,$$

where the second equality follows from (7.18). Finally, we have

$$\tau = \frac{r}{\gamma} \leq \frac{d-1}{\sqrt{2}} \cdot \operatorname{diam}(\Lambda).$$

□

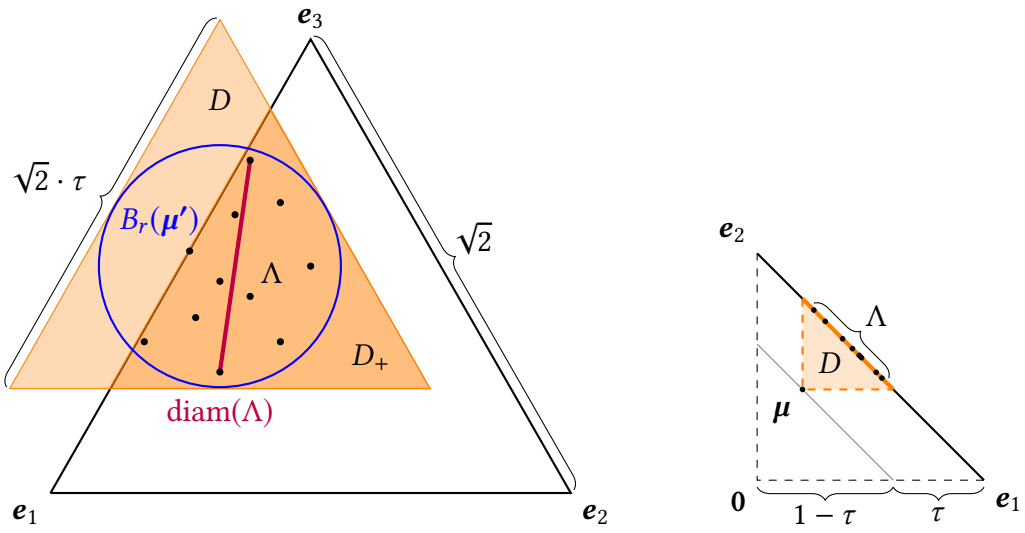

Figure 7.2: Illustration of Theorem 7.28. Left: H for $d = 3$, $\boldsymbol{\mu} \neq \boldsymbol{\mu}_+$. Right: $d = 2$, $\boldsymbol{\mu} = \boldsymbol{\mu}_+$.

Remark 7.29:
It should be noted that the factor $\frac{d-1}{\sqrt{2}}$ in the result of Theorem 7.28 is overly pessimistic and depends heavily on the shape of $\operatorname{conv}(\Lambda)$. In particular, the worst case is achieved when Λ contains the midpoint of every facet of Δ^d, yielding the translation of a rescaled version of $-\Delta^d$, which can be considered as the dual polytope of Δ^d.

However, the constraints (7.17a) actively push the solution away from this configuration, which still has an effect on the relaxations as is implicit in the relations between $\boldsymbol{\lambda}_*$ and $\boldsymbol{\lambda}_i$. Since this influence is hard to quantify, we will just stress that the preceding result describes the worst-case, and actual results may be much better in practice.

Now that we have an interest in using the diameter in place of the quality of the solution, our goal is to find partitions of Λ that minimize the largest diameter of their parts. We will do this indirectly by minimizing a close upperbound on the diameter, which can be derived from the triangle inequality as follows.

Let $X = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_n\} \subseteq \mathbb{R}^d$. Then for any $l \in [n]$, we have

$$\begin{aligned}\operatorname{diam}(X) &= \max\left\{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2 \;\middle|\; i,j \in [n]\right\}\\ &\leq \max\left\{\|\boldsymbol{x}_i - \boldsymbol{x}_l\|_2 + \|\boldsymbol{x}_l - \boldsymbol{x}_j\|_2 \;\middle|\; i,j \in [n]\right\}\\ &\leq 2 \cdot \max\{\|\boldsymbol{x}_i - \boldsymbol{x}_l\|_2 \mid i \in [n]\} \leq 2 \cdot \operatorname{diam}(X).\end{aligned}$$

In particular, $2 \cdot \max\{\|\boldsymbol{x}_i - \boldsymbol{x}_l\|_2 \mid i \in [n]\}$ is a 2-approximation for *every* $l \in [n]$, and choosing l to minimize the term will usually give an even better approximation.

Optimizing this 2-approximation of the diameter immediately brings us to the *k-center clustering problem.*

Definition 7.30 (k-center Clustering)**:**
Given a set $X = \{\boldsymbol{x}_i \mid i \in [n]\} \subseteq \mathbb{R}^d$, the k-center clustering problem is defined as

$$C_\infty(X,k) := \min\left\{ \max_{i\in[n]} \operatorname{dist}(\boldsymbol{x}_i, Y) \;\middle|\; Y \subseteq X, |Y| = k \right\}. \tag{7.19}$$

Fortunately, this is a well-studied problem with efficient and *deterministic* heuristics despite it being NP-hard.

Algorithm 7.1: Farthest Point Clustering (FPN)

Data: Data $X = \{\boldsymbol{x}_i \mid i \in [n]\} \subseteq \mathbb{R}^d$, $k \in [n]$
Result: Centers $Y \subseteq X, |Y| = k$

```
b ← ∞, Y ← ∅;
for i ∈ [n] do
    Y_i ← ∅, y_1 ← x_i;
    for j ∈ [k] do
        Y_i ← Y_i ∪ {y_j};
        y_{j+1} ← argmax {dist(x, Y_i) | x ∈ X};
    if dist(y_{k+1}, Y_i) < b then
        b ← dist(y_{k+1}, Y_i);
        Y ← Y_i;
return Y;
```

Algorithm 7.1 greedily builds the set of cluster centers Y by iteratively choosing those points which are farthest away from all prior centers. As initialization, every point is chosen as the first cluster center once and the best overall result is kept as the output of the algorithm. Given its output $Y = \{\boldsymbol{y}_1, \dots, \boldsymbol{y}_k\}$, we can construct a partition $\mathcal{T}(Y) = \{T_1, \dots, T_k\} \in \mathcal{P}_k^n$ by setting

$$T_j = \left\{ i \in [n] \;\middle|\; \operatorname{dist}(\boldsymbol{x}_i, Y) = \|\boldsymbol{x}_i - \boldsymbol{y}_j\|_2 \right\}.$$

As explained by the following theorem, Algorithm 7.1 is optimal for this problem.

Theorem 7.31 (Approximating k-Center clustering [HS85])**:**
For $d > 2$, achieving an approximation ratio for (7.19) better than 2 is NP-hard, and a 2-approximation is given by Algorithm 7.1.

Our rounding procedure can now be summarized as follows.

Algorithm 7.2: Deterministic Rounding

Data: Solution $(\boldsymbol{\Lambda}_{**}, \boldsymbol{\Lambda}_{ii})$ of (7.15).
Result: Partition $\mathcal{T}$ for (7.2).
1 set $\Lambda = \{\boldsymbol{\Lambda}_i \boldsymbol{e} \mid i \in [n]\}$;
2 compute $Y = \mathrm{FPN}(\Lambda, k)$;
3 **return** $\mathcal{T}(Y)$;

7.6 Modifications

When we introduced simplicial covers in Section 7.2, we focused on a basic setting for the sake of exposition. In this section, we return to some underlying assumptions and see how they might be generalized in order to extend the framework suggested in this chapter. In particular, we will focus on the structure of simplicial covers, including the choices of $\mathbf{V}$ and $\boldsymbol{\Omega}$, as well as the objective function.

For most of these generalizations, the constructions in Sections 7.2–7.3 can be carried through analogously, and we will only comment on the results that break down.

Removing redundant vertices

In Section 7.2, there was no assumption on the uniqueness of the columns in $\mathbf{V}$, and, as pointed out by Remark 7.9, if simplices shared vertices, we used one copy of each vertex for each simplex. The underlying idea was to ensure the blockdiagonal structure of $\boldsymbol{\Omega}$, which in turn is the foundation of Lemma 7.13 and all of its consequences throughout Section 7.2.

However, we can arrive at a slight variation of (R2) even without this assumption. To see this, we merely need to change the definition of the index sets $v(s)$ and adjust $\boldsymbol{\Omega}$

and $\boldsymbol{\lambda}_*$ accordingly. Given a matrix $\mathbf{V} = (\mathbf{V}_1, \dots, \mathbf{V}_q)$ with multiple identical columns, let $\mathbf{V}'$ collect all the unique columns of $\mathbf{V}$. Then let $v(s)$ denote the unique set of indices in $\mathbf{V}'$ corresponding to the columns of $\mathbf{V}_s$ and let Ω be the binary matrix which is one everywhere, except for the principal submatrices indexed by each $v(s)$.

Remark 7.32:
If $\mathcal{P}$ defines a simplicial complex with corresponding independence system L as in Remark 7.6, then the new matrix Ω is equal to the matrix Ω_L arising from L as defined in Section 2.2. This is analogous to the connection described in Remark 7.10 and can be seen as generalization.

As a consequence of these new index sets, we might have $\langle (\boldsymbol{\lambda}_*)_{v(s)}, \boldsymbol{e} \rangle > 1$ for some $s \in [q]$, as can be observed in Example 7.33. To resolve this, we have to weaken (R2) by only assuming inequality instead of equality in

$$\langle (\boldsymbol{\lambda}_*)_{v(s)}, \boldsymbol{e} \rangle (\boldsymbol{\lambda}_*)_{v(s)} \geq (\boldsymbol{\lambda}_*)_{v(s)}.$$

Then, by construction,

$$\min \left\{ \sum_{i \in [n]} \langle \boldsymbol{\lambda}_i, \mathrm{C}_i \boldsymbol{\lambda}_i \rangle \;\middle|\; \begin{array}{rl} \boldsymbol{\lambda}_* \in k \cdot \Delta^m, & \\ \langle (\boldsymbol{\lambda}_*)_{v(s)}, \boldsymbol{e} \rangle (\boldsymbol{\lambda}_*)_{v(s)} \geq (\boldsymbol{\lambda}_*)_{v(s)} & \forall s \in [q], \\ \boldsymbol{\lambda}_i \in \Delta^m_\Omega, & \forall i \in [n], \\ \boldsymbol{\lambda}_i \leq \boldsymbol{\lambda}_* & \forall i \in [n] \end{array} \right\} \tag{7.20}$$

is still a relaxation of the original problem. However, an analogue to Theorem 7.16 fails to hold in this situation, as showcased by the following example.

Example 7.33:
Consider the 1-dimensional Euclidean 2-clustering problem from Subsection 6.3.2 for the set of points $\{b_1, b_2, b_3, b_4\} = \{-0.6, -0.4, 0.4, 0.6\}$. Enumerating all partitions, it can be verified that the optimal solution is achieved with $x_1 = -0.5$ and $x_2 = 0.5$ for the partition $\{\{1, 2\}, \{3, 4\}\}$ and has value 0.04.

Let the simplicial cover $\mathcal{P} = P_1 \cup P_2 \subseteq \mathbb{R}$ be given by $P_1 = [-1, 0]$ and $P_2 = [0, 1]$. Then $\mathcal{P}$ is separating, and by Theorem 7.16, the optimal solution can be computed by using $\mathbf{V}_1 = \begin{pmatrix} -1 & 0 \end{pmatrix}$, $\mathbf{V}_2 = \begin{pmatrix} 0 & 1 \end{pmatrix}$ and $\mathbf{V} = \begin{pmatrix} -1 & 0 & 0 & 1 \end{pmatrix}$ to construct and solve (R2).

Now consider the reduced matrix $V' = \begin{pmatrix} -1 & 0 & 1 \end{pmatrix}$ consisting of unique columns. By the previous discussion, the indices are given as $v(1) = \{1, 2\}$ and $v(2) = \{2, 3\}$, and thus

$$\Omega = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}.$$

Then the vectors

$$(\boldsymbol{\lambda}_*, \boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \boldsymbol{\lambda}_3, \boldsymbol{\lambda}_4) = \begin{pmatrix} 0.6 & 0.6 & 0.4 & 0 & 0 \\ 0.8 & 0.4 & 0.6 & 0.6 & 0.4 \\ 0.6 & 0 & 0 & 0.4 & 0.6 \end{pmatrix}$$

are feasible for (7.20) and each $\boldsymbol{\lambda}_i$ represents its data-point b_i. Then this solution is optimal for (7.20) and has objective value 0, which is strictly lower than the real minimum of 0.04.

The example shows that while this approach reduces m and thus the size of the matrices involved in the MM approach of Section 7.3, the quality of the solutions may drop to a point where they do not yield any actual information about the optimum anymore. Additionally, we lose the computational advantages of the diagonal block structure of $\boldsymbol{\Omega}$, resulting in a weakly coupled system.

Furthermore, the example illustrates the mechanism responsible for the reduced quality of the solution as a "discount" in terms of convex combinations. In particular, raising the upperbound given by $\boldsymbol{\lambda}_*$ on those vertices occuring in multiple polytopes extends the representable points in all of these polytopes, in contrast to the vertices occuring only once. Consequently, the individual $\boldsymbol{\lambda}_i$ will have a larger spread, as can be seen by comparing Figures 7.6 and 7.7.

However, the same approach can succesfully be used to replace the simplices in our simplicial cover by general polytopes, as can be seen in the next subsection.

Polytope covers

In Section 7.2, we introduced $\mathcal{P}$ as a union of separate simplices whose dimension is fixed to that of the ambient space. We can easily generalize this concepts by allowing any combination of

- a union of general polytopes instead of simplices,
- different dimensions for each of the polytopes.

Both generalizations have in common that in principal, they only impact the number of vertices for each polytope in $\mathcal{P}$. Consequentially, only the definition of the matrix $\mathbf{V}$ in (7.4) needs to be slightly adjusted, and we can arrive at both (R2) and (7.15) without impacting any of the theoretical results.

For practical computations however, there is a slight difference between allowing lower dimensional simplices and using general polytopes instead of simplices. While the former does not impact the relaxations constructed by MM, the latter suffers from the lack of unique representation, as is implicit in the lack of uniqueness in Theorem 2.8 and showcased in the following example.

Example 7.34:
Consider the unit square $C^2 = [0,1]^2$ and a point $\boldsymbol{x} \in C^2$. Then $\mathbf{V} = \left(\begin{smallmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{smallmatrix}\right)$ describes the vertices of C^2 and any vector $\boldsymbol{\lambda} \in \Delta^4$ that satisfies $\mathbf{V}\boldsymbol{\lambda} = \boldsymbol{x}$ represents $\boldsymbol{x}$. However,

if $\boldsymbol{x} \in \operatorname{int}(C^2)$, this representation is not unique, since

$$C^2 = \operatorname{conv}\left(\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}\right) \cup \operatorname{conv}\left(\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}\right) = \operatorname{conv}\left(\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}\right) \cup \operatorname{conv}\left(\begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}\right)$$

can be decomposed in two different ways, which is visualized by Figure 7.3.

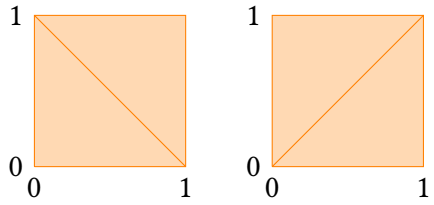

Figure 7.3: Both decompositions of C^2 as union of 2 triangles.

By Caratheodory's theorem, both decompositions define different representations and every convex combination of these representations results in yet another representation.

Most of the time, this ambiguity will weaken the relaxation (7.15) considerably, since there are more degrees of freedom available. Luckily, we can use the ideas of the preceding section to remove the ambiguity by adjusting $\boldsymbol{\Omega}$. To see this, let $\mathcal{P}$ be the union of polytopes P_s for $s \in [q]$ and $\mathbf{V}_s$ their corresponding vertices. Then we can subdivide each polytope P_s into a simplicial complex to rewrite

$$P_s = \bigcup_{i \in q_s} P_{s,i}$$

where each $P_{s,i}$ is a simplex. Then each $P_{s,i}$ defines a unique index-set $v(s,i)$ of the columns of $\mathbf{V}_s$, and we can define $\boldsymbol{\Omega}$ as the binary matrix which is one everywhere, except for the principal submatrices indexed by $v(s,i)$ for both $s \in [q]$ and $i \in [q_s]$. This approach guarantees a unique representation for almost every point in the interior of the polytope P_s. Computationally, this results in a single psd. block of size $|\mathbf{V}_s|$, compared to the unique representation achieved by using the q_s blocks of size $d+1$ for each simplex $P_{s,i}$.

Example 7.35:
Consider $\mathcal{P} = \bigcup_{s \in [4]} P_{z_s} \subseteq \mathbb{R}^2$ where $P_{z_s} = \boldsymbol{z}_s + C_2$ for $C_2 = [0,1]^2$ and

$$\begin{pmatrix} z_1 & z_2 & z_3 & z_4 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}.$$

Then

$$\begin{aligned}\mathbf{V} &= \left(\begin{array}{c|c|c|c} \mathbf{V}_1 & \mathbf{V}_2 & \mathbf{V}_3 & \mathbf{V}_4 \end{array}\right) \\ &= \left(\begin{array}{cccc|cccc|cccc|cccc} 0 & 1 & 0 & 1 & 1 & 2 & 1 & 2 & 0 & 1 & 0 & 1 & 1 & 2 & 1 & 2 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 2 & 2 & 1 & 1 & 2 & 2 \end{array}\right) \in \mathbb{R}^{2\times 16}\end{aligned}$$

and since each polytope is a translation of C_2, we can choose the same simplicial decomposition for each polytope. We thus divide each square into a lower-left and upper-right triangle, as depicted by Figure 7.4. Consequentially, we set

$$\mathbf{\Omega} = \mathbf{J}_{16} - \mathbf{I}_4 \otimes \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}$$

to get a unique representation involving 4 psd. blocks of size 4 each. This is a computational trade-off, since using the simplicial decomposition directly would have resulted in 8 sdp blocks of size 3.

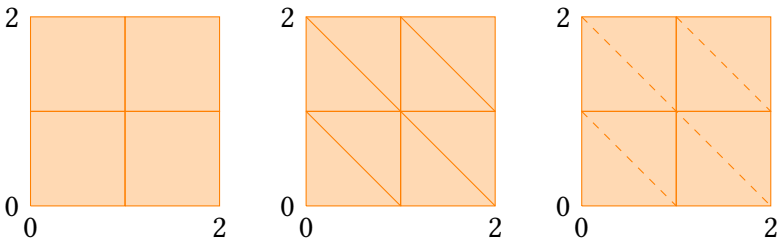

Figure 7.4: Left: Original polytope cover consisting of 4 squares. Middle: Simplicial decomposition using 8 triangles. Right: Polytope cover with implied constraints from simplicial decomposition.

Describing the σ-skeleton

As a special case of the previous subsection, $\mathcal{P}$ might be given as the σ-skeleton of a polytope, given by the following definition.

Definition 7.36:
Let $P \subseteq \mathbb{R}^d$ be a polytope with vertex set $\mathbf{V}$ and $\sigma \in [d]$. Define $G_\sigma(P)$ as the graph with vertex set $\mathbf{V}$ where the edge (v_1, v_2) is contained if and only if the line segment $\operatorname{conv}(\{v_1, v_2\})$ is contained in a face of P with dimension at most σ. Then $\mathbf{A}_{\overline{G_\sigma(P)}}$ denotes the adjacency matrix of the complementary graph and the σ-skeleton $\operatorname{skel}_\sigma(P)$

of P is formally defined as

$$\operatorname{skel}_\sigma(P) := \left\{ \boldsymbol{x} \in \mathbb{R}^d \;\middle|\; \boldsymbol{x} = \mathbf{V}\boldsymbol{\lambda},\ \ \boldsymbol{\lambda} \in \Delta^m_{\mathbf{A}_{\overline{G_\sigma(P)}}},\ \ \|\boldsymbol{\lambda}\|_0 \leq \sigma + 1 \right\}, \tag{7.21}$$

which is the union of all faces of P of dimension at most σ.

Example 7.37:
The unit square C^2 is given as the convex hull of $\mathbf{V} = \left(\begin{smallmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{smallmatrix}\right)$ and $\operatorname{skel}_1(C^2)$ consists of 4 line segments, as shown in Figure 7.5. Choosing $\mathcal{P} = \operatorname{skel}_1(C^2)$ we need 4 2-simplices and consequently $m = 8$ vertices for using the polytope cover approach from Section 7.6. Using the sparsity constraint $\|\boldsymbol{\lambda}\|_0 \leq 2$ in (7.21) allows us to use each vertex only once to end up with $m = 4$ instead.

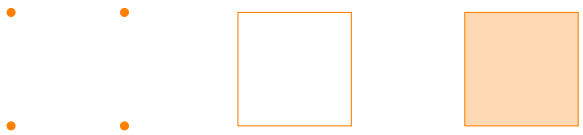

Figure 7.5: The σ-skeletons of C_2. As σ ranges from 1 to 3, the σ-skeleton describes the union of vertices, edges and the square itself.

Remark 7.38:
The sparse set $\{\boldsymbol{\lambda} \in \Delta^m \mid \|\boldsymbol{\lambda}\|_0 \leq \sigma\}$ can be described by adding the equations

$$\boldsymbol{\lambda}^{\boldsymbol{e}_S} = 0 \quad \forall\, S \subseteq [m] \colon |S| \geq \sigma + 1.$$

Using MM, these equations can be incorporated by expanding $\boldsymbol{\Omega}$ to sum over higher moments as well. However, this approach becomes impractical very quickly since it requires a stage $t > \sigma$ to work.

It would be interesting to investigate low-degree polynomials as approximations of sparsity constraints.

Simplicial Covers for Semialgebraic Sets

So far, our focus in this section was to modify the construction of the underlying simplicial cover, where we have always assumed that $\mathrm{SE}^{\mathcal{A}}_{\mathbf{B}}$ optimizes a variable $\boldsymbol{x} \in \mathbb{R}^d$ as in (7.1). However, the powerful results concerning semi-algebraic optimization in Sections 2.6 and 3.1 can be used with our framework to restrict $\boldsymbol{x}$ to a semialgebraic subset $K \subseteq \mathbb{R}^d$ as well. In particular, let

$$K = \left\{ \boldsymbol{x} \in \mathbb{R}^d \;\middle|\; g_i(\boldsymbol{x}) \geq 0,\ \ i \in \mathcal{I} \right\}$$

for multivariate polynomials $g_i \in \mathbb{R}[\boldsymbol{x}]$. Then each $g_i(\boldsymbol{x})$ can be easily turned into a polynomial expression $g'(\boldsymbol{\lambda}) = g(\mathbf{V}\boldsymbol{\lambda})$, and we can additionally assume that g' is a homogeneous polynomial of the same degree as g, such that $\boldsymbol{x} \in K$ where $\boldsymbol{x} = \mathbf{V}\boldsymbol{\lambda}$ if and only if

$$\boldsymbol{\lambda} \in K' := \left\{ \boldsymbol{\lambda} \in \mathbb{R}^m \;\middle|\; g_i'(\boldsymbol{\lambda}) \geq 0, \;\; i \in \mathcal{I} \right\}.$$

Consequentially, it suffices to add the membership $\boldsymbol{\lambda}_i \in K'$ for all $i \in [n]$ to problem (R2). Using MM, we then arrive at (7.14) with additional constraints. In general, we only need to cover K by a set of simplices to use our approach. In particular, there is no need for a simplicial cover to *approximate* K with those simplices, as long as K is *covered* by them.

Example 7.39:
Let K be the unit sphere, then $K = \left\{ \boldsymbol{x} \in \mathbb{R}^d \;\middle|\; \langle \boldsymbol{x}, \boldsymbol{x} \rangle = 1 \right\}$ is described by the homogeneous quadratic equation $\langle \boldsymbol{x}, \boldsymbol{x} \rangle = 1$. Substituting $\boldsymbol{x} = \mathbf{V}\boldsymbol{\lambda}$ yields again a quadric defined by $\boldsymbol{\lambda}^\top (\mathbf{V}^\top \mathbf{V}) \boldsymbol{\lambda} = 1$. Adding this constraint to (7.14) then turns into

$$\langle \mathbf{\Lambda}_{ii}, \mathbf{V}^\top \mathbf{V} - \mathbf{J}_m \rangle = 0 \;\; \forall i \in [n],$$

where we used homogenization.

Example 7.40:
Let K be the set of vectorized orthogonal 2×2 matrices $\mathbf{X}$ in $\mathbb{R}^4$ given by the map

$$\mathbf{X} = \begin{pmatrix} x_1 & x_{12} \\ x_{21} & x_2 \end{pmatrix} \mapsto \begin{pmatrix} x_1 & x_{12} & x_{21} & x_2 \end{pmatrix}^\top = \boldsymbol{x}.$$

Then K can be described as the zeros of four quadratic equations, one for each entry of $\mathbf{X}\mathbf{X}^\top = \mathbf{I}_2$. Denoting them by $\boldsymbol{x}^\top \mathbf{Q}_l \boldsymbol{x} = q_l$, substituting $\boldsymbol{x} = \mathbf{V}\boldsymbol{\lambda}$ yields again an intersection of quadrics defined by $\boldsymbol{\lambda}^\top (\mathbf{V}^\top \mathbf{Q}_l \mathbf{V}) \boldsymbol{\lambda} = q_l$, which turn into the homogeneous equations

$$\langle \mathbf{\Lambda}_{ii}, \mathbf{V}^\top \mathbf{Q}_l \mathbf{V} - q_l \mathbf{J}_m \rangle = 0 \;\; \forall i \in [n]$$

for (7.14).

However, we note that finding a separating simplicial cover may prove much harder when K has a complex geometry. In particular, while MM will converge towards feasibility in K, we necessarily need to start at a stage determined by the maximum degree of a polynomial describing K to avoid crude approximations.

Clustering Varieties

The task of (7.2) is to partition affine subspaces in such a way that the minimizer of $\mathrm{SE}_{\mathbf{B}}^{\mathcal{A}}$ is as close as possible to all of them. Ideally, this would mean that we get a common intersection of the corresponding affine subspaces, and so we can consider (7.2)

as a generalized common intersection point. Consequentially, we can generalize this concept from subspaces to arbitrary sets and try to adjust our approach accordingly.

An extension to the case of varieties can be done in the following way. We replace $\mathbf{A}_i\boldsymbol{x} - \boldsymbol{b}_i$ in (7.1) with $F_i(\boldsymbol{x})$, where $F_i \in \mathbb{R}[\boldsymbol{x}]$ is a multivariate polynomial describing the variety $\mathcal{V}_{\mathbb{R}}(\|F_i\|_2^2)$. Following Section 7.2, we may replace $\boldsymbol{x}$ by $\mathbf{V}\boldsymbol{\lambda}$ and homogenize $\|F_i(\mathbf{V}\boldsymbol{\lambda})\|_2^2$ using $\langle \boldsymbol{\lambda}_j, \boldsymbol{e}\rangle = 1$ to end up with a variant of (R2), where the objective function has been replaced. The results from Section 7.3 follow according to this replacement, but the stage of MM needs to be twice the maximum degree of any F_i.

Regularization

We want to point out that the reformulation (R2) never explicitly uses the number k except in the equation $\langle \boldsymbol{\lambda}_*, \boldsymbol{e}\rangle = k$ implicit in $\boldsymbol{\lambda}_* \in k \cdot \Delta^m$. We can thus treat k throughout as a variable instead and include a function of k in the objective function to dynamically search for the number of sets contained in a desired partition. The difficulty then lies in choosing this function in a way to retain a useful solution to the original problem.

7.7 Applications

We will state various applications and visualize the resulting relaxations.

Euclidean Clustering

By choosing $\mathbf{A}_i = \mathbf{I}_n$ in (7.2), we recover the classical problem (6.7) of Euclidean clustering for the points $\{\boldsymbol{b}_1, \dots, \boldsymbol{b}_n\} \subseteq \mathbb{R}^d$. As pointed out in Corollary 6.28, it is well known that $\operatorname{conv}(\{\boldsymbol{b}_1, \dots, \boldsymbol{b}_n\}) \subseteq \mathcal{P}$ will contain all optimal solutions. In particular, for the simplicial cover assumption it suffices that $\mathcal{P}$ covers a box which includes all $\{\boldsymbol{b}_1, \dots, \boldsymbol{b}_n\}$, which can be easily extracted.

We can use Euclidean clustering to get a better intuition of how the relaxation works. Regarding the choice of $\mathcal{P}$, consider Figure 7.6. Using any simplex containing all the points is the coarsest approximation but yields useless results, since each local estimate $V\boldsymbol{\lambda}_i$ can be chosen as $\boldsymbol{b}_i$.

In view of Theorem 7.16, the algorithm will perform best if the simplicial cover is separating, which means that the cluster centers are separated by the polytopes in $\mathcal{P}$. This suggests that there should be at least k simplices, and that an oversegmentation removes the need of knowing the location of the centers in advance, as can be observed in Figure 7.6 as well. Some of the variants from Section 7.6 that change how $\mathbf{V}$ and Ω are constructed are visualized in Figure 7.7, but compared to Figure 7.6, they do not offer any noteworthy improvement in quality.

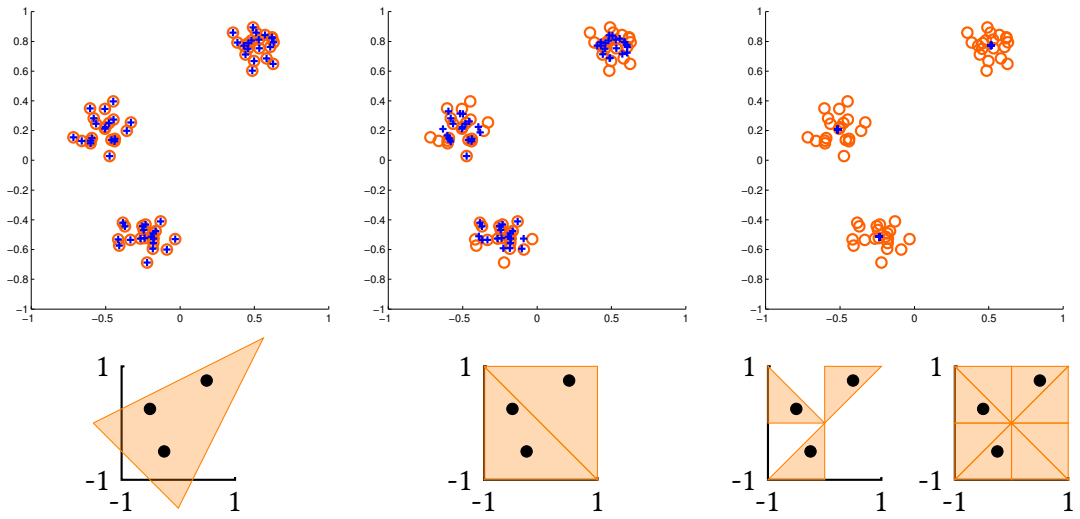

Figure 7.6: *Euclidean clustering* with $d = 2$, $k = 3$, $n = 60$. Top: Circles corresponding to input $\boldsymbol{b}_i$ and crosses corresponding to points parametrized by the $\boldsymbol{\lambda}_i$ extracted from (7.15). Bottom: Different choices of $\mathcal{P}$. From left to right: Minimal cover, non-separating cover, minimal separating cover, oversegmentation. Algorithm 7.2 was able to recover the optimal solution implied by the right plot in each scenario.

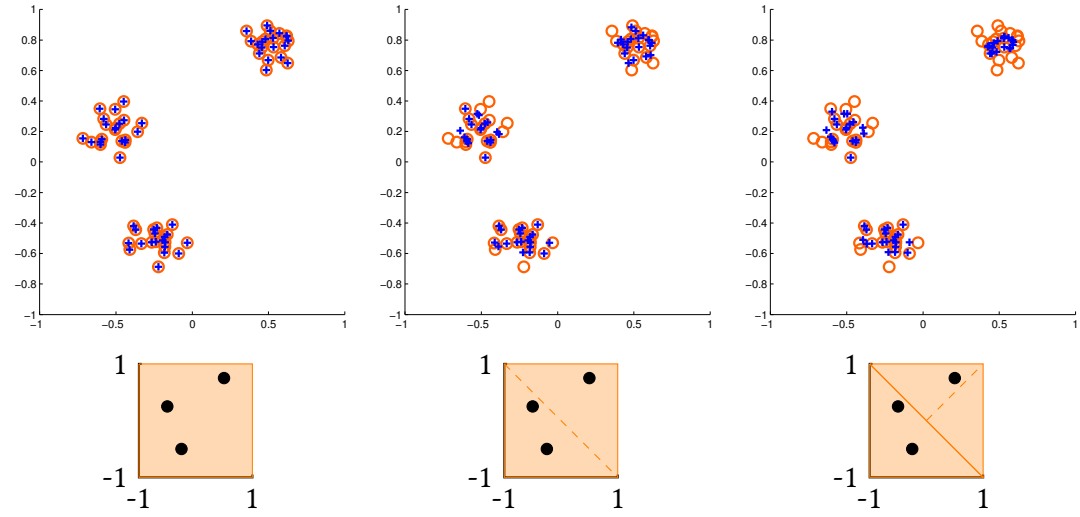

Figure 7.7: *Euclidean Clustering* with $d = 2$, $k = 3$, $n = 60$. Top: Circles corresponding to input $\boldsymbol{b}_i$ and crosses corresponding to points parametrized by the $\boldsymbol{\lambda}_i$ extracted from (7.15). Bottom: Variants described in Section (7.6) for describing the feasible set $[-1, 1]^2$. From left to right: $\mathcal{P}$ is the union of $1, 2, 3$ polytopes respectively. Vertices at dashed lines are unique rows in $\mathbf{V}$ and used in each bordering polytope.

As pointed out by Remark 7.17, we can easily restrict the feasible set in a way to force the optimal solution into specific regions. For example, by choosing each polytope in $\mathcal{P}$ to be a single vertex, we reduce (7.15) to an LP which aims to choose an optimal collection of locations from a discrete set of points, as can be seen in Figure 7.8. In this case, our experiments always returned the optimal solution.

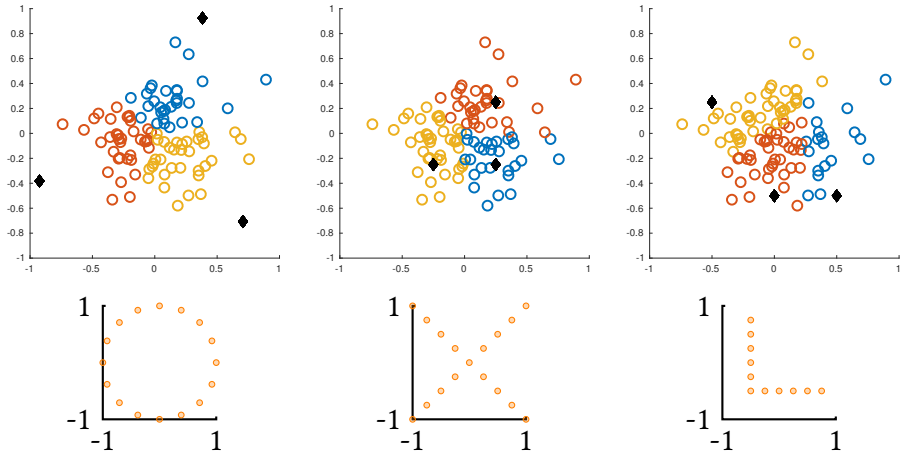

Figure 7.8: *Euclidean k-clustering* with $d = 2$, $k = 3$, $n = 100$ restricted to discrete $\mathcal{P}$. Top: Circles corresponding to input $\boldsymbol{b}_i$, diamonds corresponding to centers and colours corresponding to clusters. Bottom: Different choices of discrete $\mathcal{P}$.

Hyperplane Clustering

By choosing $\mathbf{A}_i = \boldsymbol{a}_i$ as row vectors in $\mathbb{R}^d$ and setting $b_i = 0$, (7.3) becomes the problem of choosing minimal $\langle \boldsymbol{a}_i, \boldsymbol{x}_j \rangle^2$ terms. We can interpret this as simultaneously choosing k hyperplanes parameterized by their normal vectors $\boldsymbol{x}_j$ and assigning the points $\boldsymbol{a}_i$ to them according to their weighted angle.

We can uniquely parametrize these hyperplanes by choosing an element $\boldsymbol{x}$ of their complement space which satisfies membership in both $S^d_{\|\cdot\|} = \left\{\boldsymbol{x} \in \mathbb{R}^d \;\middle|\; \|\boldsymbol{x}\| = 1\right\}$ for *any* fixed norm $\|\cdot\|$ and the 'upper halfspace' $H^d_+ = \left\{\boldsymbol{x} \in \mathbb{R}^d \;\middle|\; x_1 \geq 0\right\}$.

Note that the norm will weight each point $\boldsymbol{x} \in S^d_{\|\cdot\|} \cap H^d$ by $\|\boldsymbol{x}\|$. We will also write S^d_p for the $\|\cdot\|_p$ norms. In particular, even though any polyhedral approximation of $S^d_2 \cap H^d_+$ corresponds to a norm and can be used as $\mathcal{P}$, this will introduce a slight bias. The application of this approach is illustrated by Figure 7.9.

As an application of Example 7.39, we can also work directly with $S^2_2 \cap H^2_+$ by adding the quadratic constraint $\langle \boldsymbol{x}_j, \boldsymbol{x}_j \rangle = 1$ and choosing $\mathcal{P} \subseteq H^2_+$. Since S^2_2 is not polyhedral, we need to use 3-dimensional simplices for $\mathcal{P}$ in Figure 7.10 whereas 2-dimensional simplices suffice for the polyhedral approximation shown by Figure 7.9.

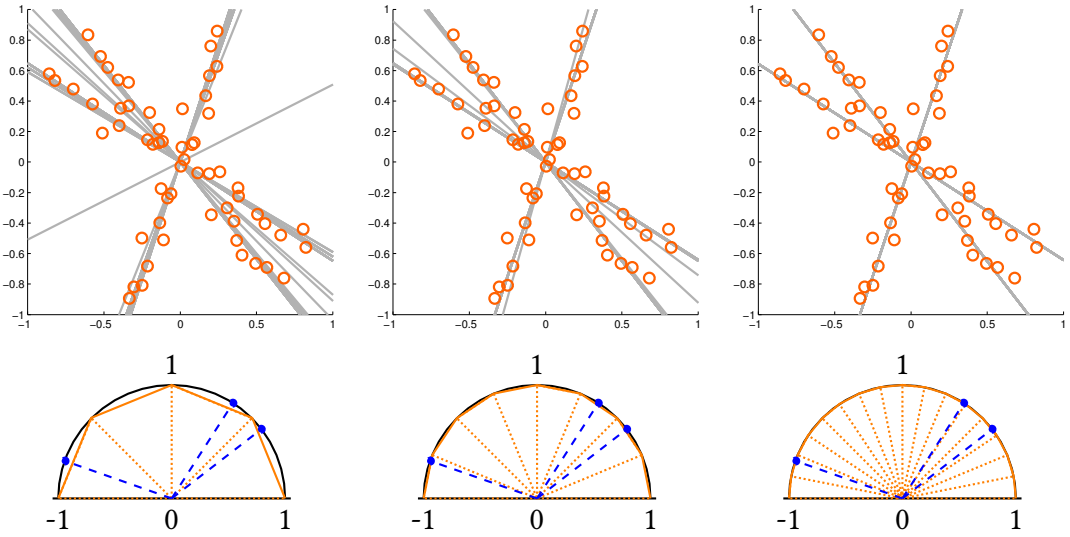

Figure 7.9: *Hyperplane k−clustering* with $d = 2$, $k = 3$, $n = 60$. Top: Circles corresponding to input $\boldsymbol{a}_i$ and lines parametrized by $\boldsymbol{\lambda}_i$ extracted from (7.15). Bottom: Approximations of $S_2^2 \cap H_+^2$ by polygonal lines. For better visibility the ends of each line segment are connected to the origin with a dotted line. Dashed lines end in optimal angles.

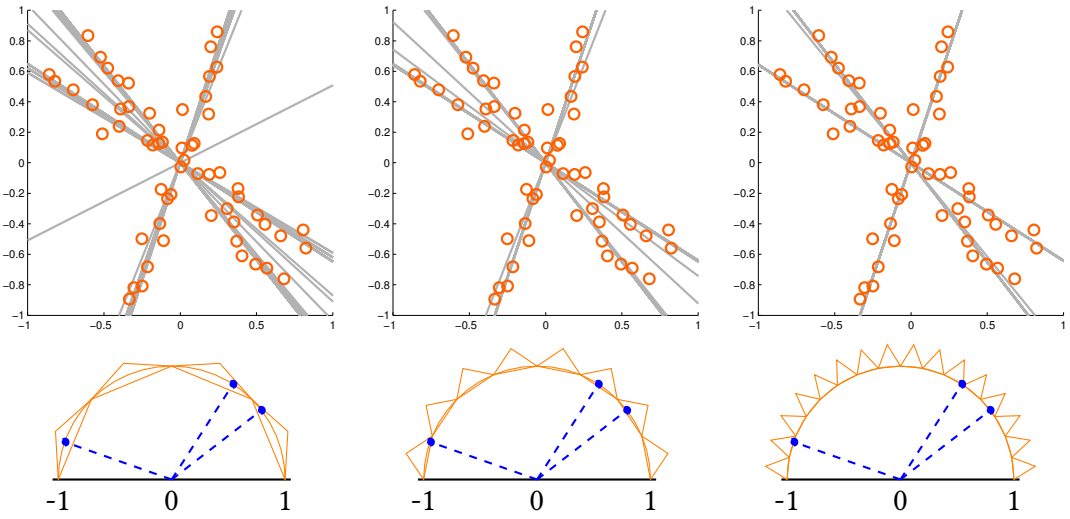

Figure 7.10: *Hyperplane Clustering* with $d = 2$, $k = 3$, $n = 60$. Top: Circles corresponding to input $\boldsymbol{a}_i$ and lines parametrized by $\boldsymbol{\lambda}_i$ extracted from (7.15). Bottom: The semicircle $S_2^2 \cap H_+^2$ is covered with triangles in $\mathcal{P}$. Dashed lines end in optimal angles.

Mixed Linear Regression

We can easily extend the hyperplane clustering to the more general case of *mixed linear regression* that we mentioned already in Section 7.4.2 by changing to homogeneous coordinates. In particular, we can encode the data point $\boldsymbol{a}_i \in \mathbb{R}^d$ as $(\boldsymbol{a}_i, 1) \in \mathbb{R}^{d+1}$ and try to find a hyperplane orthogonal to $(\boldsymbol{x}_j, -z_j) \in \mathbb{R}^d$ to get the minimization of terms like

$$\langle (\boldsymbol{a}_i, 1), (\boldsymbol{x}_j, z_j) \rangle^2 = (\langle \boldsymbol{a}_i, \boldsymbol{x}_j \rangle - z_j)^2,$$

which approximate membership in the affine hyperplane

$$\boldsymbol{a}_i \in H_{(\boldsymbol{x}_j, z_j)} = \left\{ \boldsymbol{a} \in \mathbb{R}^d \;\middle|\; \langle \boldsymbol{a}, \boldsymbol{x}_j \rangle = z_j \right\}.$$

Since the manipulation only amounts to lifting the input data $\{\boldsymbol{a}_i\}_{i \in [n]}$, this is just an instance of the hyperplane clustering problem in a space with dimension increased by 1. In particular, the problem of Figure 7.11 can be computed as an instance of clustering points from $\mathbb{R}^3$ into 2-dimensional hyperplanes.

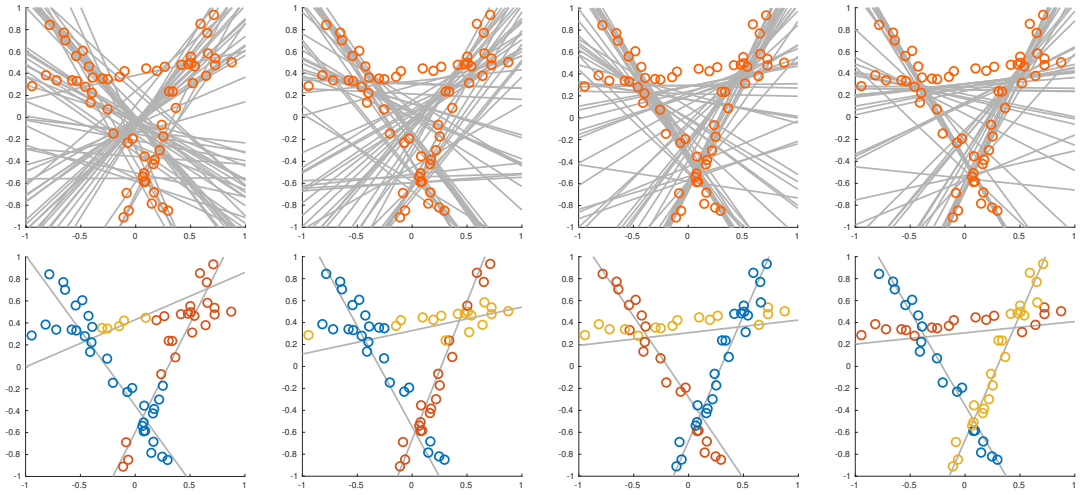

Figure 7.11: *Mixed Linear Regression* with $d = 2$, $k = 3$, $n = 60$ as a special case of *Hyperplane Clustering* with $d = 3$. Top: Circles corresponding to input $\boldsymbol{a}_i$ and lines parametrized by $\boldsymbol{\lambda}_i$ extracted from (7.15). Bottom: Gray lines corresponding to rounded solution of (7.15) and coloured data points according to the extracted clustering. From left to right: Discretization of $(S_2^2 \cap H_+^2) \times [-0.3, 0.3]$ into (2×8) , (4×4), (8×2) and (8×8) line segments, where $S_2^2 \cap H^2$ is approximated like in Figure 7.9.

Chapter 8

Conclusion

In this thesis, different representations for partitions were formulated and used to approximate partition problems. While doing so, special focus was on finding representations that are invariant under permutations of the individual parts, with the intention of evading the computational problems commonly associated with the lack of such invariance. A major tool in doing so was the method of moments, which was used to construct convex relaxations of the underlying polynomial optimization problems that can be solved by semidefinite optimization techniques.

In the first part of the thesis, we set up the machinery surrounding the method of moments in Chapters 2 and 3, while providing an explicit way of solving the underlying optimization problems. The algorithm we presented for this was the CLP-Newton method, which we introduced as a generalization of the LP-Newton method for self-dual cones. Experiments with the CLP-Newton method were provided, which indicated that, while the computation of a Newton step is quite expensive, only a small number of Newton steps were needed to come close to the optimal solution.

The second part of the thesis was concerned with classical partition problems that can be represented using various classes of matrices. For this, we started with assignment matrices and the related theory of orbitopes in Chapter 4, showcasing the problems that can arise in the convex setting when representations of partitions are not invariant under permutations of their parts.

We proceeded by showing how assignment matrices naturally give rise to partition matrices in Chapter 5, which explicitly described a connection that has been implicitly used in literature for a long time. Furthermore, the class of combinatorial projection matrices was introduced in Chapter 6, and its close connection to partitions matrices was established. As a main result, we used combinatorial projection matrices to give a new formulation of the colouring number of graphs and compared the associated relaxations to variants of the Lovász theta number. We were able to show that while the Szegedy number, a slight improvement of the Lovász theta number, is always at least as

good as our relaxation, both relaxations agree on the difficult class of vertex-transitive graphs. In particular, this result implies that relaxations for binary eigenvalues perform worse than relaxations of binary matrix-entries for this problem.

The third and final part of the thesis treated the challenging problem of affine Euclidean k-clustering in Chapter 7, which, to the best of our knowledge, has not been investigated in this generality prior to this thesis. While the classical Euclidean k-clustering allows for a reformulation based on the optimality condition of a subset of the variables, the generalized affine Euclidean k-clustering lacks this property, making it much harder to solve and relax. Assuming a certain separability property on the feasible set associated with the same subset of variables, we proposed a new approach for this problem. Using a simplicial cover of this feasible set, it was shown how to construct a reformulation that can be relaxed to yield bounds on the global optimum. In particular, it was shown how to extend this method to several variations of the affine Euclidean k-clustering problem that contain additional constraints, and the method was shown to work on a number of several different toy examples.

Further Research Directions

At the conclusion of this thesis, several intriguing questions are still open.

On the numerical side, the CLP-Newton method could be improved by using a more efficient way to solve the minimum-norm-point algorithm. While all relaxations in this thesis would benefit from a computational improvement of the underlying CLP solvers, the affine Euclidean k-clustering problem would benefit the most, given that the scaling of the method using simplicial covers makes the method currently prohibitive for practical application. In particular, the runtime of the method might be improved by utilizing recently published SOCP-hierarchies instead of the classical SDP-hierarchies associated with the method of moments. In general, switching to SOCP-hierarchies tends to improve the runtime considerably at only a marginal cost in solution quality.

On the theoretical side, it would be interesting to get a better understanding about the difference between partition matrices and combinatorial projection matrices in terms of relaxation quality. Any insights on this matter might help us to answer the deeper question of whether binary matrix entries or binary eigenvalues are a better modeling paradigm for relaxation based approaches. For the graph colouring problem in particular, it would be interesting to know if there is a class of additional constraints that could be added to combinatorial projection matrices to reduce the gap to the Szegedy number.

In terms of the affine Euclidean k-clustering, it would be helpful to know whether conditions can be given that guarantee a fixed approximation ratio of our method.

While this is unlikely for the general case due to the high expressiveness of the problem, there might be mild conditions under which certain problems have provable guarantees, even in the first stage of the hierarchy we considered. From a practical perspective, the separability assumption is mostly unexplored, and a list of problems with this property would be desirable to get further insights into the applicability of our method.

Bibliography

[AB09] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.

[ACKS15] Pranjal Awasthi, Moses Charikar, Ravishankar Krishnaswamy, and Ali Kemal Sinop. The hardness of approximation of euclidean k-means. *CoRR*, abs/1502.03316, 2015.

[ADHP09] Daniel Aloise, Amit Deshpande, Pierre Hansen, and Preyas Popat. Np-hardness of euclidean sum-of-squares clustering. *Machine Learning*, 75(2):245–248, 2009.

[AL12] Miguel F Anjos and Jean B Lasserre. *Handbook on semidefinite, conic and polynomial optimization, International Series in Operations Research & Management Science, vol. 166*. Springer, New York, 2012.

[Bac11] Francis R. Bach. Learning with submodular functions: A convex optimization perspective. *CoRR*, abs/1111.6453, 2011.

[BEF00] Benno Büeler, Andreas Enge, and Komei Fukuda. *Exact Volume Computation for Polytopes: A Practical Study*, pages 131–154. Birkhäuser Basel, Basel, 2000.

[Ber75] Sven Berg. Some properties and applications of a ratio of stirling numbers of the second kind. *Scandinavian Journal of Statistics*, 2(2):91–94, 1975.

[BPS66] Richard A Brualdi, Seymour V Parter, and Hans Schneider. The diagonal equivalence of a nonnegative matrix to a stochastic matrix. *Journal of Mathematical Analysis and Applications*, 16(1):31 – 50, 1966.

[BPT13] Grigoriy Blekherman, Pablo A. Parrilo, and Rekha Thomas, editors. *Semidefinite Optimization and Convex Algebraic Geometry*. SIAM, 2013.

[BS94] Mihir Bellare and Madhu Sudan. Improved non-approximability results. In *Proceedings of the Twenty-sixth Annual ACM Symposium on Theory of Computing*, STOC '94, pages 184–193. ACM, 1994.

[CLO15] David A. Cox, John Little, and Donal O'Shea. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*. Springer, 4th edition, 2015.

[DGK63] L. Danzer, B. Grünbaum, and V. Klee. *Helly's Theorem and Its Relatives*. Proceedings of symposia in pure mathematics: Convexity. American Mathematical Society, 1963.

[Die05] Reinhard Diestel. *Graph Theory (Graduate Texts in Mathematics)*. Springer, August 2005.

[DR07] Igor Dukanovic and Franz Rendl. Semidefinite programming relaxations for graph coloring and maximal clique problems. *Math. Program.*, 109(2):345–365, 2007.

[EZC10] Ernie Esser, Xiaoqun Zhang, and Tony F. Chan. A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science. *SIAM J. Imaging Sciences*, 3(4):1015–1046, 2010.

[FHYZ08] Satoru Fujishige, Takumi Hayashi, Kei Yamashita, and Uwe Zimmermann. Zonotopes and the LP-Newton method. *Optimization and Engineering*, 10(2):193–205, 2008.

[FK09] Yuri Faenza and Volker Kaibel. Extended formulations for packing and partitioning orbitopes. *Math. Oper. Res.*, 34(3):686–697, 2009.

[GL08] Nebojsa Gvozdenovic and Monique Laurent. The operator psi for the chromatic number of a graph. *SIAM Journal on Optimization*, 19(2):572–591, 2008.

[Har68] Bernard Harris. Statistical inference in the classical occupancy problem unbiased estimation of the number of classes. *Journal of the American Statistical Association*, 63:837–847, 1968.

[HJ16] Paul Hand and Babhru Joshi. A convex program for mixed linear regression with a recovery guarantee for well-separated data. *arXiv preprint arXiv:1612.06067*, 2016.

[HS85] Dorit S. Hochbaum and David B. Shmoys. A Best Possible Heuristic for the k-Center Problem. *Mathematics of Operations Research*, 10(2):180–184, May 1985.

[Jag13] Martin Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In Sanjoy Dasgupta and David Mcallester, editors, *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*,

volume 28, pages 427–435. JMLR Workshop and Conference Proceedings, 2013.

[KP08] Volker Kaibel and Marc E. Pfetsch. Packing and partitioning orbitopes. *Math. Program.*, 114(1, Ser. A):1–36, 2008.

[Las15] Jean Bernard Lasserre. *An Introduction to Polynomial and Semi-Algebraic Optimization*, volume 52. Cambridge University Press, 2015.

[Lau03] Monique Laurent. A comparison of the Sherali-Adams, Lovász-Schrijver, and Lasserre relaxations for 0-1 programming. *Mathematics of Operations Research*, 28(3):470–496, 2003.

[Lov79] László Lovász. On the shannon capacity of a graph. *IEEE Trans. Inf. Theor.*, 25(1):1–7, 1979.

[Lov87] László Lovász. *An algorithmic theory of numbers, graphs and convexity*, volume 50. SIAM, 1987.

[NT08] Arkadi S. Nemirovski and Michael J. Todd. Interior-point methods for optimization. *Acta Numerica*, 17:191–234, 2008.

[OCPB16] Brendan O'Donoghue, Eric Chu, Neal Parikh, and Stephen Boyd. Conic optimization via operator splitting and homogeneous self-dual embedding. *Journal of Optimization Theory and Applications*, 169(3):1042–1068, 2016.

[PW07] Jiming Peng and Yu Wei. Approximating K-means-type Clustering via Semidefinite Programming. *SIAM J. Optimization*, 18(1):186–205, 2007.

[PX05] Jiming Peng and Yu Xia. A new theoretical framework for k-means-type clustering. In *Foundations and advances in data mining*, pages 79–96. Springer Berlin Heidelberg, 2005.

[Ren10] Franz Rendl. Semidefinite relaxations for integer programming. In Michael Jünger, Thomas M. Liebling, Denis Naddef, George L. Nemhauser, William R. Pulleyblank, Gerhard Reinelt, Giovanni Rinaldi, and Laurence A. Wolsey, editors, *50 Years of Integer Programming 1958-2008 - From the Early Years to the State-of-the-Art*, chapter 18, pages 647–686. Springer, 2010.

[Roc09] R. Tyrrell Rockafellar. *Convex Analysis*. Springer, 2009.

[Sch79] Alexander Schrijver. A comparison of the Delsarte and Lovász bounds. *Information Theory, IEEE Transactions on*, 25(4):425–429, 1979.

[Sch03] Alexander Schrijver. *Combinatorial Optimization - Polyhedra and Efficiency*. Springer, 2003.

[SR16] Francesco Silvestri and Gerhard Reinelt. The LP-Newton method and conic optimization. *ArXiv e-prints*, November 2016.

[SRS15] Francesco Silvestri, Gerhard Reinelt, and Christoph Schnörr. A convex relaxation approach to the affine subspace clustering problem. In *Pattern Recognition - 37th German Conference, GCPR 2015, Aachen, Germany, October 7-10, 2015, Proceedings*, pages 67–78, 2015.

[SRS16] Francesco Silvestri, Gerhard Reinelt, and Christoph Schnörr. Symmetry-free SDP relaxations for affine subspace clustering. *ArXiv e-prints*, July 2016.

[Sta11] Richard P. Stanley. *Enumerative Combinatorics: Volume 1*. Cambridge University Press, New York, NY, USA, 2nd edition, 2011.

[Stu11] Raman Sanyal ; Frank Sottile ; Bernd Sturmfels. Orbitopes. *Mathematika : a journal of pure and applied mathematics*, 57(2):275–314, 2011.

[Sze94] Mario Szegedy. A note on the theta number of Lovász and the generalized delsarte bound. In *35th Annual Symposium on Foundations of Computer Science, Santa Fe, New Mexico, USA, 20-22 November 1994*, pages 36–39, 1994.

[Tro15] Nicolas Trotignon. Perfect graphs: a survey. *arXiv preprint arXiv:1301.5149*, 2015.

[TTT96] Kim-Chuan Toh, Michael J. Todd, and Reha H. Tütüncü. *SDPT3 — a MATLAB software package for semidefinite programming*, 1996.

[TTT03] Reha H. Tütüncü, Kim-Chuan Toh, and Michael J. Todd. Solving semidefinite-quadratic-linear programs using SDPT3. *Math. Program.*, 95(2):189–217, 2003.

[Wol] Philip Wolfe. Finding the nearest point in a polytope. *Math. Program.*, 11(1):128–149.

[XWI05] Rui Xu and Donald Wunsch II. Survey of Clustering Algorithms. *IEEE Trans. Neural Networks*, 16(3):645–678, May 2005.

[Zie95] Günter M. Ziegler. *Lectures on polytopes*. Graduate Texts in Mathematics. Springer, New York, 1995.

Index

T

U

V