Technische Universität München
TUM School of Computation, Information and Technology

Homologous Circulations, Voronoi Cells, and Densest Subgraphs

Ina Seidel

Vollständiger Abdruck der von der TUM School of Computation, Information and Technology der Technischen Universität München zur Erlangung einer

Doktorin der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitz: Prof. Dr. Ulrich Bauer

Prüfer*innen der Dissertation:

1. Prof. Dr. Stefan Weltge
2. Prof. Dr. Matthias Walter

Die Dissertation wurde am 11.01.2023 bei der Technischen Universität München eingereicht und durch die TUM School of Computation, Information and Technology am 11.04.2023 angenommen.

Abstract

There are many research directions in the field of discrete optimization, all of which interact with one another. This thesis is threefold, and each part focuses on a different aspect of discrete optimization.

The first chapter deals with a modified version of one of the most classic problems in combinatorial optimization, namely finding a minimum-cost flow in a directed graph D. For the modified version, we consider a surface-embedded digraph D together with non-negative costs on its arcs. Given any integer circulation in D, we study the problem of finding a minimum-cost non-negative integer circulation in D that is *homologous* over the integers to the given circulation. For *orientable* surfaces, polynomial-time algorithms have been obtained for different variants of this problem. We complement these results by showing that the convex hull of feasible solutions has a very simple polyhedral description. In contrast, we show that the problem is strongly NP-hard for general *non-orientable* surfaces, and give the first polynomial-time algorithm for surfaces of fixed genus. For the latter, we provide a characterization of $\mathbb{Z}$-homology that allows us to recast the problem as a special integer program, which can be efficiently solved using some general integer programming techniques.

Many techniques used in integer optimization are based on lattice theory. In the second chapter, a basic object associated to a lattice is studied: *Voronoi cells* of lattices. We consider these with respect to their polyhedral description, and aim for finding small representations using lifts. Such lifts may yield compressed representations of polytopes which are typically used to construct small-size linear programs. We construct an explicit d-dimensional lattice such that every lift of the respective Voronoi cell has $2^{\Omega(d/\log d)}$ facets. On the positive side, we show that Voronoi cells of d-dimensional root lattices and their dual lattices have lifts with $\mathcal{O}(d)$ and $\mathcal{O}(d \log d)$ facets, respectively.

Keeping the polyhedral view, we investigate another combinatorial problem in the third chapter. Given any undirected graph $G = (V, E)$, we deal with the task of finding a *densest subgraph* of G, where the density is given by the average node degree. This problem can be formulated as a linear program. We characterize adjacencies in the underlying polytope, and tailor a recently published algorithm [Dad+22] for general convex optimization problems in the separation oracle model to the densest subgraph problem.

Zusammenfassung

Diese Arbeit besteht aus drei Teilen und jeder Teil konzentriert sich auf einen anderen Aspekt der diskreten Optimierung.

Das erste Kapitel befasst sich mit einer Variante eines der klassischen Probleme der kombinatorischen Optimierung, nämlich der Suche nach einem kosten-minimalen Fluss in einem gerichteten Graphen D. Für die modifizierte Version betrachten wir D zellulär eingebettet in eine Fläche mit nicht-negativen Kosten auf seinen Bögen. Sei y eine beliebige ganzzahlige Zirkulation in D. Die Aufgabe ist es nun eine kosten-minimale, nicht-negative ganzzahlige Zirkulation in D zu finden, die über die ganzen Zahlen *homolog* zur gegebenen Zirkulation y ist. Auf *orientierbaren* Flächen ist dieses Problem effizient lösbar. Wir ergänzen diese Resultate, indem wir zeigen, dass die konvexe Hülle der zulässigen Lösungen eine sehr einfache polyedrische Beschreibung hat. Im Gegensatz dazu zeigen wir, dass das Problem für allgemeine *nicht-orientierbare* Flächen NP-schwer ist und geben den ersten polynomiellen Algorithmus für Flächen mit konstantem Euler-Geschlecht. Für letzteres liefern wir eine Charakterisierung der Homologieklassen über den ganzen Zahlen, die es uns erlaubt, das Problem als ein spezielles ganzzahliges Programm umzuformen, das mithilfe einiger Resultate aus der ganzzahligen Optimierung effizient gelöst werden kann.

Viele Techniken, die in der ganzzahligen Optimierung verwendet werden, basieren auf Resultaten über Gitter. Im zweiten Kapitel untersuchen wir *Voronoi-Zellen* von Gittern. Wir betrachten diese mit Blick auf ihre polyedrische Beschreibung und versuchen, mithilfe von erweiterten Formulierungen, kleine Darstellungen von ihnen zu finden. Erweiterten Formulierungen werden typischerweise studiert, um linearen Programme mit möglichst wenigen Ungleichungen zu finden. Wir konstruieren ein explizites d-dimensionales Gitter, sodass jede erweiterte Formulierung der entsprechenden Voronoi-Zelle $2^{\Omega(d/\log d)}$ viele Ungleichungen benötigt. Auf der positiven Seite zeigen wir, dass Voronoi-Zellen von d-dimensionalen Gittern, die von Wurzelsystemen erzeugt wurden, und deren duale Gitter erweiterte Formulierungen mit $\mathcal{O}(d)$ bzw. $\mathcal{O}(d \log d)$ Ungleichungen besitzen.

Während wir den polyedrischen Blickwinkel beibehalten, untersuchen wir im dritten Kapitel ein weiteres kombinatorisches Problem. Für einen beliebigen ungerichteten Graphen $G = (V, E)$ ist ein *dichtester Teilgraph* von G gesucht, wobei die Dichte durch den durchschnittlichen Knotengrad gegeben ist. Dieses Problem kann als lineares Programm formuliert werden. Wir charakterisieren die Adjazenzen im zugrundeliegenden Polytop und schneiden einen kürzlich veröffentlichten Algorithmus [Dad+22] für allgemeine konvexe Optimierungsprobleme auf das Problem des dichtesten Teilgraphen zu.

Acknowledgements

First of all, I would like to thank my supervisor Stefan Weltge. Without him, I probably wouldn't have done my doctorate at all. He already supervised me during my master's thesis, and we spent hours together at the blackboard. This atmosphere of collaboration, full of appreciation and enthusiasm for the beauty of mathematics, set me on the path to further years of research in the field of integer optimization. In the last years, his office door was literally always open for me, and even during (pandemic-related) days of working at home, he was always just a phone call away. He made it possible for us to attend IPCO together as a group and invited me to a wonderful workshop in Cargése. I thank him for hours-long, enlightening, inspiring, and motivating meetings, for a very positive and enjoyable yet productive working atmosphere without any pressure, and for his open ears on all topics, whether it was a research problem, a career question, general organizational stuff, or a rather personal issue.

Of course, Stefan is not solely responsible for the wonderful working atmosphere. I really would like to thank every single person on our "Discrete Optimization"-floor. Let me just name the ones that were probably most important for me and my way: The first office I walked by every morning, is the one of Anja and Mia, where I always found a small candy and a lovely chat, then, René, who gave me a warm welcome on my first day, Jamico, who you should definitely visit in need of a break, since he always has something interesting to tell, Michael, who helped me with many IT issues, and of course, my friend and office-mate Stefan. I'll miss our *weekly* cake break, enjoying "Studitopf" in the Mensa together, chatting with a cup of coffee - just being able to visit each other in the office at any time to have a break, chat, laugh, and receive (and hopefully also sometimes give) encouragement. I would particularly like to thank Anja and Stefan for being my friend and helping me through some tough days, and also for their offer of proofreading my manuscript.

In my first year, fortunately, Sarah Morell visited our group and became a co-author. Working with her was so much fun, and she was also a good support when I was struggling with all the new tasks I faced at the beginning of my doctorate studies. I also would like to thank my co-author Matthias Schymura for sharing his expertise in hours-long meetings, many valuable discussions, and for meeting me at eye level, although academically I am still in my infancy compared to him.

I would like to thank Ulrich Bauer for chairing my dissertation and for valuable discussions on my research topic. He shared his expertise on topological basics needed in my project on homologous circulations. Furthermore, I would like to thank Matthias Walter for taking time and reviewing my thesis.

My final thanks go to all the people who have supported me on my way without understanding a single word of what I have been doing for the past few years. Especially, I would like to thank my parents for their support, which allowed me to follow my path completely free without any pressure, and my partner Eric, who, contrary to plan, stayed with me in Munich for a “little” longer.

Contents

Chapter 1

Introduction

Discrete optimization is usually concerned with finding an optimal object within a finite or at least countable set. Many optimization tasks in this field rely on combinatorial structures such as graphs. Moreover, most discrete optimization problems can be recast as an integer program

$$\min\{c^\intercal x : Ax \leqslant b, x \in \mathbb{Z}^n\}, \tag{IP}$$

with $A \in \mathbb{Z}^{m\times n}, b \in \mathbb{Z}^m$, and $c \in \mathbb{R}^n$, which is equivalent to optimizing a linear objective function over the polyhedron $\operatorname{conv}\{x \in \mathbb{Z}^n : Ax \leqslant b\}$. Therefore, besides combinatorial and integer optimization, polyhedral combinatorics also plays an important role within discrete optimization. This thesis is threefold, and each part focuses on different aspects of discrete optimization. The interplay between the different disciplines in the field of discrete optimization within the chapters of this thesis is depicted in Figure 1.1.

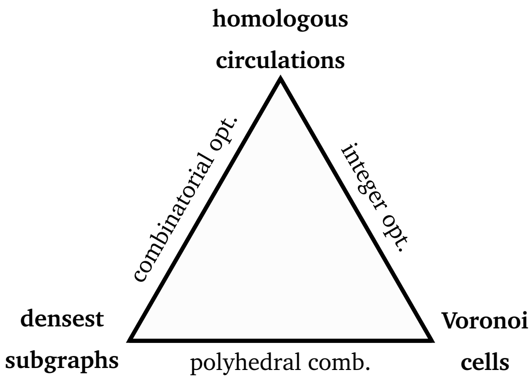


Figure 1.1: The three topics *homologous circulations* in Chapter 2, *Voronoi cells* in Chapter 3, and *densest subgraphs* in Chapter 4 of this thesis and their interplay with the different branches in discrete optimization.

For instance, finding minimum-cost circulations satisfying some additional constraints, discussed in Chapter 2, or detecting dense subgraphs, discussed in Chapter 4 are purely combinatorial problems. Combinatorial and integer optimization are very closely related to each other, since as mentioned above most combinatorial optimization problems can be formulated as integer programs. An example for this can be seen in Chapter 2. Here, we model our combinatorial circulation problem as an integer program that can be solved using general integer programming techniques. Moreover, studying these special circulations was motivated by a recent series of work considering general integer programs with constraint matrices A whose submatrices have bounded determinants [Fio+22].

Another common approach for solving discrete optimization problems deals with finding good representations of the associated polyhedral structure. For instance in Chapter 4, we consider a linear programming formulation of the densest subgraph problem, and study its feasible region. While the number of facets of this polyhedron is very small, this is not the case for most polyhedra arising in discrete optimization. However, it turns out that many polytopes that arise in the study of polyhedral combinatorics are linear projections of higher-dimensional polytopes, also called *lifts*, with significantly fewer facets. In Chapter 3, we investigate whether this also holds true for Voronoi cells of lattices, which are basic polytopes in the study of lattice problems. Since lattices play a crucial role in integer programming, Chapter 3 is concerned with both integer optimization and polyhedral combinatorics.

In what follows, we give a summary of the three parts of this thesis and highlight their main results.

Homologous circulations

Our main motivation for the study of minimum-cost circulations with additional constraints in Chapter 2 is a result regarding one of the fundamental problems in combinatorial optimization, namely the *stable-set problem*. Therefore, let us start with a small detour, taking a closer look at the stable-set problem.

Given a graph G, a stable-set in G is a subset $S \subseteq V(G)$ of the nodes in G such that no two nodes in S are adjacent. The unweighted stable-set problem asks for a stable-set of maximum cardinality, whereas in the weighted version the graph is given together with weights $w \in \mathbb{R}^{V(G)}$ on the nodes of G, and the problem asks for a stable-set S of maximum weight $\sum_{s \in S} w(s)$. Since node-covers are precisely the complement of stable-sets, and the node-cover problem, also known as the vertex-cover problem, is proven to be NP-complete in a paper by Karp [Kar72] in 1972, the decision version of the unweighted stable-set problem is among the very first problems that are shown to be NP-complete, which underlines the importance of the stable-set problem. Therefore, unless $\mathrm{P} = \mathrm{NP}$, we cannot find a polynomial-time algorithm that solves the stable-set problem on general graphs. The stable-set problem can be modeled as an integer program via $\max\left\{w^\intercal x : Ax \leqslant \mathbf{1}, x \in \mathbb{Z}_{\geqslant 0}^{V(G)}\right\}$, where A denotes the edge-node-incidence matrix of G. For bipartite graphs, the corresponding integer program is solvable in polynomial time. Recall that bipartite graphs are graphs without any odd cycle. A natural extension to this class is given by the family of graphs that do not contain $k+1$ node-disjoint odd cycles, where $k \geqslant 0$ is a fixed constant. The smallest number k such that a graph G satisfies the claimed property is called the *odd cycle packing number* of G, denoted by $\mathrm{ocp}(G)$.

This extension also translates to a property regarding the incidence matrix. To formalize this, we use the following notion. A matrix $A \in \mathbb{Z}^{m\times n}$ is called totally Δ-modular, if every determinant of a square submatrix of A is bounded by Δ in absolute value. It is well known that the incidence matrix A of G is totally $2^{\mathrm{ocp}(G)}$-modular [GKS95]. In general, determining the complexity status of integer programs (IP) with totally Δ-modular constraint matrix A for some fixed constant Δ is an open problem, see e.g., [AWZ17]. However, special cases of the above question have been solved within the last few years. For instance, Fiorini, Joret, Weltge, and Yuditsky [Fio+22] developed a strongly polynomial-time algorithm for integer programs with totally Δ-modular coefficient matrices A that contain at most two non-zero entries in each row for every fixed constant Δ, in 2022. Their work was built on a prior published work [Con+20a] about the stable-set problem for graphs with bounded genus and bounded odd cycle packing number. Within their approach, they crucially exploit that the considered graphs G admit a special embedding in a surface of bounded genus. This embedding yields an orientation of the dual graph D such that stable-sets in G directly correspond to non-negative integer circulations in D satisfying a particular topological constraint. Informally speaking, flow sent along the boundary of a face in D corresponds to excluding the corresponding node in the "primal" graph from the stable-set. Using this construction, their special stable-set problem can be efficiently reduced to some minimum-cost circulation problem with an additional topological constraint.

The standard minimum-cost circulation problem is among the most-studied problems in combinatorial optimization. Given a digraph $D = (V, A)$ with costs c on its arcs, it asks for an assignment of flow $f \in \mathbb{R}_{\geqslant 0}^{A}$ satisfying the flow conservation constraints requiring that the amount of flow "entering" a node v equals the amount "leaving" v, minimizing the total cost $c^\intercal f$.

This problem is studied in various versions. For instance, one may introduce lower and upper capacity bounds on the arcs, or require satisfying certain non-zero demands at some nodes. Besides many efficient combinatorial algorithms that solve these general circulation problems, we mention that they also admit a nice linear programming formulation. The flow conservation constraints are linear constraints, whose coefficient matrix equals the node-arc-incidence matrix of D, which is totally unimodular since D

is a digraph. Therefore, optimal circulations can always be chosen to be integral.

While these general circulation problems are extensively studied in combinatorial optimization, much less seems to be known about this version.

Problem 1.1 (see Problem 2.1)**.** *Given a directed graph D cellularly embedded in a surface together with non-negative costs c on its arcs and any integer circulation y in D, find a minimum-cost non-negative integer circulation in D that is $\mathbb{Z}$-homologous to y.*

Here, a circulation x is said to be *$\mathbb{Z}$-homologous to y* if their difference $x - y$ is a linear combination of facial circulations with integer coefficients, where a *facial circulation* is a circulation that sends one unit along the boundary of a single face, see Figure 1.2. Remember that informally speaking, a *face* is just a connected region that arises when cutting out the nodes and arcs of the embedded digraph "drawn" on the surface. As an example for Problem 1.1, if y is the all-zeros circulation, then $y = \mathbf{0}$ itself is clearly an optimal solution to Problem 2.1. The same holds true for D being planar, meaning that D is cellularly embedded in the sphere, since on the sphere all circulations are homologous, and therefore independent of y the all-zeros circulation is optimal. However, for general y and general surfaces, the all-zeros circulation might not be feasible.

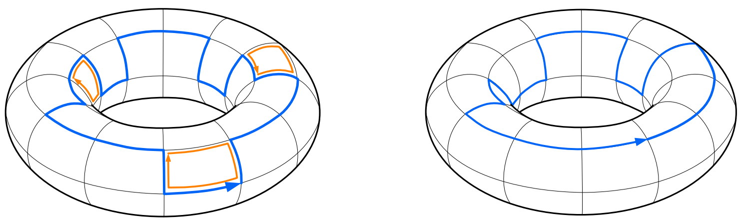


Figure 1.2: The circulations that send one unit along the blue directed cycles on the torus are $\mathbb{Z}$-homologous. In fact, their difference is the sum of three facial circulations which are depicted in orange.

We distinguish between *orientable*, such as the sphere and the torus, and *non-orientable* surfaces, such as the projective plane and the Klein bottle. In either case, we measure the "complexity" of a surface with the *(Euler) genus g*. The sphere has genus $g = 0$, the projective plane $g = 1$, and for the Klein bottle $g = 2$ holds. In Chapter 2, we will provide a more detailed introduction to surfaces and homology.

Due to its already displayed connection to the stable-set problem, one might already suspect Problem 1.1 to be hard, and this is indeed true. However, special cases are tractable. The main contribution of Chapter 2 is the following theorem.

Theorem 1.2 (see Theorem 2.4)**.** *Problem 1.1 can be solved in polynomial time on surfaces of fixed genus.*

For orientable surfaces this even holds true for general genus, which was known before, see Chambers, Erickson, and Nayyeri [CEN12]. We complement these results by showing that the convex hull of feasible solutions to Problem 1.1 has a very simple polyhedral description.

To prove Theorem 1.2 on non-orientable surfaces, we first provide a characterization of $\mathbb{Z}$-homology via $g-1$ linear constraints and one parity constraint. This characterization allows us to recast the problem as a special integer program using only properties of circulations independently of the embedding and the surface. This integer program has both a fixed number of constraints and the absolute values of entries in the constraint matrix bounded by a constant. Such integer programs can be efficiently solved using proximity results and dynamic programming, see [Art+16].

We note that Problem 1.1 on surfaces of fixed genus g can also be recast as another integer program (IP) with totally 2^g-modular constraint matrix A just like the mentioned stable-set problem with bounded odd cycle packing number. It can be shown that this integer program describing Problem 1.1 meets the properties required in [Fio+22] providing an alternative proof of Theorem 1.2. However, when we developed our algorithm, these results were not yet known.

The connection between the purely combinatorial stable-set problem or the minimum-cost circulation problem and these special integer programs, shows how closely the different branches such as combinatorial and integer optimization are intertwined. Chapter 2 contributes to both subfields and at the same time exploits techniques from both areas.

This chapter has been created on the basis of a joint publication, and the results are also presented in:

> S. Morell, I. Seidel, and S. Weltge. "Minimum-cost integer circulations in given homology classes". In: *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM. 2021, pp. 2725–2739.

Voronoi cells

In the previous section, we already mentioned some approaches to solve special integer programs efficiently, although general integer programming covers NP-hard problems.

Besides that, a landmark result by Lenstra [Len83], improved by Kannan [Kan87] and Dadush [Dad12], provides a polynomial-time algorithm that solves integer programs (IP) with a constant number of variables n. Considering any polyhedron $P = \{x \in \mathbb{R}^n : Ax \leqslant b\}$ with A, b integral, the feasibility version of integer programming asks whether there exists a point $z \in P \cap \mathbb{Z}^n$. Lenstra's algorithm efficiently decomposes this problem into a bounded number of subproblems of smaller dimension considering certain parallel slices of P. This approach heavily relies on the famous flatness theorem, first proven by Khinchine [Khi48] stating that P either does contain a lattice point or is flat in one direction in the sense that P is only contained in very few lattice hyperplanes orthogonal to this direction. Here, a *lattice* Λ is the image of $\mathbb{Z}^n$ under an injective linear map.

Computing a flat direction involves finding a short vector in the dual lattice. Therefore, Lenstra's algorithm crucially relies on lattice algorithms (approximately) solving the *shortest vector problem*. Given a lattice $\Lambda \in \mathbb{R}^n$ the shortest vector problem asks for finding a non-zero lattice point $z \in \Lambda \backslash \{\mathbf{0}\}$ minimizing $\|z\|$. The inhomogeneous version, or a generalization of this problem is the *closest vector problem*. For a target vector $t \in \mathbb{R}^n$, the closest vector problem asks for a lattice point $z \in \Lambda$ that is closest to t, meaning minimizing $\|t - z\|$. Here, $\|\cdot\|$ denotes the Euclidean norm.

These two problems are classic lattice problems. Both are assumed to be NP-hard and the closest lattice problem is even shown to be hard to approximate, see [Emd81] for the general hardness results. Dinur, Kindler, and Safra [DKS98] show that approximating the closest vector problem within a factor of $n^{(1/\log\log n)}$ is NP-hard. Special cases of these problems are even assumed to be hard in average-case. Due to their conjectured hardness, these lattice problems play a crucial role in cryptography, and serve as a security measure for many cryptographic schemes, see e.g., [MG02] for an overview.

One famous result, achieving a polynomial-time algorithm for approximating the shortest lattice vector within a factor of $2^{(n-1)/2}$ is the lattice reduction algorithm by Lenstra, Lenstra, and Lovász, also called LLL-algorithm [LLL82]. This algorithm provides a sufficiently accurate approximation for Lenstra's algorithm, and it has applications in many other areas, such as cryptoanalysis. In 2010, Micciancio & Voulgaris [MV13] gave the first deterministic algorithms solving both the shortest and the closest vector problem using only $2^{\tilde{O}(n)}$-time and space. Their approach heavily relies on *Voronoi cell* computations, which motivates our studies.

The Voronoi cell $\mathrm{VC}(\Lambda)$ of a lattice $\Lambda \subseteq \mathbb{R}^n$ is the set of all points in $\mathrm{lin}(\Lambda)$ for which the origin is among the closest lattice points, i.e.,

$$\mathrm{VC}(\Lambda) := \{x \in \mathrm{lin}(\Lambda) : \|x\| \leqslant \|x - z\| \text{ for all } z \in \Lambda\}\,,$$

where $\mathrm{lin}(\cdot)$ denotes the linear hull. The Voronoi cell of a lattice has many favorable properties. First of all, it is a polytope. Moreover, it is centrally symmetric, and the lattice translates $z + \mathrm{VC}(\Lambda)$, $z \in \Lambda$, induce a facet-to-facet tiling of $\mathrm{lin}(\Lambda)$, see Figure 1.3

As mentioned, the closest vector problem is one of the main motivations for our study of Voronoi cells. In view of that, we note that z is a closest lattice point of some $t \in \mathbb{R}^n$ if and only if $t - z \in \mathrm{VC}(\Lambda)$. Thus, a description of $\mathrm{VC}(\Lambda)$ in terms of polynomially many linear inequalities yields an efficient algorithm for testing whether a lattice point is a closest lattice vector to t. However, in view of the fact that the closest vector problem is

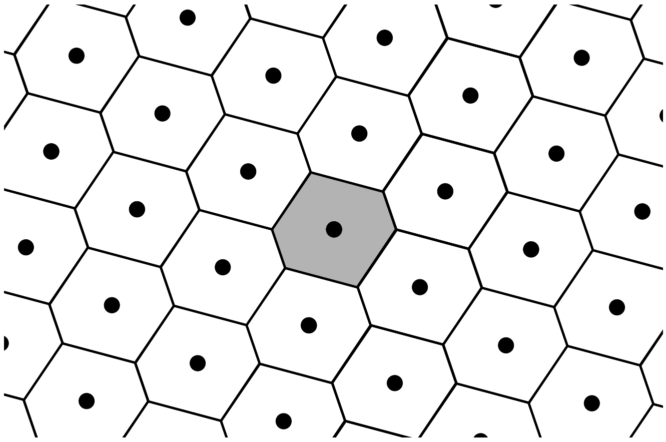


Figure 1.3: A lattice in $\mathbb{R}^2$ together with its Voronoi cell (shaded gray) and the corresponding tiling of the plane via its lattice translates.

NP-hard, and the belief that $\mathrm{NP} \neq \mathrm{coNP}$, we do not expect efficient algorithms that, for general lattices, decide whether a point is the closest lattice vector to t.

Besides that, the already mentioned work of Micciancio & Voulgaris [MV13] benefits from compact representations of Voronoi cells, which recently motivated their study, see e.g., [HRS20] by Hunkenschröder, Reuland, and Schymura. Furthermore, in his thesis [Hun20, §4.1] Hunkenschröder displayed that an optimization oracle for the Voronoi cell of a lattice is sufficient to obtain an algorithm for the closest vector problem that runs in expected polynomial time using results by Dadush & Bonifas [DB15].

Clearly, polynomial-size lifts for the Voronoi cell of a lattice yield an efficient implementation of an optimization oracle. The study of lifts of polytopes is a common tool in polyhedral combinatorics, and a very active research field over the last decade. For a polytope P, we write $\mathrm{xc}(P)$ for the minimum number of facets of any polytope that can be linearly projected onto P. This number is called the *extension complexity* of P. In Chapter 3, we provide bounds on the extension complexities of Voronoi cells of lattices. Moreover, we will also give a more detailed introduction to lattices and the concept of lifts.

We remark that the mere existence of polynomial-size lifts or equivalently descriptions using only polynomially many linear inequalities may not be immediately applicable, since finding such representations as well as verifying that they indeed yield the Voronoi cell of a given lattice might be hard. In fact, we initially considered the possibility that the rich structure of Voronoi cells of lattices results in such small lifts. Indeed, this is true for several examples as for the prominent class of *root lattices* and their duals. However, as a main result within Chapter 3, we show that Voronoi cells of lattices might be as complicated as stable-set polytopes, i.e., the convex hull of characteristic vectors of stable-sets in a graph. Matching the hardness of the stable-set problem, we know that there are families of graphs for which no lifts of the corresponding stable-set polytopes with only polynomially many facets exist [GJW18]. This yields the following theorem.

Theorem 1.3 (see Theorem 3.2)**.** *There exists a family of n-dimensional lattices Λ such that* $\operatorname{xc}(\mathrm{VC}(\Lambda)) = 2^{\Omega(n/\log n)}$.

This bound is very close to the trivial upper bound, since a Voronoi cell of a lattice can have up to $2(2^n - 1)$ many facets. In the proof of Theorem 1.3, we exploit the fact that the extension complexity of a face of a polytope P is always at most $\operatorname{xc}(P)$. Moreover, we use that $\operatorname{xc}(P) = \operatorname{xc}(P^\circ)$ holds for every polytope P with the origin in its interior, where P° is the dual polytope of P. With this in mind, we display a construction that yields a lattice for each $0/1$-polytope Q whose dual Voronoi cell has a face that projects onto Q. Very roughly speaking, we define the lattice as the set of integral solutions to a system of homogenized diophantine equations that arise from the constraints describing Q. Using this construction, a specific set of lattice vectors directly corresponds to the integer points in Q. These lattice vectors appear as outer normal vectors for $\mathrm{VC}(\Lambda)$ in one "region" of the Voronoi cell. In the dual $\mathrm{VC}(\Lambda)^\circ$, they correspond to vertices of a particular face. To prove Theorem 1.3, we are left with finding a $0/1$-polytope with high extension complexity that fits for our construction. In recent years, lower bounds on extension complexities have been established for various prominent polytopes, including many $0/1$-polytopes, such as cut polytopes [Fio+15; KW15; Cha+16], matching polytopes [Rot17], and certain stable-set polytopes [GJW18]. To the best of our knowledge, we believe that the bound on the stable-set polytope by Göös, Jain & Watson [GJW18] provides the best bound for Theorem 1.3 when using our construction. Furthermore, we even extended Theorem 1.3 for spectrahedral lifts and small approximations.

This Chapter 3 relies on a joint publication, and the results are also presented in:

M. Schymura, I. Seidel, and S. Weltge. "Lifts for Voronoi cells of lattices". In: *arXiv:2106.04432, to appear in Discrete & Computational Geometry 2023* (2021).

Densest subgraphs

As mentioned above, Chapter 4 deals with the underlying polyhedral structure of the densest subgraph problem. In general, knowing a linear programming formulation $\min\{c^\intercal x : Ax \leqslant b, x \in \mathbb{R}^n\}$ with underlying polyhedron $P = \{x \in \mathbb{R}^n : Ax \leqslant b\}$ opens a wide range of algorithms for solving the problem. Most prominent, though not guaranteed to run in polynomial-time, is Dantzig's simplex algorithm [Dan16].

Geometrically, this algorithm starts at a vertex of the feasible region P, and moves along edges of P towards vertices that admit a better objective value until it reaches an optimal vertex. Being aware of a concrete characterization of the edges and vertices of P can help to translate this general approach into a combinatorial algorithm.

Motivated by this, Hausmann and Korte characterized the edges of different polytopes associated with classical optimization problems such as the matching or independence polytope in [HK78]. One prime example of an algorithm traversing the edges of the corresponding polytope is Edmonds' Matching algorithm [Edm65a; Edm65b] following Berge's idea of using augmenting paths [Ber57]. In fact, two vertices of the matching polytope corresponding to matchings that only differ by an augmenting path are adjacent, see Chvátal [Chv75]. Moreover, the greedy algorithm for matroids [Rad57; Edm71] follows also the edges of the underlying polytope.

Within Chapter 4 we consider the *densest subgraph problem* defined as follows.

Problem 1.4 (see Problem 4.1)**.** *Given an undirected graph $G = (V, E)$, find a non-empty subgraph H of G that maximizes $\frac{|E(H)|}{|V(H)|}$.*

Analogously, we can phrase Problem 1.4 as finding a subgraph whose average degree is maximized, see Figure 1.4.

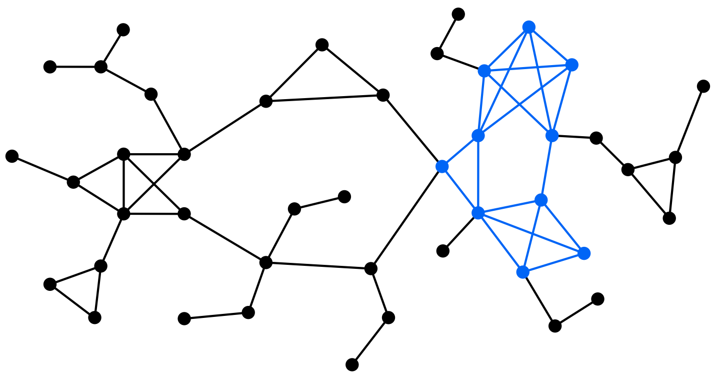


Figure 1.4: A graph G together with its densest subgraph H highlighted in blue, satisfying $|E(H)|/|V(H)| = 1.9$.

At first glance, this problem does not have a linear objective function. However, due to Charikar it admits a simple linear programming formulation [Cha00] providing solvability in polynomial time, which was known before, see [PQ82; Gol84]. In Charikar's work it is also shown that a simple greedy approach serves as a 2-approximating algorithm for Problem 1.4. Despite having exact algorithms, simple approximation algorithms seems to be very popular, see e.g., [Hoo+16], or [Boo+20]. This points out the importance of a very simple and intuitive algorithm, and indicates that the development of exact algorithms is not yet completed. To the best of our knowledge, so far none of the combinatorial algorithms directly use Charikar's linear programming formulation. Therefore, we started with exploring its polyhedral structure. We derived the following characterization of vertices and edges of the feasible region P_{dense} in Charikar's linear programming formulation.

Theorem 1.5 (see Theorem 4.2 & 4.3)**.** *Each vertex of P_{dense} directly corresponds to a non-empty connected subgraph H of G and vice versa. Moreover, two vertices of P_{dense} corresponding to different subgraphs H_1, H_2 share an edge if and only if*

- *H_1 and H_2 are disjoint, or*
- *one is a subgraph of the other, w.l.o.g. $H_1 \subset H_2$, and H_1 is obtained from H_2 either via deleting one edge, or one connected subgraph.*

After gaining this knowledge about the underlying polyhedral structure, we consider possible approaches for exploiting it in an algorithm. Obviously, all algorithms that follow the edges of the underlying polytope, can only be efficient if there exist short paths between any two vertices. The combinatorial diameter of a polytope measures the length of the greatest shortest path between two vertices. Therefore, the diameter serves as a lower bound on the number of steps a method that follows edges might take to obtain an optimal solution. In Chapter 4, we show that the combinatorial diameter of P_{dense} is bounded by 3 for every graph, which does not directly rule out the approach of following edges. However, we will also give evidence why it might be difficult to develop a simple algorithm that follows edges.

Therefore, instead of imitating the simplex procedure, we demonstrate another approach. We tailor a recently published algorithm to the densest subgraph problem. This algorithm by Dadush, Hojny, Huiberts, and Weltge [Dad+22] is designed for general convex optimization problems in the separation oracle model. Their oracle algorithm performs natural and simple update steps and is easy to implement, which makes it very applicable in practice, and therefore it fits at least our goal of having an algorithm that only performs simple updates.

This Chapter 4 resulted from conversations with my supervisor Stefan Weltge. Similar results to Theorem 1.5 discussed in Section 4.3.1 have been independently obtained by Milena Akemann during her master's thesis project [Ake22].

Preliminaries

Throughout this work, we assume the reader to be familiar with basic facts about linear and integer programming as well as polyhedra. For detailed background information, we refer to the book of Schrijver [Sch98]. Furthermore, two chapters deal with combinatorial problems based on graphs. Thus, the book of Korte & Vygen [KV11] might serve as an additional reference for standard graph terminology. Since, every chapter deals with different aspects of discrete optimization, we will introduce all problem specific concepts locally. Here, only the basic terminology and notational conventions used during the work are provided. Using a subscript, e.g., $\mathbb{R}_{\geqslant 0}$, we restrict the real numbers to be non-negative. We write $[n] := \{1, 2, \ldots, n\}$ for a positive integer n. The all-zeros vector in $\mathbb{R}^n$ is denoted by $\mathbf{0}_n$, the all-ones vector by $\mathbf{1}_n$, and we omit the subscript if the dimension is clear from the context. We use the standard Euclidean scalar product $\langle \cdot, \cdot \rangle$ as well as the associated Euclidean norm (ℓ_2-norm). Thus, $\| \cdot \|$ denotes the Euclidean norm, unless otherwise specified. For any $x \in \mathbb{R}^n$ the supremum norm $\|x\|_\infty$ denotes the greatest entry of x in absolute value.

Chapter 2

Minimum-cost integer circulations in given homology classes

2.1 Background

Finding optimal subgraphs of a surface-embedded graph that satisfy certain topological properties is a basic subject in topological graph theory and an important ingredient in many algorithms, see, e.g., [EN11, §1]. Motivated by recent work of Conforti, Fiorini, Joret, Huynh, and Weltge [Con+20a; Con+20b], we study a variant of the minimum-cost circulation problem with such an additional topological constraint. In [Con+20a; Con+20b], it was crucially exploited that the stable-set problem for graphs with bounded genus and bounded odd cycle packing number can be efficiently reduced to the below-stated Problem 2.1. While the standard minimum-cost circulation problem is among the most-studied problems in combinatorial optimization, much less seems to be known about this version.

Problem 2.1. *Given a directed graph D cellularly embedded in a surface together with non-negative costs c on its arcs and any integer circulation y in D, find a minimum-cost non-negative integer circulation in D that is $\mathbb{Z}$-homologous to y.*

Here, a circulation x is said to be *$\mathbb{Z}$-homologous to y* if their difference $x - y$ is a linear combination of facial circulations with integer coefficients, where a *facial circulation* is a circulation that sends one unit along the boundary of a single face, see Figure 2.1. If $x - y$ is a linear combination of facial circulations with real coefficients, we say that x is *$\mathbb{R}$-homologous* to y. As an example, if y is the all-zeros circulation, then y itself is clearly an optimal solution to Problem 2.1. However, for general y the all-zeros circulation might not be feasible. In fact, if the surface is different from the sphere and the projective plane, there are actually infinitely many homology classes, and their characterization is a basic subject in algebraic topology. We will provide more formal definitions in Section 2.2.

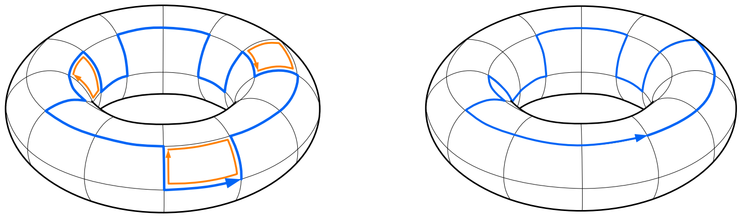


Figure 2.1: The circulations that send one unit along the blue directed cycles on the torus are $\mathbb{Z}$-homologous. In fact, their difference is the sum of three facial circulations which are depicted in orange.

In this chapter, we introduce this problem to the combinatorial optimization community, with a particular emphasis on the case in which the surface is *non-orientable*. While we complement existing results for orientable surfaces and show that the underlying polyhedra are actually easy to describe, only little seems to be known in the case of non-orientable surfaces. Our main result is a polynomial-time algorithm for non-orientable surfaces of fixed genus. Moreover, we show that the problem becomes NP-hard for general non-orientable surfaces.

For the case of *orientable* surfaces, Chambers, Erickson, and Nayyeri [CEN12] show that Problem 2.1 can be solved in polynomial time. Their approach is based on an exponential-size linear program that can be solved using the ellipsoid method in near-linear time, provided that the surface has small genus. Dey, Hirani, and Krishnamoorthy [DHK11] consider a variant of Problem 2.1 defined on simplicial complexes of arbitrary dimension, in which the (weighted) ℓ_1-norm of a chain homologous to y is to be minimized. For the case of an orientable surface, they derive a polynomial-time algorithm that is based on a linear program defined by a totally unimodular matrix.

We complement these results by showing that the convex hull of feasible solutions to Problem 2.1 has a very simple polyhedral description. To this end, notice that Problem 2.1 asks for minimizing a linear objective over the convex hull of all non-negative integer circulations in D that are $\mathbb{Z}$-homologous to y. We will denote this polyhedron by $P(D, y)$. Moreover, let $P(D)$ be the convex hull of non-negative integer circulations in D, which (as a network-flow polyhedron) has a simple linear description. Notice that any integer circulation x that is $\mathbb{Z}$-homologous to y must also be $\mathbb{R}$-homologous to y. In other words, x must be contained in the affine subspace of all circulations that are $\mathbb{R}$-homologous to y, which we denote by $L(D, y)$. Surprisingly, it turns out that it suffices to add the equations defining $L(D, y)$ to a description of $P(D)$ in order to obtain one for $P(D, y)$.

Theorem 2.2. *Let D be a directed graph that is cellularly embedded in an orientable surface and let y be any integer circulation in D. Then, $P(D, y) = P(D) \cap L(D, y)$.*

We will provide an explicit description for $L(D, y)$ later. Unfortunately, Theorem 2.2 does not hold for non-orientable surfaces. In fact, we show that Problem 2.1 becomes inherently more difficult on general non-orientable surfaces.

Theorem 2.3. *Problem 2.1 is strongly* NP*-hard on general non-orientable surfaces.*

While Dunfield & Hirani in [DH11] show that variants of Problem 2.1 become NP-hard on 3-dimensional simplicial complexes, their approach does not seem to apply to surfaces. In fact, the reduction therein crucially relies on 3-dimensional gadgets, and equivalent 2-dimensional configurations are not obvious to us. We obtain a reduction from general 3-SAT instances, showing that Problem 2.1 is indeed (strongly) NP-hard. To this end, we exploit ideas developed in [Con+20a] to reduce very particular instances of the stable-set problem to Problem 2.1.

On the positive side, we show that Problem 2.1 becomes tractable when dealing with non-orientable surfaces of *fixed* genus:

Theorem 2.4. *Problem 2.1 can be solved in polynomial time on non-orientable surfaces of fixed genus.*

A special case of Problem 2.1 was already treated and shown to be solvable in polynomial time in [Con+20a], where only instances arising from very specific stable-set problems were considered. Here, we consider the general problem. The algorithm in [Con+20a] is based on an alternative characterization of $\mathbb{Z}$-homology, which we have to replace by a more general one, see Theorem 2.5. In fact, in the instances considered in [Con+20a] the orientation of the arcs of D is already determined by an embedding scheme (defined in Section 2.2.1) of the dual graph, which we cannot assume here. Moreover, it is exploited that, in their setting, optimal circulations can be found in $\{0, 1\}^A$, which is also not the case for a general instance of Problem 2.1. Using a bound of Malnič and Mohar [MM92] on the number of certain non-freely-homotopic disjoint closed curves, it is then shown that an optimal circulation can be decomposed into few disjoint closed walks that can be enumerated efficiently. In this approach, the existence of optimal solutions with small entries is crucial to obtain a polynomial running time.

We propose another decomposition technique that enables us to reformulate Problem 2.1 as an integer program in standard form with a constant number of equality constraints, provided that the genus is fixed. Applying results on the proximity of integer programs, the resulting problem can be efficiently solved using dynamic programming.

Our approach does not rely on further topological ingredients, in particular we do not require bounds from [MM92].

Outline In Section 2.2, we provide a brief introduction to surfaces and graph embeddings, focusing on the facts that are necessary for the two subsequent sections. Our polynomial-time algorithm for Problem 2.1 on non-orientable surfaces with fixed genus is described in Section 2.3. This section relies on a characterization of $\mathbb{Z}$-homology that is provided later, so that the presented approach can be viewed without using information regarding homology. The proof of Theorem 2.3 is presented in Section 2.4. The introduction to graph embeddings continues in Section 2.5. Moreover, we present alternative characterizations of homology. More precisely, Section 2.5.1 is devoted to the case of orientable surfaces and contains a discussion of Theorem 2.2. The non-orientable case is treated in Section 2.5.2. Here, we provide a proof for a main ingredient (Theorem 2.5) of our algorithm. We close this chapter with a discussion of open problems in Section 2.6.

2.2 Preliminaries

We start with a *brief* introduction to surfaces, graph embeddings and the concept of homology. Further details and illustrations will be provided in Section 2.5.

2.2.1 Surfaces and embeddings

A *surface* is a non-empty connected compact Hausdorff topological space in which each point has an open neighborhood that is homeomorphic to the open unit disc in the plane. Examples of such surfaces are the sphere, the torus, and the projective plane. While the first two are *orientable* surfaces, the latter one is *non-orientable*. Up to homeomorphism, each surface $\mathbb{S}$ can be characterized by a single non-negative integer called the *Euler genus* g of $\mathbb{S}$ together with the information whether $\mathbb{S}$ is orientable. If $\mathbb{S}$ is orientable, then g is even and $\mathbb{S}$ can be obtained from the sphere by deleting $g/2$ pairs of open discs and, for each pair, identifying their boundaries in opposite directions ("gluing handles"). Otherwise, $\mathbb{S}$ is non-orientable and can be obtained from the sphere by deleting $g \geqslant 1$ open discs and, for each disc, identifying the antipodal points on its boundary ("gluing Möbius bands"), see Figure 2.2 for an illustration.

In the following, we consider (undirected and directed) graphs $G = (V, E)$ embedded in a surface with non-crossing edges. We require that every face of the embedding is homeomorphic to an open disc, which is called a *cellular* embedding.

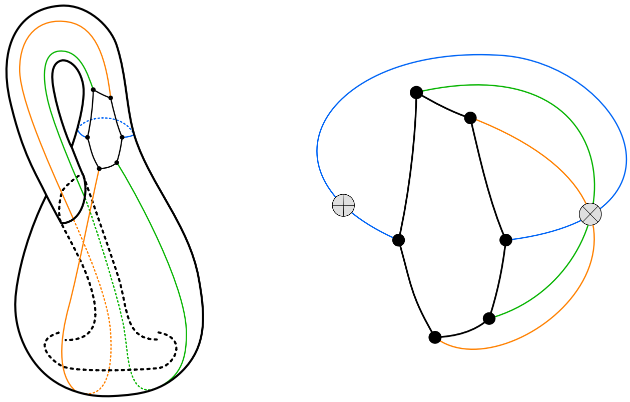


Figure 2.2: A graph embedded in the *Klein bottle*, the non-orientable surface of Euler genus 2. On the left, the surface is embedded in 3-dimensional space. Recall that the Klein bottle is obtained from the sphere by deleting two open discs and, for each disc, identifying the antipodal points on its boundary. On the right, an equivalent embedding of the same graph is shown, where these discs are depicted in gray.

Regardless of the (global) orientability of a surface, one can define a local orientation around each node v of G. If the surface is orientable, these local orientations can be chosen in a way that they are consistent along each edge. In non-orientable surfaces, this is not possible. To keep track of these inconsistencies, one can represent any cellular embedding by an *embedding scheme* $\Pi = (\pi, \lambda)$: The *rotation system* π describes, for all nodes, a cyclic permutation of the edges around a node induced by the local orientation. The *signature* $\lambda \in \{-1, +1\}^E$ indicates, for every edge, whether the two local orientations (clockwise vs. anti-clockwise) of the adjacent nodes agree $(+1)$ or not (-1), see Figure 2.4. We assume that an embedded graph is always given together with such an embedding scheme. Conversely, given *any* collection π of cyclic permutations of the edges incident to nodes and *any* vector $\lambda \in \{-1, +1\}^E$, there exists a cellular embedding for which $\Pi = (\pi, \lambda)$ is a corresponding embedding scheme.

For each face of the embedding of G, let us pick exactly one closed walk along the boundary of this face. In this way, we obtain a collection of closed walks which we call Π-*facial walks* and denote by F. A more formal definition of F is provided in Section 2.5. *Euler's Formula* states

$$|V| - |E| + |F| = 2 - g, \tag{2.1}$$

where g is the *Euler genus* of the surface.

Given a graph $G = (V, E)$ with embedding scheme $\Pi = (\pi, \lambda)$ and a set of Π-facial walks F, we define a *dual graph* $G^* = (V^*, E^*)$ as follows: Each node f^* of G^* corresponds to a Π-facial walk $f \in F$ of G and each edge $e^* \in E^*$ corresponds to an edge $e \in E$. If the edge e is part of two Π-facial walks f and g in G, the dual edge e^* is incident to f^* and g^*. As two Π-facial walks may share more than one edge and an edge might appear twice in the same Π-facial walk, the dual graph may have parallel edges and loops. If a graph is directed with arc set A, we define the dual graph to be the dual graph of the underlying undirected graph and an arc that corresponds to the dual edge e^* is called e.

We will equip the dual graph with a *dual embedding scheme* $\Pi^* = (\pi^*, \lambda^*)$: the traversing directions of the Π-facial walks in F directly correspond to the dual rotation system π^*, and the signature $\lambda^*(\cdot)$ of a dual edge is positive if the corresponding edge in G is used in opposite direction by the two corresponding Π-facial walks, and negative otherwise. This dual embedding scheme defines an embedding in the same surface. The collection F^* of Π^*-facial walks is chosen in a way that their walking directions correspond to the rotation system of G. An illustration of the dual embedding scheme and its relation to Π-facial walks is given in Figure 2.3.

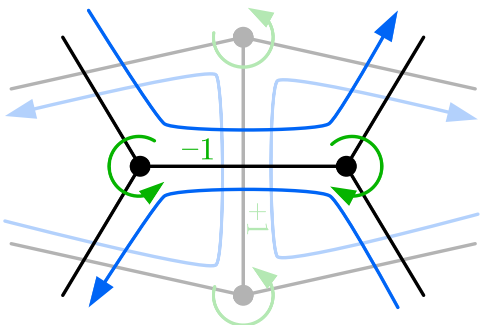


Figure 2.3: An extract of an embedded graph together with its dual graph (transparent). The embedding scheme and the associated dual embedding scheme are depicted in green. Corresponding facial walks are depicted in blue.

Let $D = (V, A)$ be a digraph with underlying undirected graph $G = (V, E)$ and dual graph G^*. For any walk $W = (v_1, e_1, v_2, e_2, \ldots, e_{\ell-1}, v_\ell)$ in G, we define the corresponding *characteristic flow* $\chi(W) \in \mathbb{Z}^A$ to be an assignment vector on the arcs of D indicating the total flow over the arcs when sending one unit of flow along W. This means that for $(v, w) \in A$, $\chi(W)((v, w))$ equals the number of appearance of the subsequence $(v, \{v, w\}, w)$ in W minus the number of appearance of $(w, \{v, w\}, v)$.

2.2.2 Homology

Given a directed graph $D = (V, A)$ cellularly embedded in $\mathbb{S}$, two integer circulations $x, y \in \mathbb{Z}^A$ are said to be *$\mathbb{Z}$-homologous* if $x - y = \sum_{f \in F} \eta_f \chi(f)$, where $\eta_f \in \mathbb{Z}$

for each facial walk $f \in F$. Let $\partial \in \mathbb{Z}^{A \times F}$ be the matrix whose columns are the vectors $\chi(f)$, $f \in F$. Problem 2.1 can then be reformulated as

$$\min\{c^\mathsf{T} x \ : \ x = y + \partial\eta,\ x \geqslant \mathbf{0},\ x \in \mathbb{Z}^A,\ \eta \in \mathbb{Z}^F\}. \tag{2.2}$$

For the case of an orientable surface, it is easy to see that the matrix ∂ is totally unimodular [DHK11], which implies that Problem 2.1 can be solved efficiently in this case. Unfortunately, ∂ is not totally unimodular whenever the surface is non-orientable.

2.3 A polynomial-time algorithm on non-orientable surfaces with fixed genus

While it is easy to obtain a polynomial-time algorithm for Problem 2.1 on orientable surfaces, much more work is required for non-orientable surfaces. In this section we describe an algorithm that runs in polynomial time for surfaces of *fixed* Euler genus. We will later see that the problem becomes NP-hard for general surfaces. A main ingredient of our algorithm is the following characterization of integer circulations that are $\mathbb{Z}$-homologous to a given one.

Theorem 2.5. *Given a digraph $D = (V, A)$ cellularly embedded in a non-orientable surface of Euler genus g, and an integer circulation y in D, there exist vectors $w_1, \dots, w_{g-1} \in \{0, \pm 1, \pm 2\}^A$, $h \in \{0, 1\}^A$ such that the following holds. An integer circulation $x \in \mathbb{Z}^A$ is $\mathbb{Z}$-homologous to y if and only if*

$$\langle w_i, x\rangle = \langle w_i, y\rangle \quad \textit{for all } i \in [g-1] \tag{2.3}$$

and

$$\langle h, x\rangle \equiv \langle h, y\rangle \pmod 2. \tag{2.4}$$

Moreover, $w_1, \dots, w_{g-1}, h$ can be computed in polynomial time.

The constraints in (2.3) describe the affine subspace of all x for which there exist $\eta_f \in \mathbb{R}$ for $f \in F$ such that $x - y = \sum_{f \in F} \eta_f \chi(f)$. The parity constraint in (2.4) then characterizes those x for which the coefficients η_f can be chosen to be integer. A proof of this characterization is given in Section 2.5.2.

From now, we fix a non-orientable surface of Euler genus g, and let $w_1, \dots, w_{g-1}, h$ be as in Theorem 2.5. Recall that in Problem 2.1 we are given costs $c \in \mathbb{R}^A_{\geqslant 0}$ and an integer circulation $y \in \mathbb{Z}^A$ in D, and we want to find a minimum-cost non-negative integer circulation that is $\mathbb{Z}$-homologous to y.

In what follows, we will exploit the basic fact that every non-negative circulation can be decomposed into circulations that correspond to directed cycles. To see whether a sum of such circulations is feasible for Problem 2.1, we make use of the following notation. Let $d \in \mathbb{Z}^{g-1}$ be the vector whose i-th entry is equal to $\langle w_i, y \rangle$. Set $e := \langle h, y \rangle \pmod 2 \in \{0, 1\}$. For each walk W in D, we consider the vector $q(W) \in \mathbb{Z}^{g-1}$ whose i-th entry is equal to $\langle w_i, \chi(W) \rangle$. Moreover, set $p(W) := \langle h, \chi(W) \rangle \pmod 2 \in \{0, 1\}$ and $B := 2|V|$.

A closed walk $W = v_1, a_1, v_2, \ldots, v_{k-1}, a_{k-1}, v_k$ in D is called a *B-walk* if $\|q(W_i)\|_\infty \leqslant B$ holds for all subwalks $W_i = v_1, a_1, v_2, \ldots, v_{i-1}, a_{i-1}, v_i$ of W. Notice that every directed cycle is a B-walk, and hence every non-negative integer circulation is the sum of circulations that correspond to B-walks. We consider the set

$$\Omega := \{(q(W), p(W)) \ : \ W \text{ is a } B\text{-walk in } D\}.$$

Lemma 2.6. *For each $(q, p) \in \Omega$, one can compute in polynomial time a B-walk $W =: W_{q,p}$ in D with $q(W) = q$ and $p(W) = p$ that minimizes $c^\intercal \chi(W)$.*

For the sake of exposition, we provide a proof at the end of this section. Notice that $|\Omega| \leqslant 2(2B + 1)^{g-1} = \operatorname{poly}(|V|)$, hence the collection $\{W_{q,p} : (q, p) \in \Omega\}$ can be computed in polynomial time. Let us now consider the following set

$$\mathcal{C} := \left\{ \sum_{(q,p)\in\Omega} z_{q,p}\chi(W_{q,p}) \ : z_{q,p} \in \mathbb{Z}_{\geqslant 0} \text{ for every } (q,p) \in \Omega, \right.$$
$$\left. \sum_{(q,p)\in\Omega} z_{q,p} q = d, \ \sum_{(q,p)\in\Omega} z_{q,p} p \equiv e \pmod 2 \right\}$$

of non-negative integer circulations in D.

Lemma 2.7. *Every circulation in $\mathcal{C}$ is feasible for Problem 2.1. Moreover, $\mathcal{C}$ contains at least one optimal solution.*

Again, we postpone the proof to the end of this section. Setting $\tilde{c}_{q,p} := c^\intercal \chi(W_{q,p})$ for each $(q, p) \in \Omega$, by Lemma 2.7 it remains to obtain a solution for

$$\min \left\{ \sum_{(q,p)\in\Omega} \tilde{c}_{q,p} z_{q,p} \ : \ z \in \mathbb{Z}^{\Omega}_{\geqslant 0}, \ \sum_{(q,p)\in\Omega} z_{q,p} q = d, \ \sum_{(q,p)\in\Omega} z_{q,p} p \equiv e \pmod 2 \right\}$$
$$= \min \left\{ \sum_{(q,p)\in\Omega} \tilde{c}_{q,p} z_{q,p} \ : \ z \in \mathbb{Z}^{\Omega}_{\geqslant 0}, \ k \in \mathbb{Z}_{\geqslant 0}, \ \sum_{(q,p)\in\Omega} z_{q,p} q = d, \ \sum_{(q,p)\in\Omega} z_{q,p} p = 2k + e \right\}.$$

Notice that the latter is an integer program in $n := |\Omega| + 1 = \operatorname{poly}(|V|)$ variables of the form

$$\min \left\{ \bar{c}^\intercal x \ : \ \bar{A}x = \bar{b}, \ x \in \mathbb{Z}^n_{\geqslant 0} \right\},$$

where $\bar{A} \in \mathbb{Z}^{g \times n}$ and $\bar{b} \in \mathbb{Z}^g$. Recall that the entries in $\bar{A}$ are polynomially bounded in n, and that g (the number of rows in $\bar{A}x = \bar{b}$) is assumed to be fixed. It is known that integer programs of this form can be solved in polynomial time. For instance, a polynomial-time algorithm for this setting is described in [Art+16], which is based on Papadimitriou's pseudopolynomial-time algorithm for integer programs with a fixed number of constraints [Pap81]. Another approach can be found in [EW19, Thm. 3.3]. This finishes the proof of Theorem 2.4. We close this section by providing the proofs for Lemma 2.6 and Lemma 2.7.

Proof of Lemma 2.6. We determine each $W_{q,p}$ by computing shortest paths in the following auxiliary graph $\bar{D} = (\bar{V}, \bar{A})$ defined by

$$\bar{V} := \left\{(v,(x,y)) \;:\; v \in V,\; x \in \{-B,\dots,B\}^{g-1}, y \in \mathbb{Z}_2\right\},$$

$$\bar{A} := \left\{ \big((v,(x,y)),(v',(x',y'))\big) \;:\; \begin{array}{l} (v,(x,y)),(v',(x',y')) \in \bar{V},\; (v,v') \in A, \\ x + M\chi((v,v')) = x', \\ y +\; \langle h, \chi((v,v'))\rangle \equiv y' \pmod 2 \end{array} \right\}.$$

Here, M is the matrix whose rows are the vectors $w_1^\intercal,\dots,w_{g-1}^\intercal$. Observe that for every walk W, we have $M\chi(W) = q(W)$. The cost $\bar{c}$ of an arc $\bar{a} = ((v,(x,y)),(v',(x',y')))$ in $\bar{A}$ is defined by $\bar{c}(\bar{a}) := c((v,v'))$. Notice that $\bar{D}$ can be constructed in polynomial time and that $\bar{c}$ is non-negative.

Let $(q,p) \in \Omega$ and fix a node $v \in V$. We observe that there is a bijection between B-walks W in D starting (and ending) at v with $q(W) = q$ and $p(W) = p$, and walks $\overline{W}$ in $\bar{D}$ from $(v,(\mathbf{0},0))$ to $(v,(q(W),p(W)))$. Moreover, the costs of W and $\overline{W}$ coincide. Indeed, let W be a B-walk in D with $q(W) = q$ and $p(W) = p$ that starts at v. Let $v_1,\dots,v_k,v_1$ be the sequence of nodes visited by W, and let W_i denote the respective subwalk from v_1 to v_i. Then, walk $\overline{W}$ in $\bar{D}$ is obtained by visiting the nodes

$$(v_1,(\mathbf{0},0)),\; (v_2,(q(W_2),p(W_2))),\dots,(v_k,(q(W_k),p(W_k))),\; (v_1,(q(W),p(W)))$$

in the given order, and the cost of W equals the cost of $\overline{W}$.

Conversely, consider any walk $\overline{W}$ from $(v,(\mathbf{0},0))$ to $(v,(q,p))$ in $\bar{D}$, and let $(v_i,(x_i,y_i))$, $i = 1,\dots,k$, be the sequence of nodes it visits. Define W to be the closed walk that visits the nodes $v_1,\dots,v_k$. We see that W is a B-walk with $q(W) = q$ and $p(W) = p$, and that the costs of $\overline{W}$ and W coincide.

We conclude that a B-walk W in D with $q(W) = q$ and $p(W) = p$ minimizing $c^\intercal\chi(W)$ can be found by computing a shortest path in $\bar{D}$ from $(v,(\mathbf{0},0))$ to $(v,(q(W),p(W)))$ for every $v \in V$, and returning the walk in D that corresponds to the path of minimum length. $\square$

Proof of Lemma 2.7. Again, let M be the matrix whose rows are the vectors $w_1^\intercal,\dots,w_{g-1}^\intercal$. Observe that for every walk W, we have $M\chi(W) = q(W)$. Moreover, by Theorem 2.5 a non-negative integer circulation x in D is feasible for Problem 2.1 if and only if $Mx = d$ and $\langle h, x\rangle \equiv e \pmod 2$.

Let $x = \sum_{(q,p)\in\Omega} z_{q,p}\chi(W_{q,p})$ be any circulation in $\mathcal{C}$. First, notice that each $\chi(W_{q,p})$ is a non-negative integer circulation in D, and so is x. Moreover, we have

$$Mx = \sum_{(q,p)\in\Omega} z_{q,p} M\chi(W_{q,p}) = \sum_{(q,p)\in\Omega} z_{q,p} q(W_{q,p}) = \sum_{(q,p)\in\Omega} z_{q,p} q = d$$

as well as

$$\langle h, x\rangle = \sum_{(q,p)\in\Omega} z_{q,p} \langle h, \chi(W_{q,p})\rangle \equiv \sum_{(q,p)\in\Omega} z_{q,p} p \equiv e \pmod 2.$$

Thus, x is feasible for Problem 2.1.

Now let x^* be an optimal solution to Problem 2.1. As discussed earlier, we may decompose x^* into B-walks $W_1, \ldots, W_k$ such that $x^* = \sum_{i=1}^{k} z_i\chi(W_i)$, where $z_1, \ldots, z_k \in \mathbb{Z}_{\geqslant 0}$. Clearly, we have that $(q_i, p_i) := (q(W_i), p(W_i)) \in \Omega$ for $i = 1, \ldots, k$. We consider the non-negative integer circulation

$$x' := \sum_{i=1}^{k} z_i\chi(W_{q_i,p_i}).$$

As x^* is feasible, we have $Mx^* = d$ and $\langle h, x^*\rangle \equiv e \pmod 2$, which yields

$$\sum_{i=1}^{k} z_i q_i = \sum_{i=1}^{k} z_i M\chi(W_i) = Mx^* = d$$

as well as

$$\sum_{i=1}^{k} z_i p_i = \sum_{i=1}^{k} z_i \langle h, \chi(W_i)\rangle = \langle h, x^*\rangle \equiv e \pmod 2.$$

This shows that $x' \in \mathcal{C}$. In particular, x' is also feasible for Problem 2.1. By definition of W_{q_i,p_i}, we have $c^\intercal\chi(W_{q_i,p_i}) \leqslant c^\intercal\chi(W_i)$ for $i = 1, \ldots, k$, which yields $c^\intercal x' \leqslant c^\intercal x^*$. Therefore, x' is also an optimal solution to Problem 2.1. $\square$

2.4 Hardness for instances on general non-orientable surfaces

In the previous section, we have shown that Problem 2.1 can be solved in polynomial time on non-orientable surfaces of fixed Euler genus. This problem becomes NP-hard on general non-orientable surfaces.

Let us consider the following problem, which is a special case of (the decision version of) Problem 2.1.

Problem 2.8. *Given a digraph $D = (V, A)$ cellularly embedded in a surface with arc costs $c \in \{0, \frac{1}{2}, 1\}^A$ such that $\mathbf{1} \in \mathbb{Z}^A$ is a circulation in D, and an integer k, decide whether there exists a non-negative integer circulation in D that is $\mathbb{Z}$-homologous to $\mathbf{1}$ and has cost at most k.*

In what follows, we will prove that Problem 2.8 is NP-hard, which implies Theorem 2.3. We will also see that the problem remains hard if we restrict ourselves to circulations in $\{0, 1\}^A$. In Section 2.4.3, we show that the following problem can be efficiently reduced to Problem 2.8.

Problem 2.9. *Given a connected graph $G = (V, E)$ together with edge costs $c \in \{0, \frac{1}{2}, 1\}^E$, and an integer k, decide whether there exists a vector $x \in \mathbb{Z}^V$ satisfying $x(v) + x(w) \leqslant 1$ for each $\{v, w\} \in E$ and $\sum_{\{v,w\} \in E} c(\{v, w\})(x(v) + x(w)) \geqslant k$.*

Problem 2.9 can be seen as a special stable-set problem where we neglect the non-negativity constraints. We will show that the following special case of the weighted stable-set problem can be efficiently reduced to Problem 2.9. A proof is given in Section 2.4.2. The node weights in Problem 2.10 and 2.9 are induced by edge costs, meaning that the weight of a node is just the sum of the costs of its incident edges.

Problem 2.10. *Given a graph $G = (V, E)$ with edge costs $c \in \{0, \frac{1}{2}, 1\}^E$ and an integer k, decide whether there exists a stable-set $S \subseteq V$ in G such that $\sum_{e \in E} c(e)|S \cap e| \geqslant k$.*

Finally, we show that Problem 2.10 is NP-hard by a reduction from 3-SAT in the next section. This concludes the proof of Theorem 2.3.

2.4.1 Hardness of Problem 2.10

The following reduction is based on the standard reduction for the classical stable-set problem, see Garey and Johnson [GJ79].

Let (U, C) be any instance of 3-SAT, where U is the set of variables and C denotes the set of clauses. Now, for each variable $u \in U$ we define the graph G^u consisting of two nodes representing u and its negation $\bar{u}$, which are joined by an edge e_u. The cost of this edge is set to $c(e_u) := 1$. Next, for each clause $c \in C$ we define a triangle graph G^c containing one node for each literal in c and three edges connecting them. We assign costs of $1/2$ to all edges in the triangle. Finally, we define $G = (V, E)$ as union of all G^u ($u \in U$) and G^c ($c \in C$) together with the following additional edges: For each literal ℓ that appears in a clause c and corresponds to variable u, connect the node in G^c that represents ℓ with the node in G^u that represents the negation of ℓ. The edge costs c for all these additional edges is defined to be zero.

Notice that every stable-set S in G satisfies $|S| = \sum_{e \in E} c(e)|S \cap e|$. We leave to the reader to check that (U, C) is satisfiable if and only if G has a stable-set S of cardinality $|S| \geqslant |U| + |C|$. A formal proof can be found in [GJ79].

2.4.2 Reduction from Problem 2.10 to Problem 2.9

We may assume G to be connected, otherwise we would treat each component separately. Since the problem becomes easy in the case of bipartite graphs, let G be non-bipartite.

It remains to show that G has a stable-set S with $\sum_{e \in E} c(e)|S \cap e| \geqslant k$ if and only if there exists a vector $x \in \mathbb{Z}^V$ satisfying $x(v) + x(w) \leqslant 1$ for each $\{v, w\} \in E$ and

$$\sum_{\{v,w\} \in E} c(\{v, w\})(x(v) + x(w)) \geqslant k. \tag{2.5}$$

If G has a stable-set S with $\sum_{e \in E} c(e)|S \cap e| \geqslant k$, we define $x \in \{0, 1\}^V$ to be the characteristic vector of S. For each edge $e = \{v, w\} \in E$, we clearly have $x(v) + x(w) \leqslant 1$, and $|S \cap e| = x(v) + x(w)$, which yields (2.5).

Conversely, suppose that

$$\max \left\{ \sum_{\{v,w\} \in E} c(\{v, w\})(x(v) + x(w)) \,:\, x \in \mathbb{Z}^V,\ x(v) + x(w) \leqslant 1 \text{ for } \{v, w\} \in E \right\} \geqslant k.$$

Since G is non-bipartite, it can be shown that the convex hull of feasible solutions to the above integer program is a pointed polyhedron. Moreover, it can be shown that each vertex of this polyhedron is a $0/1$-vector. Both facts and their proofs can be found in [Con+20a, Proposition 12]. Moreover, as c is non-negative, the above integer program is certainly bounded. This means that the optimum to the above integer program is attained at a point $x \in \{0, 1\}^V$, which yields the claim.

2.4.3 Reduction from Problem 2.9 to Problem 2.8

The following reduction is based on methods developed in [Con+20a] that are designed for graphs with a particular embedding. First, we have to construct such an embedding for the input graph G which will have the property that the edges of the dual graph G^* may be directed such that every facial walk in G^* is a directed walk.

The embedding is obtained by equipping G with a rotation system $\Pi = (\pi, \lambda)$ in which the signature of every edge is defined to be -1. Its cyclic permutations around each node can be chosen arbitrarily.

We may define the set F of Π-facial walks in G corresponding to this embedding. This can then be used to define the dual graph G^* together with a corresponding dual embedding scheme $\Pi^* = (\pi^*, \lambda^*)$, and a set F^* of exactly those Π^*-facial walks whose traversing directions correspond to the cyclic permutations π around the nodes in G. Since $\lambda = -\mathbf{1}$, all dual edges in G^* will always be used in the same direction in the Π^*-facial walks in F^*. To obtain a digraph $D = (V^*, A)$, we direct all edges in G^* corresponding to the direction in which the edges are used in the walks in F^*. Now, all Π^*-facial walks in F^* are directed walks. Observe that in D the vector $\mathbf{1} \in \mathbb{Z}^A$ is a circulation. Indeed, it is half times the sum over all characteristic flows of facial walks in F^*, i.e. $\mathbf{1} = \frac{1}{2}\sum_{v^* \in F^*} \chi(v^*)$. Since there is a one-to-one correspondence between E and A, the costs on the edges in G may also be seen as arc costs $c \in \{0, \frac{1}{2}, 1\}^A$. It remains to show that there exists a vector $x \in \mathbb{Z}^V$ satisfying $x(v) + x(w) \leqslant 1$ for each $\{v, w\} \in E$ and

$$\sum_{\{v,w\} \in E} c(\{v, w\})(x(v) + x(w)) \geqslant k, \tag{2.6}$$

if and only if there exists a non-negative integer circulation in D that is $\mathbb{Z}$-homologous to $\mathbf{1} \in \mathbb{Z}^A$ and has cost at most $\sum_{e \in E} c(e) - k$.

Suppose that there exists a vector $x \in \mathbb{Z}^V$ satisfying $x(v)+x(w) \leqslant 1$ for each $\{v, w\} \in E$ and Inequality (2.6). We consider the following integer circulation in D

$$y := \mathbf{1} - \sum_{v \in V} x(v)\chi(v^*)$$

which clearly is $\mathbb{Z}$-homologous to $\mathbf{1}$. Moreover, each arc a in D appears exactly twice in Π^*-facial walks in F^* and the nodes (or node) in G corresponding to these Π^*-facial walks are joined by an edge whose dual edge corresponds to a. Since $x(v) + x(w) \leqslant 1$ for each $\{v, w\} \in E$, the value of y assigned to the dual arc corresponding to $\{v, w\}$ is non-negative. Moreover, since each facial walk in G^* is a directed walk, we have

$$\begin{aligned} c^\intercal y = c^\intercal \left(\mathbf{1} - \sum_{v \in V} \chi(v^*)x(v)\right) &= \sum_{e \in E} c(e) - \sum_{v \in V}\left(\sum_{e \in \delta(v)} c(e)\right)x(v) \\ &= \sum_{e \in E} c(e) - \sum_{\{v,w\} \in E} c(\{v, w\})(x(v) + x(w)) \\ &\leqslant \sum_{e \in E} c(e) - k. \end{aligned}$$

Conversely, consider any non-negative integer circulation y in D that is $\mathbb{Z}$-homologous to the circulation $\mathbf{1}$ with cost at most $\sum_{e \in E} c(e) - k$. Since y is $\mathbb{Z}$-homologous to $\mathbf{1}$, there exist coefficients η_{v^*} for all $v^* \in \mathbb{Z}^{F^*}$ such that $y = \mathbf{1} - \sum_{v^* \in F^*} \eta_{v^*}\chi(v^*)$. As there is a one-to-one correspondence between nodes v in G and Π^*-facial walks v^* in F^*, we may define the vector $x \in \mathbb{Z}^V$ via $x(v) = \eta_{v^*}$ for all $v \in V$. Since y is non-negative, the sum of two coefficients η_{v^*} and η_{w^*} that correspond to facial walks using the same arc can never exceed one. Moreover, we have

$$\begin{aligned}\sum_{\{v,w\}\in E} c(\{v,w\})(x(v)+x(w)) &= \sum_{v\in V}\left(\sum_{e\in\delta(v)} c(e)\right)x(v) = c^{\mathsf{T}}\sum_{v\in V} x(v)\chi(v^*)\\ &= \sum_{e\in E} c(e) - c^{\mathsf{T}}\left(\mathbf{1} - \sum_{v\in V} x(v)\chi(v^*)\right)\\ &= \sum_{e\in E} c(e) - c^{\mathsf{T}} y \geqslant k,\end{aligned}$$

which concludes the proof.

2.5 Characterizing homology

In this section, we present alternative characterizations of homology, leading to a discussion of Theorem 2.2 and the proof of Theorem 2.5. To this end, we need to provide further details and definitions regarding surfaces and homology. For further information, we refer to the books of Hatcher [Hat05] and Mohar and Thomassen [MT01].

Consider a graph $G = (V, E)$ cellularly embedded in a surface with a corresponding embedding scheme $\Pi = (\pi, \lambda)$. In what follows, we provide a more formal definition of the set F of Π-facial walks.

Consider the following procedure which defines a Π-facial walk, see Figure 2.4: Start at an arbitrary node v and an edge e incident to v, then traverse e and continue the walk at the edge e' coming after, or before, e in the cyclic permutation given by π if the signature of e is positive, or negative, respectively. Reaching the next node, we continue again with the edge coming after, or before, the edge e' if the number of already traversed edges with negative signature is even, or odd, respectively. We continue until the following three conditions are met: (i) we reach the starting node v, (ii) the number of traversed edges with negative signature is even, (iii) the next edge would be the starting edge e. This way, we obtain a collection of closed walks which we then call Π-facial walks. Notice that the Π-facial walks of a digraph are walks in the underlying undirected graph. We consider two Π-facial walks to be equivalent if they only differ by a cyclic shift of nodes and edges or if one is the reverse of the other one. Let us pick one Π-facial walk from each equivalence class and denote the resulting set of walks by F. Notice that every edge is either contained twice in one walk in F or in exactly two walks in F.

By construction, the number of used edges with negative signature in a Π-facial walk is even. If a closed walk in G traverses an even number of edges with negative signature,

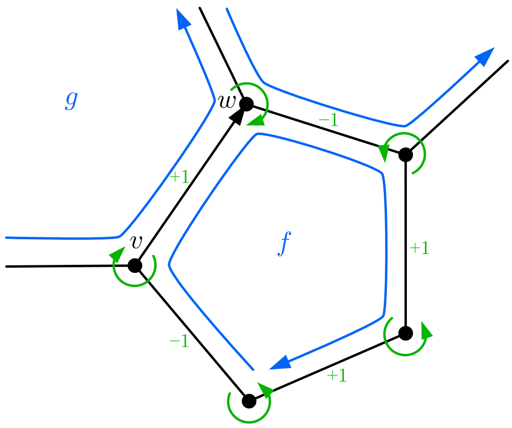


Figure 2.4: An extract of an embedded graph. The embedding scheme is depicted in green: Arrows around the nodes indicate their local orientations and the numbers on the edges the induced signature. The facial walks in F are drawn in blue.

it is called *two-sided*, otherwise it is called *one-sided*. It turns out that every two-sided cycle in the surface has a neighborhood that is homeomorphic to an annulus, whereas every one-sided cycle has a neighborhood that is homeomorphic to an open Möbius band. It follows that the presence of a one-sided cycle implies that the underlying surface is non-orientable.

To elaborate on an alternative characterization of homology, we also need the following notion. In addition to the notion of the characteristic flow $\chi(W) \in \mathbb{Z}^A$ for walks W in G, we define the vector $\xi(H^*) \in \mathbb{Z}^A$ for any walk H^* in the dual graph G^* as follows. Intuitively, we think of $\xi(H^*)$ as a flow that sends one unit along the edges in H^*. Whenever a unit is sent along a dual edge, we account it for the corresponding arc in D. The sign of this value will depend on the direction we traverse H^* along this arc. Formally, consider any arc $a = (v, w) \in A$ and let f be any Π-facial walk in F of G. Set $s(a, f) \in \{-1, 0, 1\}$ to be non-zero in the case that edge $\{v, w\}$ appears once f or twice in the same direction and zero otherwise. If f traverses the edge from v to w, then $s(a, f) = 1$, otherwise $s(a, f) = -1$. Observe that the sign of $s(a, f)$ equals the sign of $\partial_{a,f}$. For instance, in Figure 2.4, $s((v, w), f) = 1$. Now, for a walk $H^* = (f_1^*, e_1^*, f_2^*, \ldots, e_{\ell-1}^*, f_\ell^*)$ in G^* and arc $a \in A$, we define

$$\xi(H^*)(a) := \sum_{\substack{i \in \{1,\ldots,\ell-1\} \\ a = e_i}} \lambda^*(e_1^*) \cdot \cdots \cdot \lambda^*(e_{i-1}^*) s(a, f_i).$$

Observe that $\langle z, \xi(v^*) \rangle = 0$ for any circulation z in D and any Π^*-facial walk v^* in G^*.

Before we start with the characterization of $\mathbb{Z}$-homology, we consider the slightly weaker concept of $\mathbb{R}$-*homology*, which arises by dropping the integrality condition for

η_f in the definition of $\mathbb{Z}$-homology. More formally, given a directed graph $D = (V, A)$ cellularly embedded in $\mathbb{S}$, two circulations $x, y \in \mathbb{R}^A$ are called $\mathbb{R}$-homologous if $x - y$ is a linear combination of characteristic flows of Π-facial walks, which we also call *facial circulations*. That is, there exists an assignment vector $\eta \in \mathbb{R}^F$ with a coefficient $\eta_f \in \mathbb{R}$ for each facial walk $f \in F$, such that $x - y = \sum_{f \in F} \eta_f \chi(f)$. To rewrite the above in a compact way, recall the matrix $\partial = \partial_D \in \mathbb{Z}^{A \times F}$ defined by

$$\partial_{a,f} := \chi(f)(a) \quad \text{for all } a \in A,\ f \in F.$$

Circulations x, y are $\mathbb{R}$-homologous if $x = y + \partial\eta$ for some $\eta \in \mathbb{R}^F$.

Observe that two circulations x, y are $\mathbb{R}$-homologous ($\mathbb{Z}$-homologous) if and only if $x - y$ is $\mathbb{R}$-homologous ($\mathbb{Z}$-homologous) to the circulation $\mathbf{0} \in \mathbb{R}^A$. For this reason, in what follows we will first provide an alternative description of circulations that are $\mathbb{R}$-homologous ($\mathbb{Z}$-homologous) to $\mathbf{0} \in \mathbb{R}^A$, which then directly yields characterizations for $\mathbb{R}$-homology ($\mathbb{Z}$-homology) between two arbitrary circulations.

2.5.1 Orientable surfaces

For any digraph D, we denote the convex hull of non-negative integer circulations in D by $P(D)$. It is a basic fact that $P(D)$ is actually equal to the set of all non-negative circulations in D. Hence, this polyhedron can be described as the set of all $x \in \mathbb{R}^A_{\geqslant 0}$ that satisfy the "flow conservation" constraints. Regarding Problem 2.1, we are interested in the convex hull of only those integer circulations in $P(D)$ that are $\mathbb{Z}$-homologous to a given integer circulation y, and we denote the respective polyhedron by $P(D, y)$. The purpose of this section is to show that a description of $P(D, y)$ can be easily obtained in the orientable case.

As mentioned in Section 2.2.2, the matrix ∂ is totally unimodular in the case of orientable surfaces, and hence, Problem 2.1 can be solved in polynomial time. Here, we would like to elaborate on another consequence for the description of $P(D, y)$. By expressing $\mathbb{Z}$-homology using ∂, we know that

$$\begin{aligned} P(D, y) &= \operatorname{conv}\{x \in \mathbb{Z}^A \ : \ x = y + \partial\eta,\ x \geqslant \mathbf{0},\ \eta \in \mathbb{Z}^F\} \\ &= \operatorname{conv}\{x \in \mathbb{R}^A \ : \ x = y + \partial\eta,\ x \geqslant \mathbf{0},\ \eta \in \mathbb{R}^F\}, \end{aligned}$$

where the second equality follows from the integrality of the latter polyhedron, a consequence of ∂ being totally unimodular. This means that $P(D, y)$ is the set of all non-negative circulations in D that are $\mathbb{R}$-homologous to y. Denoting by $L(D, y)$ the set of all circulations in D that are $\mathbb{R}$-homologous to y, we obtain

$$P(D, y) = P(D) \cap L(D, y),$$

and hence Theorem 2.2. To obtain an even more explicit description of $P(D, y)$, observe that $L(D, y)$ is an affine subspace which is generated by all facial circulations and shifted by y. First, let us consider the case in which $y = \mathbf{0}$. The set $L(D, \mathbf{0})$ of all circulations in D that are $\mathbb{R}$-homologous to $\mathbf{0}$ is the subspace generated by all facial circulations. If the surface is orientable, the facial circulations generate a space of dimension $|F| - 1$. On the other hand, it is well-known that the space of all circulations in D is $(|A| - |V| + 1)$-dimensional. Thus, besides the constraints describing the set of all circulations, Euler's formula (2.1) yields that we need g additional constraints to obtain $L(D, \mathbf{0})$.

These constraints can be obtained by the following construction, also see [CEN12]. Pick any spanning tree K in G and observe that $G^* \backslash K^* := (V(G^*), E(G^*) \backslash \{e^* : e \in E(K)\})$ is still connected. Hence, there exists a spanning tree T^* in $G^* \backslash K^*$. By Euler's formula, there exist exactly g edges $e_1, \dots, e_g$ in G that are not contained in K and whose dual edges $e_1^*, \dots, e_g^*$ are not contained in T^*. For each $i \in [g]$, we define the cycle C_i as the unique (dual) cycle in $(V(T^*), E(T^*) \cup \{e_i^*\})$. These g cycles will yield the additional constraints needed to describe $L(D, \mathbf{0})$.

Proposition 2.11. *Let D be a digraph cellularly embedded in an orientable surface of Euler genus g, and let $C_1, \dots, C_g$ be the (dual) cycles defined above. Then,*

$$L(D, \mathbf{0}) = \left\{ x \in \mathbb{R}^A \ : \ x \text{ is a circulation and } \langle x, \xi(C_i) \rangle = 0 \ \forall i \in [g] \right\}.$$

Proof. Let L denote the linear subspace on the right-hand side.

We first show that $L(D, \mathbf{0}) \subseteq L$. Every $x \in L(D, \mathbf{0})$ is of the form $x = \sum_{f \in F} \eta_f \chi(f)$ for some coefficients $\eta_f \in \mathbb{R}$. Clearly, every cycle $H = (f_1^*, f_2^*, \dots, f_k^*)$ in the dual graph G^* is two-sided, and therefore, we have

$$\langle x, \xi(H) \rangle = \sum_{i=1}^{k} (\eta_{f_i} - \eta_{f_{i+1}}) = 0,$$

where $f_{k+1} = f_1$. This shows that $x \in L$, and hence $L(D, \mathbf{0}) \subseteq L$.

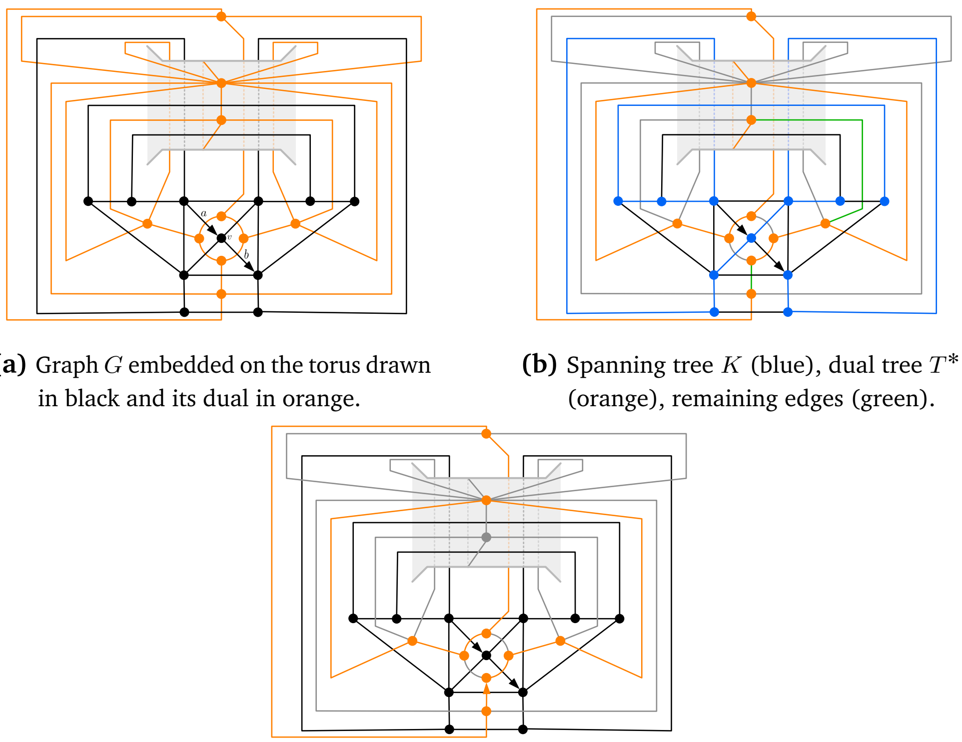
It remains to show that $\dim(L) \leqslant \dim(L(D, \mathbf{0}))$. Recall that $\dim(L(D, \mathbf{0})) = |F| - 1$ and that the space of all circulations has dimension $|F| - 1 + g$. With each constraint $\langle x, \xi(C_i) \rangle = 0$ that we iteratively add to the space of all circulations, the dimension drops by one. Indeed, for each $i \in [g]$ there is a (unique) cycle H_i in $K \cup \{e_i\}$. For this H_i, the circulation $\chi(H_i)$ satisfies all constraints $\langle \chi(H_i), \xi(C_j) \rangle = 0$ for $j \neq i$, but $\langle \chi(H_i), \xi(C_i) \rangle \neq 0$. This means that $\dim(L) \leqslant (|F| - 1 + g) - g = |F| - 1 = \dim(L(D, \mathbf{0}))$. $\square$

Corollary 2.12. *Let $D = (V, A)$ be a digraph cellularly embedded in an orientable surface of Euler genus g and let y be an integer circulation in D. Then, we can efficiently* *compute cycles $C_1, \ldots, C_g$ in the dual graph of D such that the following holds: An integer circulation $x \in \mathbb{Z}^A$ is $\mathbb{Z}$-homologous to y if and only if*

$$\langle x, \xi(C_i)\rangle = \langle y, \xi(C_i)\rangle \qquad \text{for all } i \in [g]$$

holds.

We note that the description of $P(D, y)$ following from Proposition 2.11 does not need to be totally unimodular. In Figure 2.5, we depicted a graph embedded on the torus, whose corresponding constraint matrix contains $\begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}$ as a submatrix.



(a) Graph G embedded on the torus drawn in black and its dual in orange.

(b) Spanning tree K (blue), dual tree T^* (orange), remaining edges (green).

(c) One dual cycle in orange.

Figure 2.5: In this figure, a graph G embedded on the torus is depicted. Thereby, the torus is illustrated via a plane extract of the sphere with attached handle illustrated as a bridge shaded in gray. For node v, the flow conservation constraint yields one $+1$ and one -1 for the arcs a and b. Figure 2.5b shows the construction of the trees used to define the dual cycles. For the cycle C in Figure 2.5c $\xi(C)(a) = \xi(C)(b) = 1$ holds when choosing a clockwise orientation around each node (-1 else). This shows the existence of a submatrix with determinant ± 2.

2.5.2 Non-orientable surfaces

In this section, we provide a characterization of $\mathbb{Z}$-homology that is exploited in Section 2.3. To this end, we first provide a characterization of $\mathbb{R}$-homology similar to Corollary 2.12 in the orientable case. In the previous section, we have seen that for an orientable surface two integer circulations are $\mathbb{Z}$-homologous if and only if they are $\mathbb{R}$-homologous. Unfortunately, this is not true for non-orientable surfaces. Therefore, some more arguments are required to obtain Theorem 2.5.

Let D be a digraph cellularly embedded in a non-orientable surface of Euler genus g according to an embedding scheme $\Pi = (\pi, \lambda)$. Let G^* be the dual graph canonically embedded in the same surface. In order to obtain a description as in Proposition 2.11, let us consider the following construction of $g-1$ closed walks in G^*. This construction is similar to the construction for embeddings in orientable surfaces described in Section 2.5.1.

Pick any spanning 1-tree T^* in G^* (a spanning tree with one additional edge forming one single cycle) whose cycle C^* is a one-sided cycle. Denote the set of arcs in D containing all arcs whose dual edges are in T^* by T. Notice that in most cases T does not form a tree. Since C^* is a one-sided cycle, $D\backslash T := (V(D), E(D)\backslash T)$ is still connected. Hence, there exists a spanning tree K in $D\backslash T$.

By Euler's formula (2.1), there exist exactly $g-1$ arcs $b_1, \dots b_{g-1}$ in D that are not contained in $E(K) \cup T$. For each $i \in [g-1]$, we define W_i^* as the (dual) two-sided closed walk in $T^* \cup \{b_i^*\} := (V(T^*), E(T^*) \cup \{b_i^*\})$: In case that $T^* \cup \{b_i^*\}$ contains a two-sided cycle, W_i^* equals this cycle. Otherwise, $T^* \cup \{b_i^*\}$ contains two one-sided cycles, namely C^* and a cycle containing b_i^*. In this case W_i^* walks once along C^*, along a path in T^* towards the one-sided cycle containing b_i^*, along this cycle, and finally back to C^* on the same path. For the remainder of this section, we keep $T, T^*, K, C, C^*, b_1, \dots, b_{g-1}$ and $W_1^*, \dots, W_{g-1}^*$ fixed.

Notice that we defined the above-described walks in such a way that each walk uses an edge at most twice. These $g-1$ closed walks will yield the constraints needed to describe $\mathbb{R}$-homology.

Lemma 2.13. *Let $z \in \mathbb{R}^A$ be a circulation in D and let $\eta \in \mathbb{R}^F$ be an assignment on the Π-facial walks such that $z(a) = \partial\eta(a)$ holds for all arcs $a \in T \cup \{b_1, \dots b_{g-1}\}$. Then z is $\mathbb{R}$-homologous to the all-zeros circulation.*

Proof. As $z(a) = \partial\eta(a)$ holds for all arcs $a \in T \cup \{b_1, \dots b_{g-1}\}$, $z - \partial\eta$ is a circulation in D that is zero on all arcs that are not contained in the spanning tree K. Since K does not contain any cycle, the circulation $z - \partial\eta$ must be zero on all arcs in K as well. Therefore, $z = \partial\eta$, which yields the claim. □

In what follows, we will identify linear equations that ensure the existence of a vector $\eta \in \mathbb{R}^F$ such that $z(a) = \partial\eta(a)$ for all $a \in T \cup \{b_1, \dots, b_{g-1}\}$. To this end, the following properties of the matrix ∂ will be useful. Recall that the rows in ∂ correspond to the arcs in A, and the columns in ∂ correspond to the walks in F. We have $\partial_{a,f} \neq 0$ if walk f uses the underlying undirected edge corresponding to a once, or twice in the same direction. Since there are one-to-one correspondences between arcs A and dual edges A^*, and between walks F and dual nodes V^*, respectively, ∂ may be interpreted as a matrix in $\mathbb{Z}^{A^* \times V^*}$. With this interpretation we have $\partial_{a^*,f^*} \neq 0$ if edge a^* is no loop and incident to f^* in the dual graph G^*, or if a^* is a loop at f^* with negative dual signature.

Lemma 2.14. *For a submatrix ∂_C of ∂ consisting only of the rows and columns that correspond to nodes and edges used in C^*, respectively, we have $|\det(\partial_C)| = 2$. Moreover, the absolute value of the determinant of any submatrix of ∂_C, obtained by deleting exactly one row and one column, equals one.*

Proof. In case that C^* is just a loop consisting of the edge e^*, the signature of e^* needs to be negative, since C^* is one-sided. Therefore, the corresponding arc in D is used twice in the same direction in the corresponding Π-facial walk. Therefore, $\partial_C = (\pm 2)$.

In case that C^* is not a loop, all entries in ∂ are either ± 1 or zero. Moreover, for any arc a^* and node f^* in C^*, the entry $\partial_{a^*,f^*} \neq 0$ if edge a^* is incident to f^* in the dual graph G^*. Let $C^* = (f_1^*, e_1^*, f_2^*, \dots, e_{\ell-1}^*, f_\ell^*, e_\ell^*, f_1^*)$ with $\lambda^*(e_1^*) \cdot \dots \cdot \lambda^*(e_\ell^*) = -1$. Notice that in each row and column in ∂_C, there are exactly two non-zero coefficients, each of them equals either $+1$ or -1. In a row that corresponds to an arc a, the two non-zero coefficients have the same sign if and only if the walks in F use the underlying edge of a in the same direction. This is the case if and only if the dual signature of the dual edge of a is negative. Since C^* is one-sided, the number of rows in ∂_C in which the two non-zero entries have the same sign is odd. Notice that resorting rows/columns or multiplying rows/columns by -1 does not change the absolute value of ∂_C's determinant. Moreover, multiplying rows/columns by -1 does not change the parity of the number of rows in which the two non-zero entries have the same sign. Hence, the absolute value of ∂_C's determinant may be calculated as follows:

$$|\det(\partial_C)| = \left|\det\begin{pmatrix} 1 & 1 & & & & \\ & 1 & 1 & & & \\ & & 1 & 1 & & \\ & & & \ddots & \ddots & \\ & & & & 1 & 1 \\ a & & & & & b \end{pmatrix}\right| = |b + (-1)^{\ell+1} a| = 2,$$

where $a, b \in \{-1, +1\}$. Here, the second equality follows from Laplace's formula, and the last equality is due to $ab = (-1)^{\ell+1}$. This holds true because C^* is one-sided, and hence, the displayed matrix should contain an odd number of rows, in which the two non-zero entries have the same sign.

The second assertion regarding further submatrices of ∂_C follows by a similar argumentation, ending up with a triangular matrix. $\square$

Lemma 2.15. *Given a circulation $z \in \mathbb{R}^A$ in D, there is a unique $\eta \in \mathbb{R}^F$ such that $z(a) = \partial\eta(a)$ holds for all arcs $a \in T$.*

Proof. For any $\eta \in \mathbb{R}^F$, let η_C denote the restriction of η to the Π-facial walks that correspond to nodes of C^*. Moreover, let z_C denote the restriction of z to the arcs in C. By Lemma 2.14, the determinant of ∂_C equals ± 2. Hence, ∂_C is regular and η_C is uniquely determined by the values of z_C via the equation $z_C = \partial_C \eta_C$. The remaining values of η are uniquely determined by extending η_C along the arcs in T to $\eta \in \mathbb{R}^F$ such that $z(a) = \partial\eta(a) = s(a,h)\eta_h + s(a,g)\eta_g$ for every arc $a \in T$ whose underlying edge is used in the two walks h, $g \in F$. $\square$

We are now ready to characterize circulations that are $\mathbb{R}$-homologous to $\mathbf{0}$. Intuitively, a circulation is a linear combination of facial circulations if and only if the amount of flow that crosses a two-sided closed walk W_i^* of the dual graph sums up to zero for $i \in [g-1]$, see Figure 2.6a.

Lemma 2.16. *A circulation $z \in \mathbb{R}^A$ is $\mathbb{R}$-homologous to* $\mathbf{0}$ *if and only if the following $g-1$ linear equations hold:*

$$\langle z, \xi(W_i^*)\rangle = 0 \qquad \text{for all } i \in [g-1].$$

Proof. By Lemma 2.15, let $\eta \in \mathbb{R}^F$ be the unique assignment vector, which satisfies $z(a) = \partial\eta(a)$ for all arcs $a \in T$. Lemma 2.13 states that the circulation z is $\mathbb{R}$-homologous to $\mathbf{0}$ if and only if $z(b_i) = \partial\eta(b_i)$ holds for all $b_i \in D \backslash (K \cup T)$. We show that for any $i \in [g-1]$ the equation $\langle z, \xi(W_i^*)\rangle = 0$ is equivalent to $z(b_i) = \partial\eta(b_i)$.

We think of $\langle z, \xi(W_i^*)\rangle$ as walking along W_i^* and whenever crossing an arc a of D adding the value $z(a)$ with the appropriate sign. Since W_i^* is two-sided, the appropriate sign may be interpreted as an indicator for the direction $z(a)$ crosses W_i^*. The crucial property (besides W_i^* being two-sided) leading to the claimed equivalence is that all arcs corresponding to W_i^* except b_i belong to T. Therefore, z restricted to these arcs can be written as the combination of facial circulations with coefficients η. Hence, at these positions, the values added when entering or leaving a face in D while walking along W_i^* will cancel out. As a consequence, $\langle z, \xi(W_i^*)\rangle = 0$ holds if and only if the value added (with the appropriate sign, since W_i^* is two-sided) at b_i also cancels out. This happens if and only if $z(b_i) = \partial\eta(b_i)$.

Formally, consider the following two relationships:

$$\partial\eta(a) = s(a,f)\eta_f + s(a,g)\eta_g, \text{ and} \tag{2.7}$$

$$s(a,f)s(a,g) = -\lambda^*(a^*), \tag{2.8}$$

where the underlying edge of a appears in the Π-facial walks f and g and the dual edge of a is denoted by a^*. Since a cyclic shift does not change our considerations, we assume W_i^* to be the following walk in the dual graph: $(f_1^*, e_1^*, f_2^*, \ldots, e_{\ell-1}^*, f_\ell^*, e_\ell^* = b_i^*, f_1^*)$.

We denote the corresponding arcs in D by $e_1, \ldots, e_{\ell-1}, e_\ell = b_i$. By the definition of $\xi(\cdot)$, we have

$$\begin{aligned}\langle z, \xi(W_i^*)\rangle &= \sum_{a\in A}\left(\sum_{\substack{j\in\{1,\ldots,\ell\}\\ a=e_j}} \lambda^*(e_1^*)\cdots\lambda^*(e_{j-1}^*)s(a,f_j)\right)\cdot z(a)\\ &= \sum_{j=1}^{\ell}\lambda^*(e_1^*)\cdots\lambda^*(e_{j-1}^*)s(e_j,f_j)\cdot z(e_j).\end{aligned}$$

All arcs in W_i except b_i lie in T. For those arcs, we assume that $z(e_j) = \partial\eta(e_j)$. Hence,

$$\begin{aligned}\langle z, \xi(W_i^*)\rangle &= \sum_{j=1}^{\ell-1}\lambda^*(e_1^*)\cdots\lambda^*(e_{j-1}^*)s(e_j,f_j)\cdot\partial\eta(e_j)\\ &\qquad + \lambda^*(e_1^*)\cdots\lambda^*(e_{\ell-1}^*)s(e_\ell,f_\ell)\cdot z(b_i)\\ &= \sum_{j=1}^{\ell-1}\lambda^*(e_1^*)\cdots\lambda^*(e_{j-1}^*)s(e_j,f_j)\cdot\big(\eta_{f_j}s(e_j,f_j)+\eta_{f_{j+1}}s(e_j,f_{j+1})\big)\\ &\qquad + \lambda^*(e_1^*)\cdots\lambda^*(e_{\ell-1}^*)s(e_\ell,f_\ell)\cdot z(b_i)\end{aligned}$$

holds using Equation (2.7). Now, by Equation (2.8), we have

$$\begin{aligned}\langle z, \xi(W_i^*)\rangle &= \sum_{j=1}^{\ell-1}\lambda^*(e_1^*)\cdots\lambda^*(e_{j-1}^*)\cdot\big(\eta_{f_j}-\lambda^*(e_j^*)\eta_{f_{j+1}}\big)\\ &\qquad + \lambda^*(e_1^*)\cdots\lambda^*(e_{\ell-1}^*)s(e_\ell,f_\ell)\cdot z(b_i)\\ &= \eta_{f_1}-\lambda^*(e_1^*)\cdots\lambda^*(e_{\ell-1}^*)\eta_{f_\ell}\\ &\qquad + \lambda^*(e_1^*)\cdots\lambda^*(e_{\ell-1}^*)s(e_\ell,f_\ell)\cdot z(b_i).\end{aligned}$$

Therefore, $\langle z, \xi(W_i^*)\rangle = 0$ holds if and only if

$$\begin{aligned}z(b_i) &= -\frac{\eta_{f_1}-\lambda^*(e_1^*)\cdots\lambda^*(e_{\ell-1}^*)\eta_{f_\ell}}{\lambda^*(e_1^*)\cdots\lambda^*(e_{\ell-1}^*)s(e_\ell,f_\ell)}\\ &= -\big(\eta_{f_1}-\lambda^*(e_1^*)\cdots\lambda^*(e_{\ell-1}^*)\eta_{f_\ell}\big)\cdot\big(\lambda^*(e_1^*)\cdots\lambda^*(e_{\ell-1}^*)s(e_\ell,f_\ell)\big)\end{aligned}$$

because the denominator is ± 1. Hence,

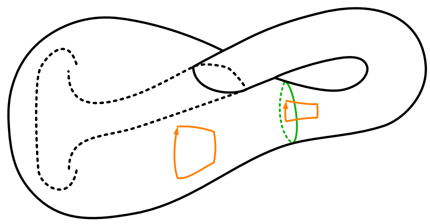
$$z(b_i) = \eta_{f_\ell} s(e_\ell, f_\ell) - \lambda^*(e_1^*) \cdots \lambda^*(e_{\ell-1}^*) \eta_{f_1} s(e_\ell, f_\ell).$$

Using Equation (2.8) and that W_i^* is two-sided, we obtain,

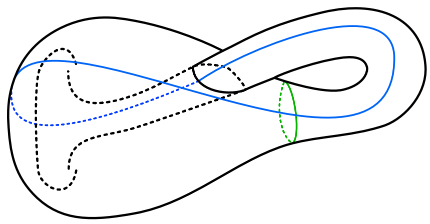
$$\begin{aligned} z(b_i) &= \eta_{f_\ell} s(e_\ell, f_\ell) + \lambda^*(e_1^*) \cdots \lambda^*(e_\ell^*) \eta_{f_1} s(e_\ell, f_1) \\ &= \eta_{f_\ell} s(e_\ell, f_\ell) + \eta_{f_1} s(e_\ell, f_1) \\ &= \partial\eta(b_i), \end{aligned}$$

which concludes the proof. □

Notice that even if the given circulation z in D is integer, the vector $\eta \in \mathbb{R}^F$ defined in Lemma 2.15 is not necessarily integer. The following lemma yields a characterization when η can be chosen to be integer. Intuitively, the coefficients η are integer if and only if the amount of flow crossing a one-sided cycle in the dual graph is even, see Figure 2.6b for an illustration of a one-sided cycle.



(a) Two-sided cycle (green) and facial circulations (orange) on the Klein bottle.



(b) Two-sided cycle (green) and one-sided cycle (blue) on the Klein bottle.

Figure 2.6: The figures illustrate the Klein bottle. The green cycle on the surface is two-sided, whereas the blue cycle is one-sided. The directed orange cycles illustrate facial circulations of a graph embedded on the Klein bottle. The flow sent across the two-sided cycle by facial circulations sums up to zero.

Lemma 2.17. *Given a circulation $z \in \mathbb{Z}^A$ in D and $\eta \in \mathbb{R}^F$ such that $z(a) = \partial\eta(a)$ for all arcs $a \in T$, η is integer if and only if $\sum_{a\in C} z(a) \equiv 0 \pmod 2$ holds.*

Proof. Observe that integrality extends along the arcs in T via the relation $z(a) = \partial\eta(a) = s(a,h)\eta_h + s(a,g)\eta_g$ for every arc $a \in T$ whose underlying edge is used in the two walks $h, g \in F$. Exploiting this fact, we fix one Π-facial walk $f \in F$ that corresponds to a node in the dual cycle C^*. Hence, it suffices to prove that $\eta_f \in \mathbb{Z}$ is equivalent to $\sum_{a\in C} z(a)$ being even. Using the same notation as in the proof of Lemma 2.15, value η_f is uniquely defined by $z_C = \partial_C \eta_C$. By Cramer's rule, $\eta_f = \det(\partial_{C_f}) / \det(\partial_C)$, where ∂_{C_f}

denotes the matrix obtained by replacing the column in ∂_C corresponding to f by z_C. By Lemma 2.14, $|\det(\partial_C)| = 2$ and deleting one row and one column in ∂_C results in a matrix that has determinant ± 1. Therefore, by Laplace's rule applied to the column that equals z_C, we conclude that $\det(\partial_{C_f}) = \sum_{a\in C} \pm 1 \cdot z(a)$. Since the signs in the summation do not affect the parity, $\det(\partial_{C_f})$ is even if and only if $\sum_{a\in C} z(a)$ is even. It follows that $\sum_{a\in C} z(a)$ is even if and only if η_f is integer. $\square$

Lemma 2.15, Lemma 2.16, and Lemma 2.17 yield the following characterization of $\mathbb{Z}$-homology.

Proposition 2.18. *Let $D = (V, A)$ be a digraph cellularly embedded in an orientable surface of Euler genus g and y an integer circulation in D. We can efficiently compute a one-sided cycle C and two-sided closed walks $W_1^*, \dots, W_{g-1}^*$ in the dual graph of D (which do not use an edge more than twice) such that the following holds: An integer circulation $x \in \mathbb{Z}^A$ is $\mathbb{Z}$-homologous to y if and only if*

$$\langle x, \xi(W_i^*)\rangle = \langle y, \xi(W_i^*)\rangle \qquad \text{for all } i \in [g-1],$$

and

$$\sum_{a\in C} x(a) \equiv \sum_{a\in C} y(a) \pmod 2.$$

Observe that Theorem 2.5 is a direct consequence of this Proposition 2.18.

2.6 Open questions

When we considered Problem 2.1 in 2021, we were wondering whether the assumptions, namely c being non-negative and not allowing for arbitrary capacity bounds on x can be dropped while still having a polynomial-time algorithm for fixed genus. For orientable surfaces, adapting the formulation in (2.2) results again in a totally unimodular system, and the discussion in Section 2.5.1 directly extends to this more general setting. However, our algorithm for non-orientable surfaces in Section 2.3 does neither cover arbitrary costs nor general bounds on x. Nevertheless, the generalized problem can be efficiently solved on surfaces of fixed genus. We remark that using the formulation (2.2) results in the following integer program

$$\min\{c^\intercal x \;:\; x = y + \partial\eta,\, \ell \leqslant x \leqslant u,\, x \in \mathbb{Z}^A,\, \eta \in \mathbb{Z}^F\},$$

where $\ell, u \in \mathbb{Z}^A$ denote lower and upper capacity bounds. This formulation is equivalent to

$$\min\{(\partial^\intercal c)^\intercal\, \eta \;:\; \partial\eta \leqslant u - y,\, \partial\eta \geqslant \ell - y,\, \eta \in \mathbb{Z}^F\},$$

Using the fact that the Euler genus of a non-orientable surface is equal to the largest number of pairwise disjoint one-sided simple closed curves and Lemma 2.14, one can show that ∂ is totally 2^g-modular. As addressed in Chapter 1, determining the complexity status of integer programs with totally Δ-modular constraint matrix is an open problem for fixed Δ. However, a strongly polynomial-time algorithm for integer programs with totally Δ-modular coefficient matrices that contain at most two non-zero entries in each row (or column) for every fixed constant Δ is given in [Fio+22]. As every arc appears in at most two facial walks, ∂ actually has at most two non-zeros in each row and hence these integer programs are solvable in polynomial time for fixed genus.

Another natural generalization of this problem arises by considering b-flows instead of just circulations. Again, a formulation according to (2.2) results in an integer program that is solvable in polynomial time for fixed genus. In the case of a constant number of nodes with non-zero demand or supply, we can even adopt our algorithm in Section 2.3 to the setting of b-flows.

Most questions answered by [Fio+22], we are still left with the following. Note that while the running time of our algorithm is polynomial for fixed Euler genus g, the degree of the polynomial depends on g (see, e.g., the cardinality of Ω in Section 2.3). We do not know whether Problem 2.1 is fixed-parameter tractable in g.

Question 2.19. *Is Problem 2.1 fixed-parameter tractable in the genus g?*

Chapter 3

Lifts for Voronoi cells of lattices

3.1 Background

Many polytopes that arise in the study of polyhedral combinatorics are linear projections of higher-dimensional polytopes, also called *lifts*, with significantly fewer facets. Prominent examples include basic polytopes such as permutahedra [Goe15], cyclic polytopes [Bog+15], and polygons [Shi14], as well as several polytopes associated to combinatorial optimization problems such as spanning tree polytopes [Mar91; Won80], subtour-elimination polytopes [Yan91], stable-set polytopes of certain families of graphs [FOS12; PS93; Con+20b], matching polytopes of bounded-genus graphs [Ger91], independence polytopes of regular matroids [AF22], or cut dominants [CCZ13].

In this chapter, we study to which extent this phenomenon also applies to Voronoi cells of lattices. Here, a *lattice* is the image of $\mathbb{Z}^k$ under an injective linear map. We say that a lattice is d-dimensional, if d is the dimension of its linear hull. The *Voronoi cell* $\mathrm{VC}(\Lambda)$ of a lattice $\Lambda \subseteq \mathbb{R}^k$ is the set of all points in $\mathrm{lin}(\Lambda)$ for which the origin is among the closest lattice points, i.e.,

$$\mathrm{VC}(\Lambda) := \{x \in \mathrm{lin}(\Lambda) : \|x\| \leqslant \|x - z\| \text{ for all } z \in \Lambda\},$$

The lattice translates $z + \mathrm{VC}(\Lambda)$ for $z \in \Lambda$, induce a facet-to-facet tiling of $\mathrm{lin}(\Lambda)$, so that in particular Voronoi cells of lattices are what is commonly called *space tiles*, see Figure 3.1. Moreover, it is known that $\mathrm{VC}(\Lambda) \subseteq \mathbb{R}^k$ is a centrally symmetric polytope with up to $2(2^k - 1)$ facets. We refer to [Gru07, Ch. 32] for background on translative tilings of space.

It is tempting to believe that the rich structure of Voronoi cells of lattices allows constructing polytopes that linearly project onto $\mathrm{VC}(\Lambda)$ having significantly fewer than $2(2^k - 1)$ facets. In fact, this is true for several examples: A lattice whose Voronoi cell has the largest possible number of facets is the d-dimensional dual root lattice $A_d^\star$, see Section 3.3.1 for a definition. However, its Voronoi cell is a permutahedron and admits a

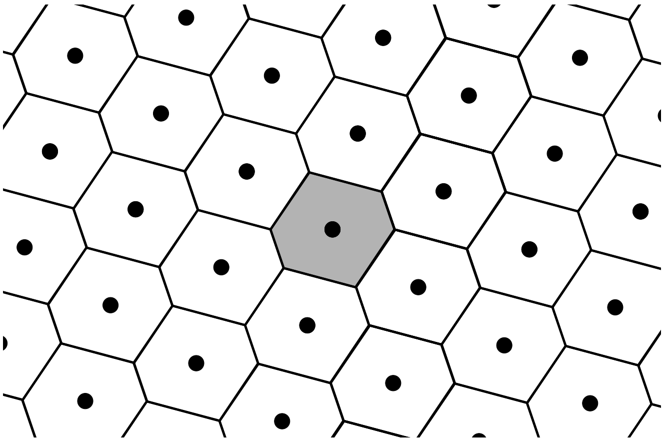


Figure 3.1: A lattice in $\mathbb{R}^2$ together with its Voronoi cell and the corresponding tiling of the plane via its lattice translates.

lift with only $\mathcal{O}(d \log d)$ facets [Goe15], see Section 3.3.1. More generally, if the Voronoi cell of a d-dimensional lattice is a zonotope, then it has $\mathcal{O}(d^2)$ generators and hence has a lift with $\mathcal{O}(d^2)$ facets. We discuss this result in detail in Section 3.3.2.

The lattice $A_d^\star$ also belongs to the prominent class of *root lattices* and their duals. By their algebraic and geometric properties, these lattices are prime examples in various contexts: For example, they play a crucial role in Coxeter's classification of reflection groups, [CS99, Ch. 4], and they yield the densest sphere packings and thinnest sphere coverings in small dimensions, see [CS99] or [Sch09].

As one part of this chapter, we show that Voronoi cells of such lattices generally admit small lifts. In what follows, for a polytope P we write $\mathrm{xc}(P)$ for the minimum number of facets of any polytope that can be linearly projected onto P. This number is called the *extension complexity* of P.

Theorem 3.1. *For every d-dimensional lattice Λ that is a root lattice or the dual of a root lattice, we have $\mathrm{xc}(\mathrm{VC}(\Lambda)) = \mathcal{O}(d \log d)$.*

This raises the question whether Voronoi cells of other lattices also have a small extension complexity, say, polynomial in their dimension. One of the main motivations for representing a polytope P as the projection of another polytope Q is that a linear optimization problem over P can be reduced to one over Q. If Q has a small number of facets, then the latter task can be expressed as a linear program with a small number of inequalities, also known as an *extended formulation*.

Thus, given a lattice $\Lambda \subseteq \mathbb{R}^d$ whose Voronoi cell has a small extension complexity, we may phrase any linear optimization problem over $\mathrm{VC}(\Lambda)$ as a small-size linear program. Such a representation may have several algorithmic consequences for the *closest vector* *problem*. In this problem, one is given Λ in terms of a lattice basis and a point $x \in \mathbb{R}^d$ and is asked to determine a lattice point that is closest to x, i.e., a point in

$$\mathrm{cl}(x, \Lambda) := \left\{ z \in \Lambda : \|x - z\| \leqslant \|x - z'\| \text{ for all } z' \in \Lambda \right\}.$$

Note that $z \in \mathrm{cl}(x, \Lambda)$ if and only if $x - z \in \mathrm{VC}(\Lambda)$. Thus, a small extension complexity of $\mathrm{VC}(\Lambda)$ would yield a small-size linear program to test whether a lattice point is the closest lattice vector to x. However, in view of the fact that the closest vector problem is NP-hard [Emd81] and the belief that $\mathrm{NP} \neq \mathrm{coNP}$, we do not expect efficient algorithms that, for general lattices, decide whether a point is the closest lattice vector to x.

Another sequence of algorithmic implications arises from the algorithm of Micciancio & Voulgaris [MV13], which also motivated other recent work on compact representations of Voronoi cells, such as [HRS20]. As discussed in the thesis of Hunkenschröder [Hun20, §4.1], an optimization oracle for the Voronoi cell of a lattice is sufficient to obtain an algorithm for the closest vector problem that runs in expected polynomial time: Dadush & Bonifas [DB15] describe an efficient procedure to almost uniformly sample a point y from $\mathrm{VC}(\Lambda)$, which can be used to traverse the so-called Voronoi graph by a path of expected polynomial length to obtain a lattice vector that is closest to a given target point x. For sampling y, they only require a membership oracle for $\mathrm{VC}(\Lambda)$, which can be obtained from an optimization oracle [GLS93, §6]. For traversing the Voronoi graph, it is necessary to have an efficient procedure for determining the normal of a facet of $\mathrm{VC}(\Lambda)$ that is intersected by a given line segment. Again, this can be implemented with an optimization oracle for $\mathrm{VC}(\Lambda)$. Clearly, a polynomial-size extended formulation for the Voronoi cell of a lattice yields an efficient implementation of an optimization oracle. This motivates the study of Voronoi cells of lattices for which small-size lifts can be efficiently constructed.

We remark that the mere *existence* of small size extended formulations of Voronoi cells may not be immediately applicable, since finding such representations as well as verifying that they indeed yield the Voronoi cell of a given lattice might be NP-hard. Thus, polynomial bounds on the extension complexities of Voronoi cells of general lattices would not contradict hardness assumptions in complexity theory. In fact, we initially considered the possibility of such bounds.

However, as our main result, we explicitly construct lattices with Voronoi cells of extension complexity close to the trivial upper bound $2(2^d - 1)$.

Theorem 3.2. *There exists a family of d-dimensional lattices Λ such that* $\mathrm{xc}(\mathrm{VC}(\Lambda)) = 2^{\Omega(d/\log d)}$.

Lower bounds on extension complexities have been established for various prominent polytopes in recent years. Of particular note are results for cut polytopes [Fio+15;

KW15; Cha+16], matching polytopes [Rot17], and certain stable-set polytopes [GJW18]. Lower bounds for other polytopes Q are typically obtained by showing that a face F of Q affinely projects onto one of the polytopes P from above and using the simple fact $\mathrm{xc}(P) \leqslant \mathrm{xc}(F) \leqslant \mathrm{xc}(Q)$. Unfortunately, it seems difficult to construct lattices for which this approach can be directly applied to the Voronoi cell. However, we will exploit the lesser known fact that $\mathrm{xc}(Q) = \mathrm{xc}(Q^\circ)$ holds for every polytope Q with the origin in its interior, where Q° is the dual polytope of Q. In fact, we will describe a way to obtain many $0/1$-polytopes as projections of faces of dual polytopes of Voronoi cells of lattices. As an example, for every n-node graph G, we can construct a lattice Λ of dimension at most $n+1$ such that the stable-set polytope of G is a projection of a face of $\mathrm{VC}(\Lambda)^\circ$. Theorem 3.2 then follows from a construction of Göös, Jain & Watson [GJW18] of stable-set polytopes with high extension complexity.

Moreover, we adopt the framework used to obtain Theorem 3.2 to more general settings. On the one hand, we consider small approximations of Voronoi cells. Furthermore, we study another prominent way of representing polytopes via linear projections of feasible regions of *semidefinite* programs, i.e., spectrahedra. We will discuss how our approach also yields versions of Theorem 3.2 for approximations, and for such semidefinite lifts with a slightly weaker but still superpolynomial bound.

Outline In Section 3.2, we provide a brief introduction to lifts of polytopes and lattices, focusing on tools and properties that are essential for our arguments in this chapter. In Section 3.3, we derive upper bounds on the extension complexity of Voronoi cells for some selected classes of lattices, such as root lattices and their duals, zonotopal lattices, and a class of lattices that do not admit a compact representation in the sense of [HRS20]. The proof of Theorem 3.2 is given in Section 3.4, more precisely in Section 3.4.1. Moreover, in Section 3.4.2, we generalize this approach towards approximations of Voronoi cells, and in Section 3.4.3, we briefly introduce semidefinite lifts and present a version of Theorem 3.2 with a superpolynomial bound on the semidefinite extension complexity. We close this chapter with a discussion of open problems in Section 3.5.

3.2 Preliminaries

3.2.1 Extension complexity: A toolbox

Throughout this chapter, we only need basic facts regarding extension complexities of polytopes and most of them are well-known. For the sake of completeness, we provide proofs here. First, we start with a simple fact already mentioned in the introduction.

Lemma 3.3. *For every polytope $P \subseteq \mathbb{R}^d$ and affine subspace $H \subseteq \mathbb{R}^d$, we have* $\mathrm{xc}(P \cap H) \leqslant \mathrm{xc}(P)$.

Proof. If P is the image of a polyhedron $Q \in \mathbb{R}^n$ with k facets under a linear map π, then $P \cap H$ is the image of $\pi^{-1}(H) \cap Q$, where $\pi^{-1}(H)$ is an affine subspace in $\mathbb{R}^n$ and hence $\pi^{-1}(H) \cap Q$ has at most k facets. □

Corollary 3.4. *For every face F of a polytope P, we have* $\mathrm{xc}(F) \leqslant \mathrm{xc}(P)$.

For the next fact, we need the notion of a slack matrix of a polytope. To this end, we consider a polytope $P = \left\{x \in \mathbb{R}^d : \langle a_i, x\rangle \leqslant b_i \text{ for } i \in [m]\right\} = \mathrm{conv}\{v_1, \ldots, v_n\}$. Corresponding to these two descriptions of P, we define the *slack matrix* $S = (S_{i,j}) \in \mathbb{R}^{m\times n}_{\geqslant 0}$ via $S_{i,j} = b_i - \langle a_i, v_j\rangle$. Yannakakis [Yan91] showed that the extension complexity $\mathrm{xc}(P)$ of P equals the *non-negative rank* of S, which is the smallest number r such that $S = FV$, where $F \in \mathbb{R}^{m\times r}_{\geqslant 0}$ and $V \in \mathbb{R}^{r\times n}_{\geqslant 0}$, and which is denoted by $r_+(S)$. This characterization yields another proof for Corollary 3.4, using that the slack matrix of a face of a polytope P is just a submatrix of S.

For a polytope P containing the origin $\mathbf{0}$ in its relative interior, the *dual polytope* of P is defined as

$$P^\circ := \{y \in \mathrm{lin}(P) : \langle x, y\rangle \leqslant 1 \text{ for all } x \in P\}.$$

It is a basic fact that P° is again a polytope with the origin in its relative interior, $\mathrm{lin}(P^\circ) = \mathrm{lin}(P)$, and $(P^\circ)^\circ = P$. Moreover, $(\lambda P)^\circ = \frac{1}{\lambda} P^\circ$ holds for $\lambda \in \mathbb{R}\backslash\{0\}$, and $Q^\circ \subseteq P^\circ$ if $P \subseteq Q$ for polytopes P, Q with $\mathbf{0} \in \mathrm{relint}(P) \cap \mathrm{relint}(Q)$. Furthermore, it is easy to see that if

$$P = \{x \in \mathrm{lin}(P) : \langle w_i, x\rangle \leqslant 1 \text{ for } i \in [m]\} = \mathrm{conv}\{v_1, \ldots, v_n\},$$

then

$$P^\circ = \{y \in \mathrm{lin}(P) : \langle v_i, x\rangle \leqslant 1 \text{ for } i \in [n]\} = \mathrm{conv}\{w_1, \ldots, w_m\}. \tag{3.1}$$

In particular, this shows that if S is a slack matrix of P induced by $v_1, \ldots, v_n$ and $w_1, \ldots, w_m$, then $S^\intercal$ is a slack matrix of P°. Since $r_+(S) = r_+(S^\intercal)$ holds, we obtain the following Lemma 3.5.

Lemma 3.5. *For every polytope $P \subseteq \mathbb{R}^d$ that contains the origin in its relative interior, we have*

$$\mathrm{xc}(P) = \mathrm{xc}(P^\circ).$$

The next statement shows that the extension complexity behaves well under Cartesian products, Minkowski sums and intersections.

Lemma 3.6. *If $P \subseteq \mathbb{R}^d$, $Q \subseteq \mathbb{R}^{d'}$ are polytopes, then*

(i) $\operatorname{xc}(P \times Q) \leqslant \operatorname{xc}(P) + \operatorname{xc}(Q)$.

Moreover, if $d = d'$, then

(ii) $\operatorname{xc}(P + Q) \leqslant \operatorname{xc}(P) + \operatorname{xc}(Q)$ *and*

(iii) $\operatorname{xc}(P \cap Q) \leqslant \operatorname{xc}(P) + \operatorname{xc}(Q)$.

Proof. (i): If P' linearly projects onto P and Q' onto Q, then $P' \times Q'$ linearly projects onto $P \times Q$. Moreover, the number of facets of $P' \times Q'$ is equal to the sum of the number of facets of P' and Q'.
(ii): The polytope $P \times Q$ linearly projects onto $P + Q$ via $(p, q) \mapsto p + q$ for $(p, q) \in P \times Q$, and hence the claim follows from (i).
(iii): If $P = \pi(P')$ and $Q = \tau(Q')$ hold for some polyhedra P', Q' and linear maps π, τ, then $P \cap Q$ is a linear image of the polyhedron $L = \{(y, z) \in P' \times Q' : \pi(y) = \tau(z)\}$. Moreover, the number of facets of L is at most the number of facets of $P' \times Q'$, which, again, is equal to the sum of the number of facets of P' and Q'. □

The next fact is a very useful result following from a work of Balas [Bal79] deriving a description of the convex hull of the union of certain polytopes. The proof of the version presented here can be found in [Wel15, Prop. 3.1.1].

Lemma 3.7. *For polytopes $P_1, \dots, P_k$, we have*

$$\operatorname{xc}(\operatorname{conv}(P_1 \cup \ldots \cup P_k)) \leqslant \sum_{i=1}^{k} \operatorname{xc}(P_i) + |\{i \in [k] : \dim(P_i) = 0\}| .$$

We mentioned already that some lattices have a permutahedron as their Voronoi cell. These polytopes arise from a single vector by permuting its coordinates in all possible ways and taking their convex hull. Let us denote the set of all bijective maps on $[d]$ by S_d. For a permutation $\pi \in S_d$ and a vector $v = (v(1), \dots, v(d)) \in \mathbb{R}^d$, let $\pi(v) := (v(\pi(1)), \dots, v(\pi(d)))$ be the vector that arises from v via permuting its entries according to π.

Lemma 3.8. *For every $v \in \mathbb{R}^d$, we have* $\operatorname{xc}(\operatorname{conv}\{\pi(v) : \pi \in S_d\}) \leqslant d^2$.

Proof. For $\pi \in S_d$, let $P(\pi) \in \{0,1\}^{d \times d}$ be the associated permutation matrix with $P(\pi)_{ij} = 1$ if and only if $\pi(i) = j$ for all $i, j \in [d]$. It is easy to see that $\operatorname{conv}\{\pi(v) : \pi \in S_d\}$ is the image of $B_d := \operatorname{conv}\{P(\pi) : \pi \in S_d\}$ under the linear map $\tau : \mathbb{R}^{d \times d} \to \mathbb{R}^d$ with

$\tau(X)_i = \sum_{j=1}^d v_j X_{ij}$ for $i \in [d]$. The latter polytope is the *Birkhoff–von Neumann polytope* [Bir46; Neu53] described via

$$B_d = \left\{ X \in \mathbb{R}^{d\times d}_{\geqslant 0} : \sum_{i=1}^{d} X_{ij} = 1 \text{ for } j \in [d],\ \sum_{j=1}^{d} X_{ij} = 1 \text{ for } i \in [d] \right\},$$

which has d^2 facets. □

Goemans [Goe15] showed that if $v = (1, 2, \dots, d)$, then the above bound can be improved to $\mathrm{xc}(\mathrm{conv}\{\pi(v) : \pi \in S_d\}) = \Theta(d \log d)$.

3.2.2 Lattices and Voronoi cells

Most basic notions regarding lattices and their Voronoi cells have been already introduced in Section 3.1. In this section, we provide some further definitions and results that we use to obtain bounds on the extension complexity of Voronoi cells of lattices.

We call two lattices $\Lambda, \Gamma \subseteq \mathbb{R}^d$ *isomorphic* if there exists an orthogonal matrix $Q \in \mathbb{R}^{d\times d}$ such that $Q\Lambda = \Gamma$. Note that $\mathrm{VC}(\Gamma) = Q\,\mathrm{VC}(\Lambda)$ and therefore the extension complexities of their Voronoi cells coincide.

In some parts, we will consider the *dual lattice* of a lattice $\Lambda \subseteq \mathbb{R}^d$, which is defined as

$$\Lambda^\star = \{x \in \mathrm{lin}(\Lambda) : \langle x, y\rangle \in \mathbb{Z} \text{ for all } y \in \Lambda\}\,.$$

Note that for every two lattices Λ, Γ, their product $\Lambda \times \Gamma$ is also a lattice. The following lemma shows that the Cartesian product behaves well with respect to Voronoi cells or duals of lattices.

Lemma 3.9. *For any two lattices $\Lambda \subseteq \mathbb{R}^d$ and $\Gamma \subseteq \mathbb{R}^{d'}$, we have*

(i) $\mathrm{VC}(\Lambda \times \Gamma) = \mathrm{VC}(\Lambda) \times \mathrm{VC}(\Gamma)$*, and*

(ii) $(\Lambda \times \Gamma)^\star = \Lambda^\star \times \Gamma^\star$.

Proof. The first claim follows since

$$\begin{aligned}
\mathrm{VC}(\Lambda \times \Gamma) &= \{(x,y) \in \mathrm{lin}(\Lambda\times\Gamma) : \|(x,y)\|^2 \leqslant \|(x,y)-(w,z)\|^2 \text{ for all } (w,z) \in \Lambda \times \Gamma\} \\
&= \{(x,y) : x \in \mathrm{lin}(\Lambda),\ y \in \mathrm{lin}(\Gamma), \\
&\qquad \|x\|^2 + \|y\|^2 \leqslant \|x-w\|^2 + \|y-z\|^2 \text{ for all } (w,z) \in \Lambda\times\Gamma\} \\
&= \{(x,y) : x \in \mathrm{lin}(\Lambda),\ y \in \mathrm{lin}(\Gamma), \\
&\qquad \|x\|^2 \leqslant \|x-w\|^2 \text{ for all } w \in \Lambda,\ \|y\|^2 \leqslant \|y-z\|^2 \text{ for all } z \in \Gamma\} \\
&= \mathrm{VC}(\Lambda) \times \mathrm{VC}(\Gamma)
\end{aligned}$$

holds.

For the second claim, it is clear that $\Lambda^\star \times \Gamma^\star \subseteq (\Lambda \times \Gamma)^\star$ holds. To see that the reverse inclusion holds as well, let $(x, y) \in (\Lambda \times \Gamma)^\star$. For every $w \in \Lambda$, we have $(w, \mathbf{0}) \in \Lambda \times \Gamma$ and hence $\langle x, w\rangle = \langle (x, y), (w, \mathbf{0})\rangle \in \mathbb{Z}$. This yields $x \in \Lambda^\star$. We obtain $y \in \Gamma^\star$ in an analogous fashion. □

A main ingredient for proving Theorem 3.2 is to consider the dual polytope $\mathrm{VC}(\Lambda)^\circ$ of $\mathrm{VC}(\Lambda)$. Recall that we have $\mathrm{xc}(\mathrm{VC}(\Lambda)) = \mathrm{xc}(\mathrm{VC}(\Lambda)^\circ)$ by Lemma 3.5. The following two observations are crucial for our arguments.

Lemma 3.10. *For every lattice Λ, we have*

$$\mathrm{VC}(\Lambda)^\circ = \mathrm{conv}\left\{ \tfrac{2}{\|z\|^2} z : z \in \Lambda\backslash\{\mathbf{0}\}\right\}.$$

Proof. In view of the identities

$$\begin{aligned}\mathrm{VC}(\Lambda) &= \left\{x \in \mathrm{lin}(\Lambda) : \|x\|^2 \leqslant \|x - z\|^2 \text{ for all } z \in \Lambda\right\}\\ &= \left\{x \in \mathrm{lin}(\Lambda) : \langle x, z\rangle \leqslant \tfrac{1}{2}\|z\|^2 \text{ for all } z \in \Lambda\right\}\\ &= \left\{x \in \mathrm{lin}(\Lambda) : \left\langle x, \tfrac{2}{\|z\|^2} z\right\rangle \leqslant 1 \text{ for all } z \in \Lambda\backslash\{\mathbf{0}\}\right\},\end{aligned}$$

the claim follows from (3.1). □

Lemma 3.11. *Let $\Lambda \subseteq \mathbb{R}^d$ be a lattice and $p \in \mathbb{R}^d$. If $\mathbf{0} \in \mathrm{cl}(p, \Lambda)$, then*

$$\mathrm{conv}\left\{ \tfrac{2}{\|z\|^2} z : z \in \mathrm{cl}(p, \Lambda)\backslash\{\mathbf{0}\}\right\}$$

is a face of $\mathrm{VC}(\Lambda)^\circ$.

Proof. Since $\mathbf{0} \in \mathrm{cl}(p, \Lambda)$, every non-zero lattice point $z \in \Lambda\backslash\{\mathbf{0}\}$ satisfies $\|p - z\|^2 \geqslant \|p\|^2$, with equality if and only if $z \in \mathrm{cl}(p, \Lambda)\backslash\{\mathbf{0}\}$. Note that the above inequality is equivalent to $\left\langle p, \tfrac{2}{\|z\|^2} z\right\rangle \leqslant 1$. Thus, due to Lemma 3.10, we see that $F := \{y \in \mathrm{VC}(\Lambda)^\circ : \langle p, y\rangle = 1\}$ is a face of $\mathrm{VC}(\Lambda)^\circ$. This establishes the claim since

$$\begin{aligned}F &= \mathrm{conv}\left\{ \tfrac{2}{\|z\|^2} z : z \in \Lambda\backslash\{\mathbf{0}\}, \left\langle p, \tfrac{2}{\|z\|^2} z\right\rangle = 1\right\}\\ &= \mathrm{conv}\left\{ \tfrac{2}{\|z\|^2} z : z \in \mathrm{cl}(p, \Lambda)\backslash\{\mathbf{0}\}\right\}.\end{aligned}$$ □

3.3 Lattices with small extension complexity

In this section, we provide bounds on the extension complexities of Voronoi cells of some prominent lattices.

3.3.1 Root lattices and their duals

We start with Voronoi cells of root lattices and their duals. An *irreducible root lattice* is a lattice Λ for which there exists a finite set S of vectors of squared length equal to one or two, such that $\Lambda = \{\sum_{b \in S} \alpha_b b : \alpha_b \in \mathbb{Z} \text{ for all } b \in S\}$. We say that a lattice is a *(general) root lattice*, if it is isomorphic to a lattice obtained by iteratively taking Cartesian products with irreducible root lattices. A well-known theorem related to the classification of reflection groups states that besides the lattice $\mathbb{Z}^d$ of integers, up to isomorphism the irreducible root lattices split into the two infinite classes

$$\begin{aligned} A_d &= \left\{ x \in \mathbb{Z}^{d+1} : x(1) + \ldots + x(d+1) = 0 \right\} \quad \text{and} \\ D_d &= \left\{ x \in \mathbb{Z}^{d} : x(1) + \ldots + x(d) \text{ is even} \right\}, \end{aligned}$$

and the three exceptional lattices

$$\begin{aligned} E_8 &= D_8 \cup \left(\tfrac{1}{2}\mathbf{1} + D_8\right), \\ E_7 &= \{x \in E_8 : \langle x, \chi_7 + \chi_8 \rangle = 0\} \text{ and} \\ E_6 &= \{x \in E_7 : \langle x, \chi_6 + \chi_8 \rangle = 0\}. \end{aligned}$$

Here and in the following, we denote by χ_i the ith standard Euclidean unit vector. Moreover, the dual lattices of the two infinite classes A_d and D_d are given by

$$A_d^\star = \bigcup_{i=0}^{d} (v_i + A_d),$$

with $v_i = \Big(\underbrace{\tfrac{i}{d+1}, \ldots, \tfrac{i}{d+1}}_{j \text{ times}}, \underbrace{-\tfrac{j}{d+1}, \ldots, -\tfrac{j}{d+1}}_{i \text{ times}} \Big)$ for $0 \leqslant i \leqslant d$ and $j = d + 1 - i$, and

$$D_d^\star = \mathbb{Z}^d \cup \left(\tfrac{1}{2}\mathbf{1} + \mathbb{Z}^d \right),$$

respectively. In the literature the dual $D_d^\star$ is usually scaled by a factor of two in order to get an integral lattice, which is often more convenient to investigate. In order to avoid confusion, we denote it by

$$\bar{D}_d^\star := 2D_d^\star = \left(2\mathbb{Z}^d\right) \cup \left(\mathbf{1} + 2\mathbb{Z}^d\right),$$

and note that this scaling has no effect on the extension complexity of its Voronoi cell. We refer to Conway & Sloane [CS99, Ch. 4 & Ch. 21] and Martinet [Mar03, Ch. 4] for proofs, original references, and background information on root lattices. Details on Voronoi cells and Delaunay polytopes of root lattices can be found in Moody &

Patera [MP92], which together with the two aforementioned monographs are our main sources of information.

Given a lattice $\Lambda \subseteq \mathbb{R}^d$, we write $|\Lambda| = \min\{\|z\| : z \in \Lambda\backslash\{\mathbf{0}\}\}$ for the length of a shortest non-trivial vector in Λ. A *minimal vector* of Λ is any vector $z \in \Lambda$ with $\|z\| = |\Lambda|$, and a *facet vector* of Λ is any vector $w \in \Lambda$, such that the constraint $\langle x, w\rangle \leqslant \frac{1}{2}\|w\|^2$ defines a facet of the Voronoi cell $\mathrm{VC}(\Lambda)$. For convenience, we write

$$\begin{aligned}\mathcal{S}(\Lambda) &= \{z \in \Lambda : \|z\| = |\Lambda|\} \quad \text{and}\\ \mathcal{F}(\Lambda) &= \{w \in \Lambda : w \text{ is a facet vector of } \Lambda\}\,,\end{aligned}$$

for the set of minimal vectors and facet vectors, respectively. In general, one has the inclusion $\mathcal{S}(\Lambda) \subseteq \mathcal{F}(\Lambda)$, which however is usually strict. Root lattices are now neatly characterized by the property that every facet vector is at the same time a minimal vector, that is, the equality $\mathcal{S}(\Lambda) = \mathcal{F}(\Lambda)$ holds, see Rajan & Shende [RS96].

Since the set of minimal vectors of the irreducible root lattices are well-understood, this allows to describe their Voronoi cells as well. For the sake of the asymptotic study of the extension complexity of their Voronoi cells, it suffices to understand the two infinite families A_d and D_d, and their duals $A_d^\star$ and $D_d^\star$. In the sequel, we provide bounds on the extension complexities of the Voronoi cells of these lattices. To achieve these bounds, we sometimes use a characterization of the facet vectors and in other cases we use a characterization of the vertices of the Voronoi cell. For the sake of easy reference, we describe the vertices and facet vectors in all cases. Due to Lemma 3.6 and Lemma 3.9, these bounds directly imply Theorem 3.1. Moreover, the bound in Theorem 3.1 is asymptotically tight since the Voronoi cell of $A_d^\star$ is a permutahedron, see Lemma 3.14.

Voronoi cell of A_d

The Voronoi cell of the root lattice A_d is given by

$$\mathrm{VC}(A_d) = \mathrm{conv}\,\{\pi(v_i) : \pi \in S_{d+1} \text{ for } i \in \{0,\dots,d\}\}\,,$$

where

$$v_i = \Big(\underbrace{\tfrac{i}{d+1},\dots,\tfrac{i}{d+1}}_{j \text{ times}}, \underbrace{-\tfrac{j}{d+1},\dots,-\tfrac{j}{d+1}}_{i \text{ times}}\Big) \in \mathbb{R}^{d+1}$$

with $j = d + 1 - i$. Moreover, we have

$$\begin{aligned}\mathrm{VC}(A_d) &= \Big\{x \in \mathbb{R}^{d+1} : \langle x, z\rangle \leqslant 1 \text{ for all } z \in \mathcal{F}(A_d)\Big\}\,, \text{ where}\\ \mathcal{F}(A_d) &= \{\pi((1,-1,0,\dots,0)) : \pi \in S_{d+1}\}\,,\end{aligned}$$

see [CS99, Ch. 21 & Ch. 4, Sec. 6].

Lemma 3.12. $\mathrm{xc}(\mathrm{VC}(A_d)) = \mathcal{O}(d)$.

Proof. Using the description of the facet vectors stated above, we obtain that $\mathrm{VC}(A_d)^\circ = S + (-S)$, where S is the d-dimensional simplex $S = \mathrm{conv}\{\chi_1, \ldots, \chi_{d+1}\}$. Hence, using Lemma 3.5 and Lemma 3.6, we obtain the upper bound $\mathrm{xc}(\mathrm{VC}(A_d)) \leqslant 2(d+1)$. □

Voronoi cell of D_d

The Voronoi cell of D_d is given by

$$\mathrm{VC}(D_d) = \mathrm{conv}\left(\{\pm\chi_1, \ldots, \pm\chi_d\} \cup \{-\tfrac{1}{2}, \tfrac{1}{2}\}^d\right).$$

Moreover, we have

$$\mathrm{VC}(D_d) = \left\{x \in \mathbb{R}^{d+1} : \langle x, z\rangle \leqslant 1 \text{ for all } z \in \mathcal{F}(D_d)\right\}, \text{ where}$$
$$\mathcal{F}(D_d) = \{\pm\chi_i \pm \chi_j : 1 \leqslant i < j \leqslant d\}.$$

This follows from the characterization of the minimal (and thus facet) vectors of D_d given in [CS99, Ch. 4, Sec. 7]. The inner description of the Voronoi cell can be read off from the vertices of a fundamental simplex for D_d, see [CS99, Ch. 21, Fig. 21.7].

Lemma 3.13. $\mathrm{xc}(\mathrm{VC}(D_d)) = \mathcal{O}(d)$.

Proof. Using the description of the vertices of $\mathrm{VC}(D_d)$ stated above, we obtain that the dual of the Voronoi cell is the intersection of a hypercube and a crosspolytope, i.e.,

$$\mathrm{VC}(D_d)^\circ = 2 \cdot \mathrm{conv}\{\pm\chi_1, \ldots, \pm\chi_d\} \cap [-1, 1]^d.$$

Since

$$\mathrm{xc}\left([-1,1]^d\right) = \mathrm{xc}(\mathrm{conv}\{\pm\chi_1, \ldots, \pm\chi_d\}) = 2d, \tag{3.2}$$

see, e.g., [GPS18, Cor. 2.5]. Lemmas 3.5 and 3.6 imply $\mathrm{xc}(\mathrm{VC}(D_d)) = \mathcal{O}(d)$. □

Voronoi cell of $A_d^\star$

The Voronoi cell of the dual of the root lattice $A_d^\star$ is given by

$$\mathrm{VC}(A_d^\star) = \mathrm{conv}\{\pi(v) : \pi \in S_{d+1}\},$$

where

$$v = \tfrac{1}{2d+2}(-d, -d+2, -d+4, \ldots, d-4, d-2, d) \in \mathbb{R}^{d+1}.$$

Moreover, we have

$$\mathcal{F}(A_d^\star) = \{v \in \mathrm{lin}(A_d^\star) : v \text{ is a vertex of } \mathrm{VC}(A_d)\}.$$

The characterization of the vertices can be found in [CS99, Ch. 21, Sec. 3F] and the fact that the facet vectors are exactly the vertices of $\mathrm{VC}(A_d)$ is explained in detail in the unpublished monograph [EMS04, Ch. 3.5].

Lemma 3.14. $\mathrm{xc}(\mathrm{VC}(A_d^\star)) = \Theta(d \log d)$.

Proof. Using the description of the vertices of $\mathrm{VC}(A_d^\star)$ stated before, we obtain that $\mathrm{VC}(A_d^\star)$ is an affine linear transformation of the standard permutahedron

$$P_d = \{(\pi(1), \ldots, \pi(d+1)) : \pi \in S_{d+1}\}.$$

In fact,

$$\mathrm{VC}(A_d^\star) = \tfrac{1}{d+1} P_d - \tfrac{d+2}{2d+2}\mathbf{1}.$$

The claim follows, since Goemans [Goe15] showed that the extension complexity of P_d is in $\Theta(d \log d)$. □

Voronoi cell of $D_d^\star$

As explained before, we consider the integral lattice $\bar{D}_d^\star$ instead of $D_d^\star$. The Voronoi cell of $\bar{D}_d^\star$ is given by

$$\mathrm{VC}\left(\bar{D}_d^\star\right) = \mathrm{conv}\left\{\pi(v) : \pi \in S_d, v \in V\right\},$$

where

$$V = \begin{cases} \{0\}^{\frac{d}{2}} \times \{-1,1\}^{\frac{d}{2}} & \text{, if } d \text{ is even,} \\ \{0\}^{\frac{d-1}{2}} \times \left\{-\frac{1}{2}, \frac{1}{2}\right\} \times \{-1,1\}^{\frac{d-1}{2}} & \text{, if } d \text{ is odd.} \end{cases}$$

Moreover, we have

$$\mathcal{F}(\bar{D}_d^\star) = \{\pm 2\chi_1, \ldots, \pm 2\chi_d\} \cup \{-1,1\}^d.$$

We refer to [CS99, Ch. 21, Sect. 3E] for the characterization of the facet vectors and the inner description of the Voronoi cell, which is therein denoted by the symbols $\beta(d, d/2)$ for even d, and $\frac{1}{2}\delta(d, (d-1)/2)$ for odd d.

Lemma 3.15. $\mathrm{xc}(\mathrm{VC}(D_d^\star)) = \mathcal{O}(d)$.

Proof. Using the above description of the facet vectors, we obtain that the Voronoi cell of $D_d^\star$ is the intersection of a hypercube and a crosspolytope, i.e.,

$$\mathrm{VC}\left(\bar{D}_d^\star\right) = [-1,1]^d \cap \tfrac{d}{2} \cdot \mathrm{conv}\{\pm\chi_1, \ldots, \pm\chi_d\}.$$

As in the case of the root lattice D_d, the stated bound follows by Lemma 3.6 and Equation (3.2). □

Note that all the bounds stated in Lemmas 3.12, 3.13, and 3.15 are asymptotically tight, since the extension complexity of a polytope grows at least linearly with its dimension [Fio+13, Eq. 2 & Prop. 5.2].

3.3.2 Zonotopal lattices

A *zonotope* $Z \subseteq \mathbb{R}^d$ is the Minkowski sum of finitely many line segments, that is, there are vectors $a_1, b_1, \ldots, a_m, b_m \in \mathbb{R}^d$ such that $Z = \sum_{i=1}^m \operatorname{conv}\{a_i, b_i\}$. The non-zero vectors $z_i = b_i - a_i$ are usually called the *generators* of the zonotope, and clearly, Z is an affine projection of the m-dimensional cube $[-1,1]^m$ via $\chi_i \mapsto z_i$ for $1 \leqslant i \leqslant m$ and a suitable translation. Regarding the extension complexity of a zonotope Z, the bound $\operatorname{xc}(Z) \leqslant 2m$ immediately follows from the definition.

A lattice $\Lambda \subseteq \mathbb{R}^d$ is said to be *zonotopal* if its Voronoi cell is a zonotope. Every lattice of dimension at most three is zonotopal, but from dimension four on there exist non-zonotopal lattices. For instance, the Voronoi cell of the root lattice D_4 is the non-zonotopal 24-cell. Examples of classes of zonotopal lattices are $\mathbb{Z}^d$, the root lattice A_d, its dual lattice $A_d^\star$, lattices of Voronoi's first kind, and the tensor product $A_d \otimes A_{d'}$. Zonotopal space tiles have been extensively studied over the years, mostly due to their combinatorial connections to regular matroids, hyperplane arrangements, and totally unimodular matrices. For a detailed account on zonotopal lattices and pointers to the original works containing the previous statements we refer to [McC+21, Sect. 2].

The tiling constraint on a zonotope that arises as the Voronoi cell of a lattice, allows it to have at most quadratically many generators in terms of its dimension. In particular, these polytopes admit lifts with quadratically many facets.

Theorem 3.16. *Each zonotopal lattice $\Lambda \subseteq \mathbb{R}^d$ satisfies $\operatorname{xc}(\operatorname{VC}(\Lambda)) \leqslant d(d+1)$.*

Proof. It suffices to argue that the Voronoi cell is generated by at most $\binom{d+1}{2}$ line segments. Indeed, each line segment L satisfies $\operatorname{xc}(L) = 2$ and hence the statement follows using Lemma 3.7.

Erdahl [Erd99, Sect. 5] proved that the generators of a space tiling zonotope correspond to the normal vectors of a certain dicing. A *dicing* in $\mathbb{R}^d$ is an arrangement of hyperplanes consisting of $r \geqslant d$ families of infinitely many equally-spaced hyperplanes such that: (1) there are d families whose corresponding normal vectors are linearly independent, and (2) every vertex (0-dimensional affine subspace arising form intersecting hyperplanes) of the arrangement is contained in a hyperplane of each family.

By [Erd99, Theorem. 3.3], every dicing is affinely equivalent to a dicing whose set of hyperplane normal vectors – one normal vector for each of the r families – consists of the columns of a totally unimodular $d \times r$ matrix. By construction, this totally unimodular matrix is such that for any two of its columns v, w, we have $v \neq \pm w$ and $v, w \neq \mathbf{0}$. A classical result that is often attributed to Heller [Hel57], but already appears in Korkine & Zolotarev [KZ77], yields that every such totally unimodular $d \times r$ matrix has at most $r \leqslant \binom{d+1}{2}$ columns. Thus, the zonotopal Voronoi cell $\operatorname{VC}(\Lambda)$ is generated by at most $\binom{d+1}{2}$ line segments. □

Alternatively, the fact that zonotopal Voronoi cells in $\mathbb{R}^d$ are generated by at most $\binom{d+1}{2}$ line segments also follows from Voronoi's reduction theory. The Delaunay subdivisions of zonotopal lattices correspond to certain polyhedral cones (Voronoi's L-types) in the cone $\mathcal{S}^d_{\geqslant 0}$ of positive semi-definite $d \times d$ matrices that are generated by rank one matrices. Since $\mathcal{S}^d_{\geqslant 0}$ has dimension $\binom{d+1}{2}$, Carathéodory's Theorem yields the bound. We refer the reader to Erdahl [Erd99, Sect. 7] for an intuitive description and references to the original works.

3.3.3 Lattices defined by simple congruences

For any $a \in \mathbb{Z}_{\geqslant 1}$, we consider the lattice

$$\Lambda_d(a) := \left\{ x \in \mathbb{Z}^d : x_1 \equiv x_2 \equiv \ldots \equiv x_d \mod a \right\}. \tag{3.3}$$

The case $a = \left\lceil \frac{d}{2} \right\rceil$ played a special role in [HRS20, Thm. 2] for the determination of lattices that do not have a basis that admits a compact (in their setting) representation of the Voronoi cell. To this end, the authors specified the set $\mathcal{F}\left(\Lambda_d\left(\left\lceil \frac{d}{2} \right\rceil\right)\right)$ of facet vectors explicitly. Note that there are exponentially many of facet vectors. However, their proof can be extended to general a to give a description of the facet vectors of $\mathcal{F}(\Lambda_d(a))$ that is precise enough to allow drawing conclusions towards small extended formulations.

Lemma 3.17. *For all $a \in \mathbb{Z}_{\geqslant 1}$, the set of facet vectors of $\Lambda_d(a)$ is contained in*

$$\begin{aligned} \mathcal{F}(\Lambda_d(a)) \subseteq \{\mathbf{1}, -\mathbf{1}\} &\cup \{\pm a \chi_i : i \in [d]\} \\ &\cup \left\{ v_{S,\ell} \in \mathbb{R}^d : \varnothing \neq S \subsetneq [d], \ell \in \left\{ \left\lfloor \tfrac{a|S|}{d} \right\rfloor, \left\lceil \tfrac{a|S|}{d} \right\rceil \right\} \right\}, \end{aligned}$$

where $v_{S,\ell}(i) = a - \ell$, if $i \in S$, and $v_{S,\ell}(i) = -\ell$, if $i \notin S$.

Proof. Follows directly with the proof of [HRS20, Lem. 3]. □

Theorem 3.18. *For all $a \in \mathbb{Z}_{\geqslant 1}$, we have $\mathrm{xc}(\mathrm{VC}(\Lambda_d(a))) = \mathcal{O}(d^3)$.*

Proof. Due to Lemma 3.17, Equation (3.1), the definition of the Voronoi cell and the fact that the set of points given in Lemma 3.17 does only contain lattice vectors, the dual polytope of the Voronoi cell of $\Lambda_d(a)$ equals

$$\mathrm{VC}(\Lambda_d(a))^\circ = \mathrm{conv}\left(V_{\pm\mathbf{1}} \cup V_{\pm a} \cup \bigcup_{k,\ell} V_{k,\ell} \right),$$

where the last union is over all $k \in [d-1]$, $\ell \in \left\{\left\lfloor \frac{ak}{d} \right\rfloor, \left\lceil \frac{ak}{d} \right\rceil\right\}$ and the sets $V_{\pm\mathbf{1}}, V_{\pm a}$ and $V_{k,\ell}$ are defined as follows:

$$\begin{aligned} V_{\pm\mathbf{1}} &:= \operatorname{conv}\left\{\tfrac{2}{d}\mathbf{1}, -\tfrac{2}{d}\mathbf{1}\right\}, \\ V_{\pm a} &:= \operatorname{conv}\left\{\pm\tfrac{2}{a^2}a\chi_i : i \in [d]\right\} \text{ and} \\ V_{k,\ell} &:= \operatorname{conv}\Big\{\tfrac{2}{k(a-\ell)^2+(d-k)\ell^2}z : z \in \{a-\ell, -\ell\}^d \\ &\qquad\qquad\qquad\qquad \text{with exactly } k \text{ entries equal to } a-\ell\Big\}. \end{aligned}$$

Clearly, $\operatorname{xc}(V_{\pm\mathbf{1}}) = 2$ holds. Since $V_{\pm a}$ is a crosspolytope, $\operatorname{xc}(V_{\pm a}) = 2d$ holds, see (3.2). Furthermore, for $k \in [d-1]$ and $\ell \in \left\{\left\lfloor \frac{ak}{d} \right\rfloor, \left\lceil \frac{ak}{d} \right\rceil\right\}$, using Lemma 3.8 and the fact that $V_{k,\ell}$ up to scaling equals

$$\operatorname{conv}\{\pi(v_{k,\ell}) : \pi \in S_d\},$$

where $v_{k,\ell} = (a-\ell,\ldots,a-\ell,-\ell,\ldots,-\ell)$ with exactly k entries equal to $a-\ell$, we obtain $\operatorname{xc}(V_{k,\ell}) \leqslant d^2$.

Combining these bounds and applying the Lemmas 3.5 and 3.7, we obtain the desired bound. □

3.4 Lower bounds on the extension complexity of Voronoi cells

The main contribution of this section is the proof of Theorem 3.2. After that, we generalize the ideas to approximations and semidefinite lifts.

3.4.1 Lattices with large extension complexity

The aim of this section is to prove Theorem 3.2. Inspired by Kannan's proof [Kan87, Sec. 6] of the NP-hardness of the closest vector problem, we gain the following result. For every 0/1-polytope P, we are able to construct a lattice such that a face of its dual Voronoi cell projects onto P. To obtain a lattice of small dimension, P needs to fulfill some extra condition.

Lemma 3.19. *Let $H \subseteq \mathbb{R}^k$ be an affine subspace such that all vectors in $X := \{0,1\}^k \cap H$ have the same norm. There is a lattice Λ with $\dim(\Lambda) \leqslant \dim(H)+1$ such that $\operatorname{conv}(X)$ is a linear projection of a face of $\operatorname{VC}(\Lambda)^\circ$.*

Proof. Let $\gamma \geqslant 0$ be such that $\|x\| = \gamma$ for all $x \in X$. We may assume that H is nonempty and that $\gamma > 0$, otherwise $\operatorname{conv}(X)$ is empty or consists of a single point, in which case the claim is trivial. Consider $h \in H$ and let L be the linear subspace such that $H = L + h$. Now, consider the lattice

$$\Lambda := \left\{z = (z', z'') \in \mathbb{Z}^k \times \gamma\mathbb{Z} : z' + \tfrac{1}{\gamma}z''h \in L,\ \langle \mathbf{1}, z'\rangle + \gamma z'' = 0\right\}, \tag{3.4}$$

and let $p := (\mathbf{0}, -\gamma) \in \mathbb{R}^{k+1}$. We will show that

$$\mathrm{cl}(p, \Lambda) = \{\mathbf{0}\} \cup \{(x, -\gamma) : x \in X\} =: U \tag{3.5}$$

holds. By Lemma 3.11, this will imply that

$$\mathrm{conv}\left\{\tfrac{2}{\|(x,-\gamma)\|^2}(x, -\gamma) : x \in X\right\} = \mathrm{conv}\left\{\tfrac{1}{\gamma^2}(x, -\gamma) : x \in X\right\}$$

is a face of $\mathrm{VC}(\Lambda)^\circ$ that linearly projects onto $\mathrm{conv}(X)$.

First note that $U \subseteq \Lambda$ holds. Moreover, we have $\|p - \mathbf{0}\| = \gamma$, and $\|p - (x, -\gamma)\| = \|x\| = \gamma$ holds for each $x \in X$. Thus, in order to establish (3.5) it remains to show that every lattice point $z = (z', z'') \in \Lambda \backslash U$ satisfies $\gamma < \|p - z\|$. Equivalently, we have to show that every such point satisfies

$$f(z) := \|z'\|^2 + \|z'' + \gamma\|^2 > \gamma^2. \tag{3.6}$$

This is clear if $z'' \notin \{0, -\gamma, -2\gamma\}$. If $z'' = 0$, then since $z \notin U$, we must have $z' \neq \mathbf{0}$. Hence, $f(z) = \|z'\|^2 + \gamma^2 > \gamma^2$ holds.

If $z'' = -\gamma$, then $z' \in H$ and $\langle \mathbf{1}, z' \rangle = \gamma^2$ hold. Since $z' \in \mathbb{Z}^k$, we obtain $f(z) = \|z'\|^2 \geqslant \langle \mathbf{1}, z' \rangle = \gamma^2$ with equality only if $z' \in \{0,1\}^k$. However, in the latter case, we would have $z' \in \{0,1\}^k \cap H = X$ and hence $z \in U$, a contradiction. Thus, we obtain $f(z) > \gamma^2$.

Finally, if $z'' = -2\gamma$, then $f(z) = \|z'\|^2 + \gamma^2$ and $\langle \mathbf{1}, z' \rangle = 2\gamma^2 > 0$, implying $z' \neq \mathbf{0}$ and hence (3.6) holds. □

While the previous lemma appears quite restrictive, the next lemma shows that we may apply it to a large class of $0/1$-polytopes.

Lemma 3.20. *Let $X = \{x \in \{0,1\}^k : Ax \leqslant b\}$, for some $A \in \mathbb{R}^{m \times k}$, $b \in \mathbb{R}^m$ such that $b - Ax \in \{0,1\}^m$, for all $x \in X$. There is a lattice Λ of dimension at most $k+1$ such that $\mathrm{conv}(X)$ is the linear projection of a face of $\mathrm{VC}(\Lambda)^\circ$.*

Proof. Consider the set

$$X' := \left\{(x, x', s, s') \in \{0,1\}^{k+k+m+m} : Ax + s = b, x + x' = \mathbf{1}, s + s' = \mathbf{1}\right\},$$

and observe that projecting X' onto the first k coordinates yields the set X. Moreover, notice that every vector in X' consists of exactly $k + m$ ones. In other words, the norm of every vector in X' equals $\sqrt{k+m}$ and hence, we may apply Lemma 3.19 to obtain a lattice Λ with dimension at most $k+1$ such that $\mathrm{conv}(X')$ is the linear projection of a face F of $\mathrm{VC}(\Lambda)^\circ$. Since $\mathrm{conv}(X)$ is a linear projection of $\mathrm{conv}(X')$, we see that $\mathrm{conv}(X)$ is also a linear projection of F. □

Proof of Theorem 3.2. We use a result of Göös, Jain & Watson [GJW18] that yields a family of n-node graphs G such that the stable-set polytope P_G of G satisfies $\mathrm{xc}(P_G) = 2^{\Omega(n/\log n)}$. Let $X \subseteq \{0,1\}^n$ denote the set of characteristic vectors of stable-sets in G. Notice that

$$X = \{x \in \{0,1\}^n : x(i) + x(j) \leqslant 1 \text{ for all } \{i,j\} \in E(G)\}.$$

By Lemma 3.20, there is a d-dimensional lattice Λ with $d \leqslant n+1$ such that $\mathrm{conv}(X)$ is a linear projection of a face F of $\mathrm{VC}(\Lambda)^\circ$. We conclude

$$\mathrm{xc}(\mathrm{VC}(\Lambda)) = \mathrm{xc}(\mathrm{VC}(\Lambda)^\circ) \geqslant \mathrm{xc}(F) \geqslant \mathrm{xc}(\mathrm{conv}(X)) = \mathrm{xc}(P_G) = 2^{\Omega(n/\log n)},$$

and the claim follows since $d = \mathcal{O}(n)$. $\square$

3.4.2 Generalization to approximations of Voronoi cells

In Section 3.4.1, we have seen that the extension complexity of Voronoi cells can be superpolynomial. Within this section, we study whether this still holds true if we allow approximations instead of exact descriptions. For $\alpha \geqslant 1$, we say that a polytope Q is an α-*approximation* of a polytope P if $P \subseteq Q \subseteq \alpha P$. This is a common way to define approximations of polytopes, since it also perfectly matches the approximation factor when optimizing over the respective polytopes. For any objective vector c, we have

$$\max\{c^\intercal x : x \in P\} \leqslant \max\{c^\intercal x : x \in Q\} \leqslant \alpha \max\{c^\intercal x : x \in P\}.$$

In Section 3.4.1, the lower bound on the extension complexity was obtained by constructing a lattice whose dual Voronoi cell possesses a face that projects onto a stable-set polytope and exploiting the lower bound for stable-set polytopes. It turns out that there also is a superpolynomial lower bound on the extension complexity of small approximations of stable-set polytopes, see Lemma 3.23. Therefore, we tailor the approach of Section 3.4.1 for our result on approximations. Dualizing polytopes behaves well with containment, and therefore we can easily extend the idea of considering the dual Voronoi cell to approximations. Furthermore, Lemma 3.22 shows how to generalize the idea of projecting a face. Combining all these ingredients results in the following theorem.

Theorem 3.21. *There exists a family of d-dimensional lattices Λ such that for any polytope Q satisfying $\mathrm{VC}(\Lambda) \subseteq Q \subseteq \left(1 + \frac{1}{192d^4}\right) \mathrm{VC}(\Lambda)$, we have $\mathrm{xc}(Q) = 2^{\Omega(\sqrt{d})}$.*

The next Lemma 3.22 deals with the generalization of projecting a face. Therefore, consider a polytope V, which possesses a face F that projects onto a polytope P. For a polytope Q approximating V, Lemma 3.22 essentially shows that Q intersected with the affine hull of F will project onto a polytope K that approximates P. To be more precise, this only holds true if $\mathbf{0} \in \operatorname{relint}(P)$. Since we aim to apply this lemma for V being the dual Voronoi cell and P the stable-set polytope, we will only have $\mathbf{0} \in P$. In this case, we will have to cut the projected slice of Q with the positive hull of P, which is defined by

$$\operatorname{pos}(P) := \bigcap_{\substack{C \supseteq P, \\ C \text{ convex cone}}} C.$$

Lemma 3.22. *Let $\gamma > 0$, $P \subseteq \mathbb{R}^n$ and $V \subseteq \mathbb{R}^d$ be polytopes with $\mathbf{0} \in \operatorname{relint}(V)$, $\mathbf{0} \in P$, and F be a face of V such that $\pi(F) = P$, where π is a linear map. Moreover, let $a \in \mathbb{R}^d$ such that the hyperplane $H = \{y \in \mathbb{R}^d : \langle a, y \rangle = 1\}$ satisfies $F = V \cap H$. Then, there exists an $\varepsilon > 0$ such that for any polytope Q satisfying*

$$V \subseteq Q \subseteq (1+\varepsilon)V,$$

we have

$$P \subseteq \pi(Q \cap H) \cap \operatorname{pos}(P) \subseteq (1+\gamma)P.$$

Here, ε depends on P, V, γ, and π and can be determined as follows. Let $\varsigma > 0$ be such that $\langle a, v \rangle + \varsigma \leqslant 1$ holds for all vertices v of V that do not lie in the face F. Let r_V satisfy $r_V \geqslant \max\{\|y\| : y \in V\}$ and for P described via $P = \{x \in \mathbb{R}^n : \langle c_i, x \rangle \leqslant 1 \text{ for all } i \in [k]\} \cap \operatorname{pos}(P)$, let r_P° be a bound for the coefficient vectors, i.e., $r_P^\circ \geqslant \max\{\|c_i\| : i \in [k]\}$. Then, the above statement is true for all

$$\varepsilon \leqslant \frac{\gamma\varsigma}{\varsigma + r_V r_P^\circ \|\pi\|},$$

where $\|\pi\| < \infty$ denotes the operator norm of π corresponding to the Euclidean norms in $\mathbb{R}^n$ and $\mathbb{R}^d$.

Proof. The first inclusion is trivial, since $F \subseteq Q \cap H$ and therefore $P = \pi(F) \subseteq \pi(Q \cap H)$.

For the second inclusion, we consider $y \in Q \cap H \subseteq (1+\varepsilon)V$. Hence, y may be written as a convex combination of the vertices of $(1+\varepsilon)V$, which are just the vertices of V scaled by $(1+\varepsilon)$. We split this convex combination into a convex combination v_F of vertices in F and a combination $v_{\bar{F}}$ of vertices not in F. Therefore, we can write y as

$$y = (1+\varepsilon)\left(\lambda v_F + (1-\lambda) v_{\bar{F}}\right)$$

for some $\lambda \in [0,1]$. Since we already know that $\pi(v_F) \in P$, we only need to show that the impact of $v_{\bar{F}}$ is small enough. Hence, we first bound λ from below. Therefore, we consider

$$1 = \langle a, y \rangle = (1+\varepsilon)\left(\lambda + (1-\lambda)\langle a, v_{\bar{F}} \rangle\right) \leqslant (1+\varepsilon)(\lambda + (1-\lambda)(1-\varsigma)),$$

which is equivalent to $0 \leqslant \lambda\varsigma + \varepsilon\lambda\varsigma - \varsigma - \varepsilon\varsigma + \varepsilon$. In turn, this yields a bound for λ via

$$1 - \lambda \leqslant \frac{\varepsilon}{\varsigma(1+\varepsilon)}.$$

We consider

$$(1+\gamma)P = \{x \in \mathbb{R}^n : \langle c_i, x\rangle \leqslant 1 + \gamma \text{ for all } i \in [k]\} \cap \operatorname{pos}(P),$$

and verify that $\pi(y) \in (1+\gamma)P$ holds if $\pi(y) \in \operatorname{pos}(P)$, by checking all the inequalities.

$$\begin{aligned}\langle c_i, \pi(y)\rangle &= (1+\varepsilon)\left(\lambda\langle c_i, \pi(v_F)\rangle + (1-\lambda)\langle c_i, \pi(v_{\bar{F}})\rangle\right)\\ &\leqslant (1+\varepsilon)\left(\lambda + (1-\lambda)\|c_i\|\|\pi\|\|v_{\bar{F}}\|\right)\\ &\leqslant (1+\varepsilon)\left(1 + \frac{\varepsilon}{\varsigma(1+\varepsilon)} r_P^\circ \|\pi\| r_V\right)\\ &= 1 + \varepsilon\left(1 + \frac{r_P^\circ\|\pi\| r_V}{\varsigma}\right) \leqslant 1+\gamma,\end{aligned}$$

for all $i \in [k]$, which proves the claim. □

The following lemma uses the notion of a *line graph* $L(G)$ of a graph $G = (V, E)$, defined by $L(G) := (E, \{\{e, f\} \subseteq E : e \cap f \neq \varnothing\})$.

Lemma 3.23. *There exists a family of n-node graphs G with corresponding stable-set polytopes P_G such that for any polytope Q satisfying $P_G \subseteq Q \subseteq (1 + \frac{2}{1+\sqrt{8n+1}})P_G$, we have $\operatorname{xc}(Q) = 2^{\Omega(\sqrt{n})}$. Moreover, the graphs G can be chosen as line graphs of complete graphs. Therefore, there exists a set of vectors $c_1, \ldots, c_k$ with*

$$P_G = \left\{x \in \mathbb{R}^{V(G)}_{\geqslant 0} \ : \ \langle c_i, x\rangle \leqslant 1 \text{ for all } i \in [k]\right\},$$

and $\|c_i\|_\infty \leqslant 1$ for all $i \in [k]$.

Proof. Due to Braun & Pokutta [BP14], we know that the matching polytope P_M, more precisely the convex hull of all matchings in the complete graph K_n on n nodes, possesses the following lower bound. For any polytope Q satisfying $P_M \subseteq Q \subseteq \left(1 + \frac{1}{n}\right) P_M$, we have $\operatorname{xc}(Q) = 2^{\Omega(n)}$. We note that matchings in K_n directly correspond to stable-sets in the line graph $L(K_n)$, and the corresponding polytopes coincide. Note that $L(K_n)$ has $n(n-1)/2$ many nodes, which yields the claim. Furthermore, due to [Edm65a], the polytope P_M may be described using only inequalities of the claimed type. □

Proof of Theorem 3.21. Analogously to the proof of Theorem 3.2, we let $G = (V, E)$ be an n-node graph with $n \geqslant 5$ having the properties described in Lemma 3.23, and define the corresponding d-dimensional lattice Λ with $d \leqslant n + 1$ such that the stable-set polytope P_G is the image of a face F of $\mathrm{VC}(\Lambda)^\circ$ under a linear map π. We aim to apply Lemma 3.22 with $V = \mathrm{VC}(\Lambda)^\circ$ and $P = P_G$. Therefore, we claim that we can choose the parameters in Lemma 3.22 as follows.

$$r_P^\circ = \sqrt{n}, \qquad (3.7) \qquad\qquad \|\pi\| = n + m, \text{ and} \qquad (3.9)$$

$$r_V = 2, \qquad (3.8) \qquad\qquad \varsigma = \tfrac{1}{n+m}, \qquad (3.10)$$

where $|E| = m$. Moreover, corresponding to Lemma 3.23, we choose $\gamma = \frac{2}{1+\sqrt{8n+1}}$. According to Lemma 3.22, we consider any $\varepsilon > 0$ bounded via

$$\varepsilon \leqslant \frac{\frac{2}{1+\sqrt{8n+1}} \cdot \frac{1}{n+m}}{\frac{1}{n+m} + 2\sqrt{n}(n+m)}, \tag{3.11}$$

and any polytope Q with $\mathrm{VC}(\Lambda) \subseteq Q \subseteq (1+\varepsilon)\,\mathrm{VC}(\Lambda)$. This is equivalent to $\frac{1}{1+\varepsilon}\,\mathrm{VC}(\Lambda)^\circ \subseteq Q^\circ \subseteq \mathrm{VC}(\Lambda)^\circ$, which, in turn, is equivalent to $\mathrm{VC}(\Lambda)^\circ \subseteq (1+\varepsilon)Q^\circ \subseteq (1+\varepsilon)\,\mathrm{VC}(\Lambda)^\circ$. Note that we have $\mathrm{xc}(Q) = \mathrm{xc}(Q^\circ) = \mathrm{xc}((1+\varepsilon)Q^\circ)$, where the first equality follows by Lemma 3.5. With the assumptions from above, Lemma 3.22 states that

$$P_G \subseteq \pi((1+\varepsilon)Q^\circ \cap H) \cap \mathrm{pos}(P_G) \subseteq \left(1 + \frac{2}{1+\sqrt{8n+1}}\right) P_G,$$

with $H \cap \mathrm{VC}(\Lambda)^\circ = F$. Moreover,

$$\begin{aligned} \mathrm{xc}(Q) = \mathrm{xc}((1+\varepsilon)Q^\circ) &\geqslant \mathrm{xc}((1+\varepsilon)Q^\circ \cap H) \geqslant \mathrm{xc}(\pi((1+\varepsilon)Q^\circ \cap H)) \\ &\geqslant \mathrm{xc}((\pi((1+\varepsilon)Q^\circ \cap H)) \cap \mathrm{pos}(P_G)) - n, \end{aligned}$$

where the first inequality is due to Lemma 3.3, and the last inequality follows with Lemma 3.6, since $\mathrm{pos}(P_G) = \mathbb{R}^V_{\geqslant 0}$ has n facets. Due to Lemma 3.23, $\mathrm{xc}((\pi((1+\varepsilon)Q^\circ \cap H)) \cap \mathrm{pos}(P_G)) - n = 2^{\Omega(\sqrt{n})} - n$ holds, which yields $\mathrm{xc}(Q) = 2^{\Omega(\sqrt{n})}$.

Showing that $\varepsilon = 1/(192d^4)$ satisfies the bound in (3.11), proves the claim. Therefore, note that the bound in (3.11) can be shown to be at least $1/\left(12n(n+m)^2\right)$. Furthermore, the number of edges of line graphs with n nodes may be calculated via $m = \frac{n}{2}\left(1 + \sqrt{8n+1} - 2\right) \leqslant 3n\sqrt{n}$. This yields that every ε with $\varepsilon \leqslant 1/(192n^4)$ satisfies (3.11). We will see that $d = n + 1$ holds, which then yields the claim.

We are left with showing that the parameters in (3.7) – (3.10) are chosen correctly.

First, we observe that P_G has the properties stated in Lemma 3.23, and therefore it possesses a description $P_G = \left\{x \in \mathbb{R}^V_{\geqslant 0} : \langle c_i, x\rangle \leqslant 1 \text{ for } i \in [r]\right\}$ with $\max\{\|c_i\|_\infty : i \in [r]\} \leqslant 1$, which in turn proves $\max\{\|c_i\| : i \in [r]\} \leqslant \sqrt{n}$ validating (3.7).

Now, we validate the parameters regarding the dual Voronoi cell. Therefore, we first explicitly state the lattice Λ indicated in the previous section in (3.4):

$$\Lambda := \Big\{(x, x', s, s', t) \in \mathbb{Z}^{2V} \times \mathbb{Z}^{2E} \times \sqrt{n+m}\mathbb{Z} :$$
$$\sum_{i \in V}(x(i) + x'(i)) + \sum_{e \in E}(s(e) + s'(e)) + \sqrt{n+m}t = 0 \tag{3.12}$$
$$x(i) + x(j) + s(\{i,j\}) + \tfrac{1}{\sqrt{n+m}}t = 0 \text{ for } \{i,j\} \in E(G), \tag{3.13}$$
$$x + x' + \tfrac{1}{\sqrt{n+m}}t\mathbf{1} = \mathbf{0}, \tag{3.14}$$
$$s + s' + \tfrac{1}{\sqrt{n+m}}t\mathbf{1} = \mathbf{0}\Big\}. \tag{3.15}$$

We note that (3.12) is implied by (3.14) and (3.15), showing that $\dim(\Lambda) = n + 1$. According to Lemma 3.10, the dual of the Voronoi cell of Λ is described via

$$\mathrm{VC}(\Lambda)^\circ = \mathrm{conv}\left\{\frac{2}{\|z\|^2}z : z \in \Lambda\backslash\{\mathbf{0}\}\right\}.$$

Therefore, the norm of every $y \in \mathrm{VC}(\Lambda)^\circ$ is bounded by two, which validates (3.8).

Following the proof of Lemma 3.19, a face F of $\mathrm{VC}(\Lambda)^\circ$ projects onto P_G via $\pi(y) = (n+m)y_V$, where y_V equals the vector y restricted to the first n coordinates. Therefore, the operator norm is given by $\|\pi\| = n + m$, which matches (3.9).

We are left with validating the Parameter (3.10) regarding the slacks for the face F of $\mathrm{VC}(\Lambda)^\circ$. Again, following the proof of Lemma 3.19, the face F with the desired property is described via

$$F = \mathrm{conv}\left\{\tfrac{2}{\|z\|^2}z : z \in \mathrm{cl}\left(\left(\begin{smallmatrix}\mathbf{0}\\ -\sqrt{n+m}\end{smallmatrix}\right), \Lambda\right)\right\} = \mathrm{VC}(\Lambda)^\circ \cap H, \text{ with}$$
$$H = \left\{y \in \mathrm{lin}(\Lambda) : \left\langle \left(\begin{smallmatrix}\mathbf{0}\\ -\sqrt{n+m}\end{smallmatrix}\right), y\right\rangle = 1\right\}.$$

Here, the second description follows from the proof of Lemma 3.11. Let $y = \frac{2}{\|z\|^2}z$ with $z = (x, x', s, s', \sqrt{n+m}k) \in \Lambda\backslash\{\mathbf{0}\}$ be a vertex of $\mathrm{VC}(\Lambda)^\circ$ that does not lie in F. To validate (3.10), we have to show that

$$\left\langle \left(\begin{smallmatrix}\mathbf{0}\\ -\sqrt{n+m}\end{smallmatrix}\right), y\right\rangle + \frac{1}{n+m} = -\frac{2k(n+m)}{\|z\|^2} + \frac{1}{n+m} \leqslant 1 \tag{3.16}$$

holds for all such vectors y. If $k \geqslant 0$ the first summand in (3.16) becomes non-positive, and (3.16) is trivially satisfied.

Now, let us assume that $k = -1$. Since $y \notin F$, we have that $z \notin \mathrm{cl}\left(\left(\begin{smallmatrix}\mathbf{0}\\ -\sqrt{n+m}\end{smallmatrix}\right), \Lambda\right)$. Recalling (3.5) in the proof of Lemma 3.19, this shows that the vector (x, x', s, s') needs to contain entries that are not in $\{0, 1\}$. For $i \in V, e \in E$, the Constraints (3.14) & (3.15) imply $x(i), x'(i), s(e), s'(e) \notin \{0, 1\}$ whenever $x'(i), x(i), s'(e), s(e) \notin \{0, 1\}$, respectively. For a pair $x(i), x'(i)$, or $s(e), s'(e)$ being not in $\{0, 1\}$, at least one of the two entries will be strictly greater than one in absolute value. Moreover, for every $i \in V$, and $e \in E$, the Constraints (3.14) & (3.15) imply $x(i) \neq 0$ or $x'(i) \neq 0$, and $s(e) \neq 0$ or $s'(e) \neq 0$, resulting in at least $n+m$ non-zero entries. Therefore,

$$\|z\|^2 \geqslant (n+m) + 2^2 + (n+m) = 2(n+m) + 4, \text{ which implies}$$
$$-\frac{-2(n+m)}{\|z\|^2} \leqslant \frac{2(n+m)}{2(n+m)+4} = 1 - \frac{2}{n+m+2} \leqslant 1 - \frac{1}{n+m},$$

showing (3.16).

Since k is integral, we are left with the case in which $k \leqslant -2$. In this case the Equations (3.14) & (3.15) imply at least $n+m$ entries in (x, x', s, s') that are at least $-k/2$. Therefore,

$$\|z\|^2 \geqslant \frac{k^2}{4}(n+m) + k^2(n+m), \text{ which implies}$$
$$-\frac{2k(n+m)}{\|z\|^2} \leqslant \frac{-2k(n+m)}{k^2(n+m)(1+\frac{1}{4})} = \frac{-2}{k(1+\frac{1}{4})} \leqslant \frac{1}{1+\frac{1}{4}} = 1 - \frac{1}{5},$$

which is smaller than $1 - \frac{1}{n+m}$, showing (3.16). □

3.4.3 Generalization to spectrahedral lifts

A generalization of linear lifts of a polytope is the following. By $\mathcal{S}^m$, we denote the set of all symmetric, real $m \times m$ matrices. Moreover, we denote the set of all those matrices in $\mathcal{S}^m$ that are positive semidefinite (PSD), by $\mathcal{S}^m_+$. A *spectrahedron* is a set containing all vectors $x \in \mathbb{R}^n$ that fulfill conditions of the form $M(x) \in \mathcal{S}^m_+$, where $M : \mathbb{R}^n \to \mathcal{S}^m$ is an affine function. For a polytope P, the pair (Q, π), where $Q \subseteq \mathbb{R}^n$ is a spectrahedron and $\pi : \mathbb{R}^n \to \mathbb{R}^d$ is an affine map with $\pi(Q) = P$, is called a *(PSD) lift* of P. The *size* of this lift refers to the dimension of the matrix $M(x)$. For $Q = \{x \in \mathbb{R}^n : M(x) \in \mathcal{S}^m_+\}$ the size equals m. The *semidefinite extension complexity* of P, denoted by $\operatorname{sxc}(P)$, is defined as the smallest size of any of its (PSD) lifts.

Given a polyhedron $Q = \{x \in \mathbb{R}^n : \langle a_i, x\rangle \leqslant b_i \text{ for } i \in [m]\}$, we can define $M : \mathbb{R}^n \to \mathcal{S}^m$ via $M(x)_{ii} = b_i - \langle a_i, x\rangle$ for all $i \in [m]$ and $M(x)_{ij} = 0$ for $i \neq j$ and hence $Q = \{x \in \mathbb{R}^n : M(x) \in \mathcal{S}^m_+\}$. This shows that every polyhedron is a spectrahedron and therefore

$$\operatorname{sxc}(P) \leqslant \operatorname{xc}(P).$$

Hence, the upper bounds obtained in Section 3.3 also apply to the semidefinite case.

Furthermore, it is clear from the definition that for any polyhedron P and any affine map π we have that $\operatorname{sxc}(\pi(P)) \leqslant \operatorname{sxc}(P)$. Moreover, Corollary 3.4 and Lemma 3.5 analogously hold in the semidefinite case, since Yannakakis' result on the non-negative rank of a slack matrix was extended to (PSD) lifts in [Fio+12; GPT13]: The semidefinite extension complexity P equals the *PSD rank* of S, which is the smallest dimension r for which there exist PSD matrices $F_1, F_2, \ldots, F_m \in \mathcal{S}_+^r$ and $V_1, V_2, \ldots, V_n \in \mathcal{S}_+^r$ such that $S_{ij} = \langle F_i, V_j \rangle$ where the scalar product of two matrices is defined via $\langle A, B \rangle = \sum_{i,j} A_{ij} B_{ij}$.

We obtain a superpolynomial lower bound on the semidefinite extension complexity of Voronoi cells of certain lattices using a lower bound of Lee, Raghavendra, and Steurer [LRS15] on semidefinite extension complexities of correlation polytopes.

Theorem 3.24. *There exists a family of d-dimensional lattices Λ such that* $\mathrm{sxc}(\mathrm{VC}(\Lambda)) = 2^{\Omega(d^{1/13})}$.

Proof. In [LRS15] it is proven that the semidefinite extension complexity of the *correlation polytope*

$$P_n = \mathrm{conv}\left\{xx^\intercal : x \in \{0,1\}^n\right\}$$

is bounded from below by $2^{\Omega(n^{2/13})}$. Notice that

$$\begin{aligned} P_n = \mathrm{conv}\Big\{Y \in \{0,1\}^{n\times n} : Y_{ij} \leqslant Y_{ii}\ ,\ Y_{ij} \leqslant Y_{jj} \text{ and } Y_{ii} + Y_{jj} - 1 \leqslant Y_{ij}, \\ \text{for all } i, j \in [n] \text{ with } i \neq j\Big\}. \end{aligned}$$

Hence, the correlation polytope can be written as the convex hull of binary vectors $Y \in \{0,1\}^{n\times n}$ satisfying linear inequalities whose slacks only have values in $\{0,1\}$. Therefore, by Lemma 3.20 there is a lattice of dimension d where $d \leqslant n^2 + 1 = \Theta(n^2)$ such that P_n is a linear projection of a face F of $\mathrm{VC}(\Lambda)^\circ$. Analogously to the proof of Theorem 3.2 for the linear extension complexity, we conclude

$$\mathrm{sxc}(\mathrm{VC}(\Lambda)) = \mathrm{sxc}(\mathrm{VC}(\Lambda)^\circ) \geqslant \mathrm{sxc}(F) \geqslant \mathrm{sxc}(P_n) = 2^{\Omega(n^{2/13})},$$

and the claim follows since $n = \Omega(\sqrt{d})$. □

3.5 Open questions

We conclude our investigations of the extension complexity of Voronoi cells of lattices with a collection of some open problems that naturally arise from our studies and which we find interesting to pursue in future research.

In view of Theorem 3.2, a natural question is whether the logarithmic term in the lower bound $2^{\Omega(d/\log d)}$ on the extension complexity of certain Voronoi cells can be removed.

Question 3.25. *Does there exist a family of d-dimensional lattices Λ such that $\mathrm{xc}(\mathrm{VC}(\Lambda)) = 2^{\Omega(d)}$?*

We remark that our bound relies on a lower bound by Göös, Jain & Watson [GJW18] on extension complexities of stable-set polytopes, which meet the criteria of Lemma 3.20. It is known that there *exist* d-dimensional $0/1$-polytopes with extension complexity $2^{\Omega(d)}$, see [Rot13]. However, no explicit construction of such polytopes is known and so it is unclear how to transform such polytopes in order to apply Lemma 3.19 efficiently.

Comparing the superpolynomial bound in Theorem 3.2 with the polynomial upper bounds for certain classes of lattices in Section 3.3, the question arises what to expect from the extension complexity of the Voronoi cell of a "generic" lattice.

Question 3.26. *What is $\mathrm{xc}(\mathrm{VC}(\Lambda))$ for a "random" d-dimensional lattice Λ?*

Of course, this requires a suitable notion of a random lattice. Our question refers to interesting examples such as Siegel's measure [Sie45] or uniform distributions over integral lattices of a fixed determinant, see Goldstein & Mayer [GM03].

In Theorem 3.2, we have shown that *exactly* describing a Voronoi cell of a lattice may require superpolynomial-size extended formulations. We even extended the idea of the proof in Theorem 3.21 to show that $\left(1 + \frac{1}{\Theta(d^4)}\right)$-approximations may also require superpolynomial-size extended formulations. Next, it would be interesting to consider greater approximation factors, in particular in view of various results on the complexity of the approximate closest vector problem, see e.g. Aharonov & Regev [AR05].

Question 3.27. *What can be said about (semidefinite) extension complexities of α-approximations of Voronoi cells of lattices?*

Clearly, for huge factors of α, the approximation becomes trivial, in particular, in view of the John-Löwner ellipsoid that approximates centrally symmetric convex bodies within a factor of $\sqrt{d}$. Therefore, there exist α-approximations with small extension complexities for α greater than $\sqrt{d}$ in the semidefinite case and for α greater than d in the linear case. Furthermore, considering complexity results reveals ranges of α that are interesting to consider. On the one hand, approximating the closest vector problem within a factor of $\sqrt{d}$ is in coNP, see [AR05], on the other hand, approximating the closest vector problem within a factor of $d^{(1/\log\log d)}$ is NP-hard, see [DKS98]. Therefore, it would be interesting to gain a better understanding of α-approximations for α being a constant factor up to fractional powers of d.

We have seen in Theorem 3.1 that not only the root lattices but also their dual lattices have polynomial extension complexity. Is that a general phenomenon?

Question 3.28. *Given a d-dimensional lattice Λ, is there a polynomial relationship between $\mathrm{xc}(\mathrm{VC}(\Lambda))$ and $\mathrm{xc}(\mathrm{VC}(\Lambda^\star))$?*

We note that our arguments in Theorem 3.16 leading to a quadratic upper bound for zonotopal lattices are not constructive. Given the fact that the closest vector problem on such lattices can be solved in polynomial time [McC+21] one might expect that small-sized lifts of the corresponding Voronoi cells can actually be constructed explicitly.

Question 3.29. *Given a basis of a d-dimensional zonotopal lattice Λ, is it possible to construct an explicit lift of $\mathrm{VC}(\Lambda)$ with polynomially many facets in polynomial time?*

As discussed in the introduction, a small lift for the Voronoi cell of a lattice gives an expected polynomial-time algorithm for the closest vector problem. Similarly, given a *c-compact basis* $b_1, \dots, b_d$ of a lattice Λ, i.e.,

$$\mathcal{F}(\Lambda) \subseteq \left\{ \sum_{i=1}^{d} \lambda_i b_i : \lambda \in [-c, c]^d \cap \mathbb{Z}^d \right\},$$

one can adjust the algorithm of Micciancio & Voulgaris to obtain a polynomial-space algorithm for the closest vector problem with running time $(2c)^{\mathcal{O}(d)}$. This notion was introduced in [HRS20]. In Section 3.3.3, we have shown that there are lattices that do not admit c-compact bases for constant c but whose Voronoi cells have small lifts. One may ask whether the converse holds as well, or equivalently:

Question 3.30. *Given a d-dimensional lattice Λ and a c-compact basis of Λ, can $\mathrm{xc}(\mathrm{VC}(\Lambda))$ be bounded by a polynomial in d for fixed c?*

Chapter 4

Exploring the densest subgraph LP

4.1 Background

Given any undirected graph $G = (V, E)$, we aim to find a *densest subgraph* of G. In the literature different notions of density have been studied over the last decades [FR19; Lee+10]. In this chapter, we measure density simply by the ratio between the number of edges and nodes of a graph.

Problem 4.1. *Given an undirected graph $G = (V, E)$, find a non-empty subgraph H of G that maximizes $\frac{|E(H)|}{|V(H)|}$.*

Analogously, we can phrase Problem 4.1 as finding a subgraph whose average degree is maximized.

In practice, detecting dense substructures in huge graphs is of great interest. Many web systems and social networks, as well as biological data, can be represented in graphs. Therefore, Problem 4.1 has applications in various fields, for instance in computational biology for finding complex patterns in a gene annotation graph [Sah+10], or detecting regulatory DNA motifs [Fra+06]. Moreover, Problem 4.1 appears when detecting link spam in the web graph [GKT05], or within community detection in social networks [CS10]. It can be used to detect fake reviews in online marketplaces [Hoo+16]: Here users and the products they reviewed are represented by a bipartite graph. One expects that fake reviews are written by fake users that create reviews for multiple products and on the other hand huge groups of fake users are controlled by one agent and write reviews for the same products. Therefore, the set of fake users and their reviewed products form a dense subgraph.

Based on this huge amount of applications, Problem 4.1 and multiple variations have been studied extensively. Depending on the variation, the complexity of the problems varies widely. Some versions appear to be NP-hard due to their close connection to the clique problem. For instance, fixing or bounding the number of nodes in H makes the problem NP-hard, see Feige, Peleg, and Kortsarz [FPK01], or Khuller & Saha [KS09], respectively. Furthermore, using the ratio between the number of edges and the number of all possible edges as the measure of density, the problem will again become NP-hard. This and other variants are displayed in [FR19].

In contrast, Problem 4.1 can be solved in polynomial time. The first and best exact algorithms rely on network flow computations. The first algorithms are by Picard & Queyranne [PQ82], and Goldberg [Gol84], and their running time can be improved to $\mathcal{O}(|V||E|\log(|V|^2/|E|))$ using the preflow-push algorithm in the parametric network by Gallo, Grigoriadis, and Tarjan [GGT89]. Moreover, Charikar [Cha00] provides a linear programming formulation for Problem 4.1.

As already mentioned, applications of Problem 4.1 deal with huge graphs. Therefore, a lot of research is concerned with finding very fast and simple algorithms, at the cost of exact solutions. In [Cha00], Charikar presents a very intuitive greedy approach that basically removes the node with the smallest degree in each iteration. Clearly, the number of iterations of this algorithm is linear. This greedy approach first appeared in a work of Asahiro, Iwama, Tamaki, and Tokuyama [Asa+00]. Charikar proves that this algorithm is a 2-approximation. The best, known to us, ε-approximations outputting a value of at least $(1-\varepsilon)OPT$ runs in $\mathcal{O}(|E|\log|V|/\varepsilon^2)$ time, see Bahmani, Goel, and Munagala [BGM14], and relies on more general linear programming techniques.

Motivated by the fact that the simplest algorithm one can think of already yields a 2-approximation, we hope to find an exact algorithm that also uses only simple updates. Such "simple" steps might be moving from one dense subgraph to another by only changing few nodes, or updating values for nodes and edges following a simple and intuitive rule. By now, no such algorithm exist, as most exact algorithms rely on computations of maximum flows in auxiliary networks and do not reveal intermediate primal solutions.

In this chapter, we explore Charikar's polyhedral description to support the development of new algorithms. We characterize the vertices and edges of the corresponding polytope, see Section 4.3. Characterizing adjacencies in underlying polytopes of optimization problems is essential for many approaches. Because whenever we find a vertex of our feasible region whose objective value is greater than the value of all its neighbors, we are done. Moreover, following the idea of the simplex method, we may find an optimal solution via traversing the edges of the polytope. Motivated by this, Hausmann and Korte [HK78] characterized the edges of different polytopes associated with classical optimization problems such as the matching or independence polytope. One prime example of an algorithm traversing the edges of the corresponding polytope is Edmonds' Matching algorithm [Edm65a; Edm65b] following Berge's idea of using augmenting paths [Ber57]. In fact, two vertices of the matching polytope corresponding to matchings that only differ by an augmenting path are adjacent, see Chvátal [Chv75]. Moreover, also the greedy algorithm for matroids [Rad57; Edm71] follows the edges of the underlying polytope.

Obviously, all algorithms that follow the edges of the underlying polytope can only be efficient if there exist short paths between any two vertices. The *(combinatorial) diameter* of a polytope measures the length of the greatest shortest path between two vertices. Therefore, the diameter serves as a lower bound on the number of steps a method that follows edges might take to obtain an optimal solution. We compute the diameter in Section 4.3.2 and give evidence why it might be difficult to develop a simple algorithm that follows edges.

So instead of imitating the simplex procedure, we demonstrate another approach in Section 4.4. We tailor a recently published algorithm to the densest subgraph problem. This algorithm by Dadush, Hojny, Huiberts, and Weltge [Dad+22] is designed for general convex optimization problems in the separation oracle model. In theory, for general problems, the ellipsoid method runs faster than this algorithm. However, this does not have to apply for our specific problem. In [Dad+22] the authors demonstrated in various experiments that their approach works much better in practice, especially if one is only interested in approximations. Moreover, the oracle algorithm performs natural and simple update steps and is easy to implement, which makes it very applicable in practice. This fits our goals of having an algorithm that only performs simple updates on variables associated with nodes and adjacencies of the graph following a simple and rather intuitive rule.

Outline In Section 4.2, we provide Charikar's linear programming description of the densest subgraph problem, which serves as the basis for our studies. Section 4.3 is dedicated to the underlying polyhedral structure of the linear program. More precisely, in Section 4.3.1, we characterize the vertices and edges, and the bound on the diameter is proven in Section 4.3.2. Our algorithm is described in Section 4.4. We close this chapter with a discussion of open problems in Section 4.5.

4.2 Review of Charikar's formulation

This section is dedicated to the study of Charikar's LP formulation for Problem 4.1. Therefore, let us fix the undirected graph $G = (V, E)$ as the graph for which we want to solve Problem 4.1. Throughout the chapter, we assume that $|E| > 0$, to rule out trivial instances. Moreover, to simplify notation, we assume that G does not have any isolated nodes. This can be done since isolated nodes will never be present in any optimal solution. First, we provide a short revision of Charikar's work on deriving the LP formulation.

For G the densest subgraph problem reads as follows.

$$\begin{aligned}
\max \quad & \frac{\sum_{e \in E} y(e)}{\sum_{v \in V} x(v)} \\
\text{s.t.} \quad & y(e) - x(v) \leqslant 0 \qquad \forall e \in E, v \in V \text{ with } v \in e, \\
& x(v), y(e) \in \{0,1\} \quad \forall v \in V, e \in E.
\end{aligned} \tag{4.1}$$

We note that the linear constraints in (4.1) form a totally unimodular system. Relaxing the binary conditions and transforming the fractional problem into a linear one yields

$$\begin{aligned}
\max \quad & \sum_{e \in E} y(e) \\
\text{s.t.} \quad & \sum_{v \in V} x(v) = 1, \\
& y(e) - x(v) \leqslant 0 \quad \forall e \in E, v \in V \text{ with } v \in e, \\
& x(v), y(e) \geqslant 0 \quad \forall v \in V, e \in E.
\end{aligned} \tag{4.2}$$

The transformation to the linear problem is achieved by first introducing a new variable $z = 1/\sum_{v \in V} x(v)$ and afterwards substituting $zx(v)$ and $zy(e)$ by $x(v)$ and $y(e)$, respectively for $e \in E, v \in V$. It actually holds true that the optimal values of (4.1) & (4.2) coincide for every graph G, which is formally proven in [Cha00]. Moreover, this fact follows also from total unimodularity using some arguments that are displayed in the next section to determine the vertices of the feasible region.

Within the following Section 4.3, we will stick to the above linear programming description. However, we can also relax the first constraint to $\sum_{v \in V} x(v) \leqslant 1$. With this relaxed constraint, the feasible region gains exactly one additional vertex, namely $\mathbf{0}$, and therefore, the optimal objective value will not change. Moreover, $x(v) \geqslant 0$ is implied by $y(e) \geqslant 0$ via the constraint $y(e) - x(v) \leqslant 0$ for $e \in E, v \in V$ with $v \in e$. Therefore, the non-negativity constraints on the x variables are redundant. On the other hand, we might drop the non-negativity constraints on the y variables and stick to the constraints on the x variables. That will not change the objective value, since we are maximizing $\sum_{e \in E} y(e)$, and assigning a negative value to some $y(e)$ does not affect the choice on the other variables, since the only constraints regarding $y(e)$ relate this variable to non-negative x variables. So in any optimal solution, the values assigned to the edges of the graph will be non-negative. In Section 4.4, we need inequalities instead of equations and to simplify notation, as few of these inequalities as possible. Therefore, we use a modified LP in Section 4.4.

4.3 Polyhedral description and properties

Knowing Charikar's LP, we can characterize the vertices and edges of its feasible region. We define P_{dense} to be the feasible region of the densest subgraph LP, i.e.,

$$\begin{aligned} P_{\text{dense}} := \Big\{(x,y) \in \mathbb{R}^V \times \mathbb{R}^E : \sum_{v \in V} x(v) = 1, \\ y(e) - x(v) \leqslant 0 \quad & \text{for all } e \in E, v \in V \text{ with } v \in e, \\ y(e) \geqslant 0 \quad & \text{for all } e \in E\Big\}. \end{aligned}$$

We note that dropping the $\sum_{v \in V} x(v) = 1$ constraint, which just "scales all subgraphs", and (re-)inserting an upper bound on the variables, we end up with the following polytope P_{subgraph}.

$$P_{\text{subgraph}} := \left\{(x,y) \in [0,1]^V \times [0,1]^E : y(e) - x(v) \leqslant 0 \text{ for } e \in E, v \in V \text{ with } v \in e\right\}.$$

Clearly, every vertex of P_{subgraph} is the characteristic vector of a subgraph of G, since the polytope is described by a totally unimodular system. With these observations, the stage is set for determining the vertices of P_{dense}.

4.3.1 Vertices and edges

We note that P_{dense} results from P_{subgraph} by cutting with a hyperplane, i.e.,

$$P_{\text{dense}} = P_{\text{subgraph}} \cap \left\{(x,y) \in \mathbb{R}^V \times \mathbb{R}^E : \sum_{v \in V} x(v) = 1\right\}.$$

Therefore, a vertex of P_{dense} is either a vertex of P_{subgraph} or lies on an edge of P_{subgraph}. To satisfy $\sum_{v \in V} x(v) = 1$, these vertices need to lie on an edge between $\mathbf{0}$ and some other characteristic vector of a subgraph. For this reason, every vertex (x^*, y^*) of P_{dense} is of the following form.

$$x^*(v) = \begin{cases} \frac{1}{|V(H)|} & \text{if } v \in V(H) \\ 0 & \text{if } v \notin V(H) \end{cases} \quad \text{and } y^*(e) = \begin{cases} \frac{1}{|V(H)|} & \text{if } e \in E(H) \\ 0 & \text{if } e \notin E(H) \end{cases} \quad \text{all } e \in E, v \in V,$$

for some non-empty subgraph H of G. We use characteristic vectors in order to address points of the above form. For any non-empty graph H, we define the characteristic vector of H via $\chi_H := \left(\chi_{V(H)}, \chi_{E(H)}\right) \in \{0,1\}^V \times \{0,1\}^E$ having an entry of 1 for every $v \in V(H), e \in E(H)$ and 0 else. Using this notation, we can simply write $(x^*, y^*) = \frac{1}{|V(H)|}\chi_H$.

Theorem 4.2. *A point (x^*, y^*) is a vertex of P_{dense} if and only if $(x^*, y^*) = \frac{1}{|V(H)|}\chi_H$ for a non-empty connected subgraph H of G.*

Proof. We already know that the set of vertices $\mathcal{V}(P_{\text{dense}})$ of P_{dense} satisfies

$$\mathcal{V}(P_{\text{dense}}) \subseteq \left\{ \frac{1}{|V(H)|}\chi_H : H \text{ is a non-empty subgraph of } G \right\}.$$

Let H be a non-empty connected subgraph of G. Corresponding to H, we define the following objective vector $c \in \mathbb{R}^V \times \mathbb{R}^E$.

$$c(v) = \begin{cases} -\deg_H(v) & \text{if } v \in V(H) \\ -1 & \text{if } v \notin V(H) \end{cases} \text{ and } c(e) = \begin{cases} 2 & \text{if } e \in E(H) \\ -1 & \text{if } e \notin E(H) \end{cases} \text{ for all } e \in E, v \in V.$$

We claim that $\frac{1}{|V(H)|}\chi_H$ is the unique optimizer of $\max\{c^\mathsf{T} z : z \in P_{\text{dense}}\}$, and therefore, a vertex of P_{dense}. Clearly, $\frac{1}{|V(H)|}\chi_H$ is feasible, and $c^\mathsf{T}\frac{1}{|V(H)|}\chi_H = 0$. For any non-empty subgraph H', we have

$$\begin{aligned} c^\mathsf{T}\frac{1}{|V(H')|}\chi_{H'} = -\frac{1}{|V(H')|}\Bigg(&|E(H')\backslash E(H)| + |V(H')\backslash V(H)| \\ &+ \sum_{v \in V(H') \cap V(H)} \left|\left\{e \in E(H)\backslash E(H') : v \in e\right\}\right| \Bigg), \end{aligned}$$

which is non-positive.

Let us assume that $c^\mathsf{T}\frac{1}{|V(H')|}\chi_{H'} = 0$ holds. Hence, each summand has to be zero. The first two being zero implies $E(H')\backslash E(H) = \varnothing$, meaning $E(H') \subseteq E(H)$, and $V(H')\backslash V(H) = \varnothing$, meaning $V(H') \subseteq V(H)$, which in addition implies $V(H') \cap V(H) \neq \varnothing$. The last summand yields that $\{e \in E(H)\backslash E(H') : v \in e\} = \varnothing$ holds for all $v \in V(H') \cap V(H)$. Together with $E(H') \subseteq E(H)$, this implies that a node $v \in V(H') \cap V(H)$ is incident to the same set of edges in H and H'. Therefore, no edge in H connects a node $v \in V(H') \cap V(H) \neq \varnothing$ and a node $w \in V(H)\backslash V(H')$. Since H is connected, $V(H)\backslash V(H') = \varnothing$ must hold. So, $H = H'$ holds proving that $\frac{1}{|V(H)|}\chi_H$ is actually the unique maximizer.

Now, we consider any non-empty subgraph H of G that is not connected. For two non-empty subgraphs H_1, H_2 of H, where H_1 is a connected component of H and H_2 is the subgraph induced by $V(H)\backslash V(H_1)$, we have

$$\begin{aligned} \frac{\chi_H}{|V(H)|} &= \frac{\chi_{H_1} + \chi_{H_2}}{|V(H_1)| + |V(H_2)|} \\ &= \frac{|V(H_1)|}{|V(H_1)| + |V(H_2)|} \cdot \frac{\chi_{H_1}}{|V(H_1)|} + \frac{|V(H_2)|}{|V(H_1)| + |V(H_2)|} \cdot \frac{\chi_{H_2}}{|V(H_2)|} \\ &= \frac{|V(H_1)|}{|V(H_1)| + |V(H_2)|} \cdot \frac{\chi_{H_1}}{|V(H_1)|} + \left(1 - \frac{|V(H_1)|}{|V(H_1)| + |V(H_2)|}\right) \frac{\chi_{H_2}}{|V(H_2)|}, \end{aligned}$$

proving that $\frac{1}{|V(H)|}\chi_H$ is not a vertex of P_{dense}, since it is a non-trivial convex combination of $\frac{1}{|V(H_1)|}\chi_{H_1}$ and $\frac{1}{|V(H_2)|}\chi_{H_2}$. □

The following theorem characterizes the edges of P_{dense}. It basically tells that two vertices corresponding to subgraphs are connected by an edge whenever the subgraphs are either disjoint or one can be obtained from the other via adding/deleting some connected part. In order to formalize this, we use some shorthand notation. For any graph G, node set $S \subseteq V(G)$, and edge set $F \subseteq E(G)$, we use $G[S]$ for the induced subgraph with node set S containing all edges in $E(G)$ that have both endpoints in S, and $G - F$ for the graph $(V(G), E(G)\backslash F)$.

Theorem 4.3. *Two vertices $\frac{1}{|V(H_1)|}\chi_{H_1}$ and $\frac{1}{|V(H_2)|}\chi_{H_2}$ corresponding to different non-empty connected subgraphs H_1, H_2 with $|E(H_1)| \leqslant |E(H_2)|$ share an edge in P_{dense} if and only if one of the following applies.*

P1) The subgraphs are disjoint, i.e., $V(H_1) \cap V(H_2) = \varnothing$, or

P2) H_1 is a subgraph of H_2 and

- *a) H_1 is obtained from H_2 via deleting a non-empty connected subgraph from H_2. Formally, $E(H_1) \subseteq E(H_2)$, $H_2[V(H_2)\backslash V(H_1)]$ is connected, and every $e \in E(H_2)\backslash E(H_1)$ is incident to some $v \in V(H_2)\backslash V(H_1)$, or*
- *b) H_1 is obtained by H_2 via deleting one edge, i.e., $V(H_1) = V(H_2)$, $E(H_1) \subseteq E(H_2)$, and $|E(H_2)\backslash E(H_1)| = 1$.*

Proof. Let H_1, H_2 be two distinct non-empty connected subgraphs of G with $|E(H_1)| \leqslant |E(H_2)|$, such that they satisfy one of the properties stated in the theorem. Corresponding to these subgraphs H_1, H_2, we define the following objective vector $c \in \mathbb{R}^V \times \mathbb{R}^E$.

$$c(v) = \begin{cases} -1 & \text{if } v \notin V(H_1) \cup V(H_2) \\ -\deg_{H_1}(v) & \text{if } v \in V(H_1) \\ -\deg_{H_2}(v) & \text{if } v \in V(H_2) \backslash V(H_1) \end{cases} \qquad \text{for all } v \in V, \text{ and}$$

$$c(e) = \begin{cases} -1 & \text{if } e \notin E(H_1) \cup E(H_2) \\ 2 & \text{if } e \in E(H_1) \\ |\{e \cap V(H_2) \backslash V(H_1)\}| & \text{if } e \in E(H_2) \backslash E(H_1) \end{cases} \qquad \text{for all } e \in E.$$

We may easily calculate that $c^\intercal \frac{1}{|V(H_1)|}\chi_{H_1} = 0$ and $c^\intercal \frac{1}{|V(H_2)|}\chi_{H_2} = 0$. For any non-empty connected subgraph H of G, we calculate

$$c^\intercal \frac{1}{|V(H)|}\chi_H = -\frac{1}{|V(H)|}\Bigg(|V(H)\backslash (V(H_1) \cup V(H_2))| + |E(H)\backslash (E(H_1) \cup E(H_2))| + \sum_{v\in V(H)\cap(H_1)} |\{e \in E(H_1)\backslash E(H) : v \in e\}| + \sum_{v\in V(H)\cap V(H_2)\backslash V(H_1)} |\{e \in E(H_2)\backslash E(H) : v \in e\}| \Bigg),$$

which is non-positive.

We assume that $c^\intercal \frac{1}{|V(H)|}\chi_H = 0$ holds. Considering the first two summands, we conclude that $V(H) \subseteq V(H_1) \cup V(H_2)$ and $E(H) \subseteq E(H_1) \cup E(H_2)$. Therefore, the first or the second sum is not empty.

First, suppose that the second sum is empty, meaning that $V(H) \cap V(H_2)\backslash V(H_1) = \emptyset$, or equivalently $V(H) \subseteq V(H_1)$, holds. In this case, the first sum is non-empty, and for every $v \in V(H_1) \cap V(H) \neq \emptyset$ all edges $e \in E(H_1)$ incident to v satisfy $e \in E(H)$. Since H_1 is connected, this implies $V(H_1) \subseteq V(H)$ and $E(H_1) \subseteq E(H)$. If $E(H) \subseteq E(H_1)$, we have $H = H_1$. Otherwise, $E(H)$ contains an additional edge from $E(H_2)$ connecting two nodes of $V(H_1)$, which is only possible in case P2)b). This yields $H = H_2$.

Now, suppose that the second sum is non-empty, meaning that $V(H)\cap V(H_2)\backslash V(H_1) \neq \emptyset$ holds. Therefore, $V(H_1) \neq V(H_2)$ holds, and H_1, H_2 either satisfy Property P1), or P2)a). Using that $H_2[V(H_2)\backslash V(H_1)]$ is connected and every edge in $E(H_2)\backslash E(H_1)$ is incident to a node in $V(H_2)\backslash V(H_1)$, we can infer $V(H_2)\backslash V(H_1) \subseteq V(H)$ and $E(H_2)\backslash E(H_1) \subseteq E(H)$ analogously to the arguments displayed above for a non-empty first sum. If H_1 and H_2 are disjoint, $H = H_2$, since H is connected. If H_1, H_2 satisfy P2)a), $E(H_2)\backslash E(H_1) \subseteq E(H)$ implies $V(H_1) \cap V(H) \neq \emptyset$ yielding a non-empty first sum. With the above argumentation, we obtain $V(H_1) \subseteq V(H)$ and $E(H_1) \subseteq E(H)$ and hence $H = H_2$ holds.

Therefore, no vertex of P_{dense} distinct from $\frac{1}{|V(H_1)|}\chi_{H_1}$ and $\frac{1}{|V(H_2)|}\chi_{H_2}$ is a maximizer for c. This proves that they actually share an edge.

On the other hand, let H_1, H_2 be distinct connected subgraphs of G with $|E(H_1)| \leqslant |E(H_2)|$, not satisfying one of the properties of the theorem. Therefore, $V(H_1)\cap V(H_2) \neq \emptyset$. Moreover, if H_1 is a subgraph of H_2 the subgraphs have to differ by more than one edge or one connected subgraph.

We will consider several cases, in each following the same idea. We provide two additional non-empty subgraphs H' and H'', such that at least one of the vectors $\frac{1}{|V(H')|}\chi_{H'}$, $\frac{1}{|V(H'')|}\chi_{H''}$ cannot lie on the line segment between $\frac{1}{|V(H_1)|}\chi_{H_1}$ and $\frac{1}{|V(H_2)|}\chi_{H_2}$.

Moreover, we choose the subgraphs in such a way that they satisfy

$$\chi_{H_1} + \chi_{H_2} = \chi_{H'} + \chi_{H''}. \tag{4.3}$$

This yields

$$\begin{aligned}
&\frac{|V(H_1)|}{|V(H_1)| + |V(H_2)|} \cdot \frac{\chi_{H_1}}{|V(H_1)|} + \left(1 - \frac{|V(H_1)|}{|V(H_1)| + |V(H_2)|}\right) \frac{\chi_{H_2}}{|V(H_2)|} \\
&= \frac{|V(H_1)|}{|V(H_1)| + |V(H_2)|} \cdot \frac{\chi_{H_1}}{|V(H_1)|} + \frac{|V(H_2)|}{|V(H_1)| + |V(H_2)|} \cdot \frac{\chi_{H_2}}{|V(H_2)|} \\
&= \frac{\chi_{H_1} + \chi_{H_2}}{|V(H_1)| + |V(H_2)|} \\
&= \frac{\chi_{H'} + \chi_{H''}}{|V(H')| + |V(H'')|} \\
&= \frac{|V(H')|}{|V(H')| + |V(H'')|} \cdot \frac{\chi_{H'}}{|V(H')|} + \left(1 - \frac{|V(H'|)}{|V(H')| + |V(H'')|}\right) \frac{\chi_{H''}}{|V(H'')|},
\end{aligned}$$

which shows that an inner point of the line segment between $\frac{1}{|V(H_1)|}\chi_{H_1}$ and $\frac{1}{|V(H_2)|}\chi_{H_2}$ is a non-trivial convex combination of the feasible points $\frac{1}{|V(H')|}\chi_{H'}$ and $\frac{1}{|V(H'')|}\chi_{H''}$, which proves that $\frac{1}{|V(H_1)|}\chi_{H_1}$ and $\frac{1}{|V(H_2)|}\chi_{H_2}$ do not share an edge in P_{dense}.

Let us start with the case in which $V(H_1) \cap V(H_2) \neq \varnothing$ but H_1 is not contained in H_2. In this case, we define the two non-empty subgraphs $H' := (V(H_1) \cap V(H_2), E(H_1) \cap E(H_2))$ and $H'' := (V(H_1) \cup V(H_2), E(H_1) \cup E(H_2))$ of G.

Now, we assume that H_1 is a subgraph of H_2. Let $\bar{E}$ denote the set of edges in $E(H_2) \backslash E(H_1)$ that are not incident to a node in $V(H_2) \backslash V(H_1)$.

If $H_2[V(H_2) \backslash V(H_1)]$ is empty, meaning that $V(H_2) = V(H_1)$ holds, we have $\bar{E} = E(H_2) \backslash E(H_1) = \{e_1, \ldots, e_k\}$ with $k > 1$. In this case, we define $H' := (V(H_1), E(H_1) \cup \{e_1\})$ and $H'' := (V(H_1), E(H_1) \cup \{e_2, \ldots, e_k\})$.

If $H_2[V(H_2) \backslash V(H_1)]$ is non-empty and connected, the set $\bar{E}$ is not empty. So, we define $H' := H_2[V(H_1)]$ and $H'' := (V(H_2), E(H_2) \backslash \bar{E})$.

If $H_2[V(H_2) \backslash V(H_1)]$ is not connected, it has connected components $C_1, \ldots, C_k$ with $k > 1$. Here, $\bar{E}$ might be empty. We note that, each connected component is attached to H_1 by an edge, since H_2 is connected. So, we define the non-empty connected graphs $H' := H_2[V(H_1) \cup V(C_1)] - \bar{E}$ and $H'' := H_2[V(H_1) \cup \bigcup_{i=2}^{k} V(C_i)]$.

We note that all these graphs H', H'' are non-empty and distinct from H_1, H_2. Moreover, H'' is connected in all cases. Therefore, $\frac{1}{|V(H'')|}\chi_{H''}$ is a vertex in P_{dense} distinct from $\frac{1}{|V(H_1)|}\chi_{H_1}, \frac{1}{|V(H_2)|}\chi_{H_2}$, showing that it does not lie on the line segment between $\frac{1}{|V(H_1)|}\chi_{H_1}$ and $\frac{1}{|V(H_2)|}\chi_{H_2}$. Furthermore, in all cases H', H'' satisfy (4.3), which proves the claim. □

4.3.2 Diameter

Using the information about adjacency in P_{dense}, we show that the diameter of this polytope is very small.

Theorem 4.4. *The diameter of the polytope P_{dense} is at most 3 for every graph G and there exist graphs for which the diameter actually equals 3.*

Proof. Given two vertices of P_{dense} corresponding to the connected subgraphs H_1 and H_2 of G, there exist nodes $v_1 \in V(H_1)$ and $v_2 \in V(H_2)$ such that $H_1[V(H_1)\setminus\{v_1\}]$ and $H_2[V(H_2)\setminus\{v_2\}]$ are connected. (Choose v_i to be a leaf in a spanning tree of H_i for $i \in \{1,2\}$.) Due to Theorem 4.3, the vertices corresponding to $(\{v_i\}, \varnothing)$ and H_i share an edge for $i \in \{1,2\}$, as well as the vertices corresponding to $(\{v_1\}, \varnothing)$ and $(\{v_2\}, \varnothing)$ share an edge if they are not equal. This proves that the diameter is at most three.

For the graph in Figure 4.1, the diameter of P_{dense} actually equals three, which we easily see by considering the two vertices corresponding to the depicted subgraphs.

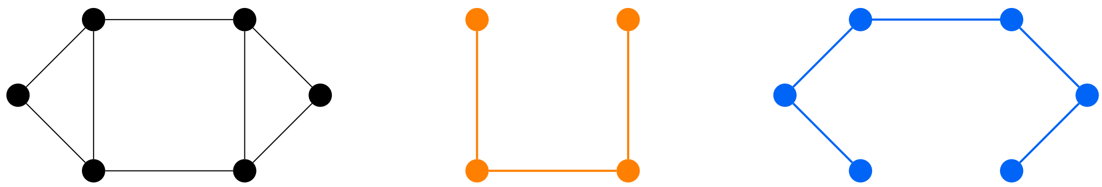


Figure 4.1: Graph $G = (V, E)$ and subgraphs, one depicted in blue and one in orange such that the corresponding vertices have distance 3 in P_{dense}.

□

Despite having small diameter, developing an easy and efficient algorithm that follows edges does not seem to be straight forward. Clearly, we like to follow only edges along which the objective value increases. Obviously, the described paths of length three do not satisfy this property. Furthermore, the naive approach of only following improving edges that correspond to solely adding or deleting one node (with all incident edges) will not always work out, see Figure 4.2.

Furthermore, when we mimic the simplex algorithm, we have to deal with highly degenerated vertices of P_{dense}. To us, it seems as if simply following edges won't lead to an efficient algorithm, although the diameter is just a constant. This phenomenon is not unusual. The diameter of the travelling salesperson polytope is bounded by a constant [PR74; RC98] and the cut polytope (of the complete graph) has a diameter of one [BM86], but optimization over these polytopes is NP-hard [Kar72; GJ79].

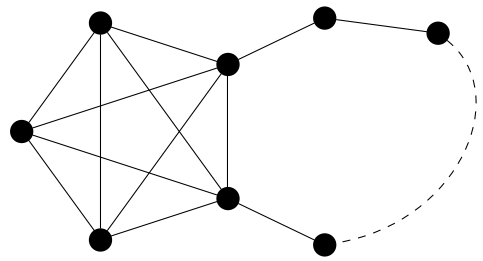


Figure 4.2: Here K_5 (and not the whole graph) is optimal, but removing only one node from the whole graph results in a subgraph having smaller density than the whole graph. Therefore, this simple update is not improving.

4.4 Algorithm

As discussed in the previous section, following the edges of P_{dense} does not directly yield an efficient algorithm. Nevertheless, we still can use Charikar's LP formulation to develop an algorithm. Our algorithm is based on the following LP model.

$$\begin{aligned} \max \quad & \sum_{e \in E} y(e) \\ \text{s.t.} \quad & \sum_{v \in V} x(v) \leqslant 1, \\ & y(e) - x(v) \leqslant 0 \quad \forall e \in E, v \in V \text{ with } v \in e, \\ & -x(v) \leqslant 0 \quad \forall v \in V. \end{aligned} \tag{4.4}$$

As discussed, in an optimal solution of (4.4) all values assigned to the edges of the graph will be non-negative. Moreover, there is always an optimal vertex solution (x^*, y^*) for (4.4) satisfying $\sum_{v \in V} x^*(v) = 1$. Therefore, the above formulation is equivalent to Charikar's LP formulation (4.2) for the densest subgraph problem.

The dual of this problem reads as follows.

$$\begin{aligned} \min \quad & a \\ \text{s.t.} \quad & \sum_{e \in E, v \in e} b(e, v) \leqslant a \quad \forall v \in V, \\ & b(\{v,w\}, v) + b(\{v,w\}, w) = 1 \quad \forall \{v,w\} \in E, \\ & a \geqslant 0, \\ & b(e, v) \geqslant 0 \quad \forall e \in E, v \in V \text{ with } v \in e. \end{aligned} \tag{4.5}$$

Provided these LPs one can apply standard linear programming techniques such as the ellipsoid method. However, this method is not simple, and it also does not perform well in practice. The authors of [Dad+22] tackled exactly that issue and provide a method that is much easier to implement. Their algorithm serves as an approximation algorithm for convex optimization problems whose feasible region is given via a separation oracle. The algorithm and the proofs displayed in this section are based on this work. Since in our case the description of the feasible region is fully accessible, we do not have to deal with adding new constraints. Moreover, we are only dealing with a linear optimization problem instead of general convex optimization. This gives us the ability to provide the proofs and the algorithm in a much more direct way.

Our displayed approach crucially relies on the interplay between primal and dual solutions. We will use some kind of dual multipliers to assign values on the nodes and edges of our graph. These node and edge values will provide intermediate primal solutions. During the algorithm, we iteratively update the multipliers and therefore, the node and edge values. Moreover, we use a collection of linear expressions telling us which dual multiplier needs to be increased and when we reach optimality.

To define our multipliers, we consider an index set $I := \{0,1\} \cup \{(e,v) : e \in E, v \in e\} \cup V$, where 0 is associated with the objective of (4.4) and the indices $1, (e,v)$ for $e \in E, v \in e$, and v for $v \in V$ correspond to the inequalities of (4.4). Moreover, for $\lambda \in \mathbb{R}^I_{\geqslant 0}$, we define node and edge values $(p,q) \in \mathbb{R}^V \times \mathbb{R}^E$ via

$$p(v) := \sum_{e \in E, v \in e} \lambda(e,v) + \lambda(v) - \lambda(1), \text{ for all } v \in V$$
$$q(\{v,w\}) := \lambda(0) - \lambda(\{v,w\},v) - \lambda(\{v,w\},w), \text{ for all } \{v,w\} \in E.$$

Furthermore, let $LB \in \mathbb{R}$ be any lower bound to the optimal value of the densest subgraph problem. We consider a vector $R \in \mathbb{R}^I$, whose entries are computed via the following linear expressions that will serve as an indicator for optimality.

$$\begin{aligned}
R(0) &:= -\sum_{e \in E} q(e) + (LB(\lambda(1) - \lambda(0)LB)) \\
R(1) &:= \sum_{v \in V} p(v) - (\lambda(1) - \lambda(0)LB)\,, \\
R(e,v) &:= q(e) - p(v) && \forall e \in E, v \in V \text{ with } v \in e, \\
R(v) &:= -p(v) && \forall v \in V.
\end{aligned}$$

Now, the following two lemmas show the interplay between the signs of R and the lower bound.

Lemma 4.5. *There exist $\lambda \in \mathbb{R}^I_{\geqslant 0}$ with $\lambda(0) > 0$ such that $R = \mathbf{0}$ if and only if LB is the optimal value of the densest subgraph problem.*

Proof. If there are multipliers $\lambda \in \mathbb{R}^{I}_{\geqslant 0}$ with $\lambda(0) > 0$ satisfying $R = \mathbf{0}$, we obtain $0 = p(v) = q(e) = \lambda(1) - \lambda(0)LB$ for all $e \in E, v \in V$. Since $\lambda(0) > 0$ holds, we have $LB = \lambda(1)/\lambda(0)$. We may define a dual solution via $a^* := LB$ and $b^*(e,v) := \lambda(e,v)/\lambda(0) \geqslant 0$ for all $e \in E, v \in e$. To check feasibility, we observe that $p(v) = 0$ and $\lambda(v) \geqslant 0$ imply $\sum_{e \in E, v \in e} b^*(e,v) \leqslant a^*$ for all $v \in V$, and $q(\{v,w\}) = 0$ implies $b^*(\{v,w\},v) + b^*(\{v,w\},w) = 1$ for all $\{v,w\} \in E$. Therefore, (a^*, b^*) is feasible for the dual LP, and attains a value of LB, proving optimality by using weak duality.

On the other hand, we suppose that LB is optimal. Due to duality, there exist a^*, b^* that are feasible for the dual LP and attain a value of $LB = a^*$. We set $\lambda(1) := a^*$, $\lambda(e,v) := b^*(e,v)$ for all $e \in E, v \in e$, $\lambda(v) := a^* - \sum_{e \in E, v \in e} b^*(e,v) \geqslant 0$ for all $v \in V$, and $\lambda(0) := 1$. We obtain $p(v) = 0$ and $q(e) = 0$ for all $v \in V, e \in E$ and $\lambda(1) - \lambda(0)LB = 0$. This implies $R = \mathbf{0}$. ☐

Lemma 4.6. *If $R < \mathbf{0}$, then $\frac{1}{\lambda(1) - \lambda(0)LB}(p,q)$ is a feasible solution for (4.4) with an objective value strictly greater than LB.*

Proof. First, we observe that $\lambda(1) - \lambda(0)LB > 0$, if $R < \mathbf{0}$. Since $R(0) < 0$ holds, we have $\sum_{e \in E} \frac{1}{\lambda(1) - \lambda(0)LB} q(e) > LB$ proving a strictly greater objective value. Moreover, $R(1) < 0$ implies $\sum_{v \in V} \frac{1}{\lambda(1) - \lambda(0)LB} p(v) \leqslant 1$, $R(e,v) < 0$ implies $\frac{1}{\lambda(1) - \lambda(0)LB} q(e) - \frac{1}{\lambda(1) - \lambda(0)LB} p(v) \leqslant 0$ for all $e \in E, v \in e$, and $R(v) < 0$ implies $\frac{1}{\lambda(1) - \lambda(0)LB} p(v) \geqslant 0$ for all $v \in V$, which shows that $\frac{1}{\lambda(1) - \lambda(0)LB}(p,q)$ is feasible. ☐

Bearing Lemma 4.5 and Lemma 4.6 in mind, we design an algorithm that manipulates λ to decrease R whenever R has positive entries. We note that increasing $\lambda(i)$, decreases $R(i)$ for $i \in I$. Whenever $R < \mathbf{0}$ holds, the algorithm updates the lower bound in accordance to Lemma 4.6. This results in the better lower bound $\frac{\sum_{e \in E} q(e)}{\lambda(1) - \lambda(0)LB}$. Since solely updating LB might increase $R(1)$, we also increase $\lambda(1), \lambda(v)$ for $v \in V$ in such a way that it evens out the update of LB. The algorithm terminates whenever $R = \mathbf{0}$. Unfortunately, we cannot guarantee that we reach this state. Nevertheless, we will quantify the approximation error whenever the entries of R are close to zero. Obviously, this is only meaningful if λ is in some sense bounded away from zero. Therefore, we normalize λ. We note that normalizing λ results in scaling R by a factor of at most 1 if we start with a normalized λ. Hence, this step will not increase a positive R. Algorithm 1 follows exactly these ideas.

This algorithm serves as a conceptually simple framework that leaves several degrees of freedom to obtain optimized implementations. Within this section, we show that there is a way to specify these steps such that the framework becomes an algorithm running in polynomial time. Thereby, to keep the analysis simple, we do not care about achieving the smallest possible degree of the polynomial.

Algorithm 1 Approximating density

initialize LB, λ
while $R \neq \mathbf{0}$ **do**
  if $R(i) \geqslant 0$ for some $i \in I$ **then** increase $\lambda(i)$
  else
    $LB' \leftarrow \frac{\sum_{e \in E} q(e)}{\lambda(1) - \lambda(0) LB}$
    $\lambda(1) \leftarrow \lambda(1) + \lambda(0)(LB' - LB)$
    $\lambda(v) \leftarrow \lambda(v) + \lambda(0)(LB' - LB)$ for all $v \in V$
    $LB \leftarrow LB'$
  end if
  normalize λ
end while
return LB

For our analysis, we assume the following initialization. We start with the trivial lower bound, namely the density of the whole graph $\frac{|E|}{|V|}$. Moreover, we need non-negative starting values for λ ensuring $\lambda(0) > 0$. Therefore, we start with $\lambda(0) = 1$, and $\lambda(i) = 0$ for $i \in I \backslash \{0\}$. We note that during the algorithm, we either increase a variable or normalize all of them, which maintains positivity. Therefore, $\lambda(0) > 0$ and $\lambda(i) \geqslant 0$ for $i \in I$ holds at any state during the algorithm.

Another degree of freedom is the value by which $\lambda(i)$ is increased. We choose that value to be the ℓ defined in Lemma 4.7 below. In the original algorithm in [Dad+22] a different step length is chosen to obtain a better running time. Furthermore, normalizing λ can be done in different ways. We choose to normalize with respect to the ℓ_1-norm, which means that the entries of λ serve as convex coefficients satisfying

$$\lambda(0) + \lambda(1) + \sum_{e \in E, v \in e} \lambda(e, v) + \sum_{v \in V} \lambda(v) = 1. \tag{4.6}$$

To analyze the algorithm the following notations will come in very handy. We define

$$z_{\lambda, LB} := \begin{pmatrix} -p \\ -q \\ \lambda(1) - \lambda(0) LB \end{pmatrix}.$$

For simpler notation, we will often only use z in cases where the dependency on the variables is clear. We note that $R = \mathbf{0}$ holds if and only if $z = \mathbf{0}$. So, instead of measuring the distance of R to $\mathbf{0}$, we will consider the Euclidean norm of z. Furthermore, we define

$$r_0 := \begin{pmatrix} \mathbf{0}_V \\ -\mathbf{1}_E \\ -LB \end{pmatrix}, r_1 := \begin{pmatrix} \mathbf{1}_V \\ \mathbf{0}_E \\ 1 \end{pmatrix}, r_{e,v} := \begin{pmatrix} -\chi_v \\ \chi_e \\ 0 \end{pmatrix} \text{ for } e \in E, v \in e, \text{ and}$$
$$r_v := \begin{pmatrix} -\chi_v \\ \mathbf{0}_E \\ 0 \end{pmatrix} \text{ for } v \in V,$$

where $\chi_v \in \{0,1\}^V, \chi_e \in \{0,1\}^E$ denote the standard unit vectors. We note that $z = \sum_{i \in I} \lambda(i) r_i$, and $\langle r_i, z \rangle = -R(i)$ for $i \in I$.

A closer look at the vectors r_i for $i \in I \backslash \{0\}$ reveals that they are actually the coefficient vectors of the linear inequalities of the LP (4.4) with appended right-hand side, and the first coordinates of $-r_0$ coincide with the objective vector of that LP. This observation serves another interpretation of the algorithm. We aim to find coefficients $\lambda(i)$ such that combining the inequalities of the LP with these coefficients proves $\sum_{e \in E} y(e) \leqslant LB$.

The following Lemma 4.7 specifies the increase of $\lambda(i)$ for $i \in I$ in the algorithm and provides first indications towards the convergence rate.

Lemma 4.7. *Let λ, LB and the corresponding vector z be given such that $\|z\|^2 < 16|V|^2$ and $R(j) \geqslant 0$ holds for some $j \in I$. We define*

$$\ell := \frac{\|z\|^2}{16|V|^2 - \|z\|^2}$$

and

$$\lambda'(j) := \frac{\lambda(j) + \ell}{\ell + 1}, \quad \lambda'(i) := \frac{\lambda(i)}{\ell + 1} \quad \text{for all } i \in I \backslash \{j\}.$$

Then, the following holds

$$\left\|z_{\lambda',LB}\right\|^2 \leqslant \|z_{\lambda,LB}\|^2 - \frac{1}{16|V|^2} \|z_{\lambda,LB}\|^4.$$

Note that if λ satisfies (4.6), the same holds true for λ'.

Proof. Since z is linear in each component of λ,

$$z_{\lambda',LB} = (z_{\lambda,LB} + \ell r_j) \frac{1}{\ell + 1}$$

holds. For $i \in I \backslash \{0\}$, we easily see that $\|r_i\| \leqslant 2|V|$ holds. Moreover, the optimal value and therefore LB is always bounded by $|V|$. Therefore, $\|r_i\| \leqslant 2|V|$ even holds for all

$i \in I$. Since z is just a convex combination of these vectors, $\|z\| \leqslant 2|V|$ holds as well. With these observations, we conclude

$$\begin{aligned}
\|z_{\lambda',LB}\|^2 &= \left\| z_{\lambda,LB} + \frac{\ell}{\ell+1}\left(r_j - z_{\lambda,LB}\right) \right\|^2 \\
&= \|z_{\lambda,LB}\|^2 - 2\frac{\ell}{\ell+1}\left\langle z_{\lambda,LB}, z_{\lambda,LB} - r_j \right\rangle + \left(\frac{\ell}{\ell+1}\right)^2 \|z_{\lambda,LB} - r_j\|^2 \\
&\leqslant \|z_{\lambda,LB}\|^2 - 2\frac{\ell}{\ell+1}\left\langle z_{\lambda,LB}, z_{\lambda,LB} \right\rangle + \left(\frac{\ell}{\ell+1}\right)^2 \|z_{\lambda,LB} - r_j\|^2 \\
&\leqslant \|z_{\lambda,LB}\|^2 - 2\frac{\ell}{\ell+1}\|z_{\lambda,LB}\|^2 + \left(\frac{\ell}{\ell+1}\right)^2 16|V|^2 \\
&= \|z_{\lambda,LB}\|^2 - 2\frac{\|z_{\lambda,LB}\|^2}{16|V|^2}\|z_{\lambda,LB}\|^2 + \left(\frac{\|z_{\lambda,LB}\|^2}{16|V|^2}\right)^2 16|V|^2 \\
&= \|z_{\lambda,LB}\|^2 - \frac{\|z_{\lambda,LB}\|^4}{16|V|^2},
\end{aligned}$$

using that $\langle r_j, z_{\lambda,LB}\rangle = -R(j) \leqslant 0$ holds. □

As mentioned, other step lengths can be chosen. In [Dad+22], the vectors r_i for $i \in I$ are scaled to obtain a better step length and therefore also a better running time. We focus on the version without scaling, to obtain a simpler algorithm and analysis.

The following Lemma 4.8 links the norm of z with the approximation error.

Lemma 4.8. *For $\|z\| < \frac{1}{8|V|}$, we have*

$$LB \leqslant \max\left\{\tfrac{|E(H)|}{|V(H)|} : H \text{ is a non-empty subgraph of } G\right\} \leqslant LB + \frac{2\|z\|\left(1+16|V|^2\right)}{1-8\|z\|\|V|}.$$

Proof. For any optimal vertex solution (x, y), we have

$$\begin{aligned}
\left\langle \begin{pmatrix}\mathbf{0}\\ \mathbf{1}\end{pmatrix}, \begin{pmatrix}x\\ y\end{pmatrix}\right\rangle - LB &= \frac{1}{\lambda(0)}\lambda(0)\left\langle \begin{pmatrix}\mathbf{0}\\ \mathbf{1}\\ LB\end{pmatrix}, \begin{pmatrix}x\\ y\\ -1\end{pmatrix}\right\rangle \\
&= \frac{1}{\lambda(0)}\left(\left\langle \lambda(0)\begin{pmatrix}\mathbf{0}\\ \mathbf{1}\\ LB\end{pmatrix} - \sum_{i\neq 0}\lambda(i) r_i, \begin{pmatrix}x\\ y\\ -1\end{pmatrix}\right\rangle + \left\langle \sum_{i\neq 0}\lambda(i) r_i, \begin{pmatrix}x\\ y\\ -1\end{pmatrix}\right\rangle\right) \\
&= \frac{1}{\lambda(0)}\left(\left\langle -z, \begin{pmatrix}x\\ y\\ -1\end{pmatrix}\right\rangle + \sum_{i\neq 0}\lambda(i)\left\langle r_i, \begin{pmatrix}x\\ y\\ -1\end{pmatrix}\right\rangle\right) \\
&\leqslant \frac{1}{\lambda(0)}\|z\|\left\|\begin{pmatrix}x\\ y\\ -1\end{pmatrix}\right\| \leqslant \frac{2}{\lambda(0)}\|z\|,
\end{aligned}$$

where the first inequality follows by the Cauchy–Schwarz inequality, and because (x, y) is feasible, which is equivalent to

$$\left\langle r_i, \begin{pmatrix} x \\ y \\ -1 \end{pmatrix} \right\rangle \leqslant 0$$

for all $i \neq 0$. For the second inequality, we note that y is non-negative and $\sum_{v \in V} x(v) = 1$ holds in any optimal vertex solution. Therefore, it suffices to consider only the vertices of P_{dense} to bound $\|(x, y, -1)\|$. As observed in Section 4.3, a vertex of P_{dense} is of the form $\frac{1}{|V(H)|}\chi_H$, where H is a subgraph of G, and

$$\left\| \begin{pmatrix} \frac{1}{|V(H)|}\chi_H \\ -1 \end{pmatrix} \right\| \leqslant \sqrt{|V(H)| \left(\frac{1}{|V(H)|}\right)^2 + |E(H)| \left(\frac{1}{|V(H)|}\right)^2 + 1} \leqslant \sqrt{\frac{1}{|V(H)|} + 2} \leqslant 2.$$

We are left with bounding $\lambda(0)$. Therefore, we consider the feasible point $(\bar{x}, \bar{y}) := \left(\frac{1}{2|V|}\mathbf{1}, \frac{1}{4|V|}\mathbf{1}\right)$ that is not tight at any constraint. It even satisfies

$$\left\langle r_i, \begin{pmatrix} -\bar{x} \\ -\bar{y} \\ 1 \end{pmatrix} \right\rangle \geqslant \frac{1}{4|V|}$$

for all $i \neq 0$. We note that

$$\|(-\bar{x}, -\bar{y}, 1)\| = \sqrt{|V|\frac{1}{4|V|^2} + |E|\frac{1}{16|V|^2} + 1} \leqslant 2.$$

Therefore, again using the Cauchy–Schwarz inequality, we obtain

$$\begin{aligned}
\|z\| \left\| \begin{pmatrix} -\bar{x} \\ -\bar{y} \\ 1 \end{pmatrix} \right\| &\geqslant \left\langle \sum_{i \neq 0} \lambda(i) r_i - \lambda(0) \begin{pmatrix} \mathbf{0} \\ \mathbf{1} \\ LB \end{pmatrix}, \begin{pmatrix} -\bar{x} \\ -\bar{y} \\ 1 \end{pmatrix} \right\rangle \\
&\geqslant \sum_{i \neq 0} \lambda(i) \frac{1}{4|V|} - \lambda(0) \left\langle \begin{pmatrix} \mathbf{0} \\ \mathbf{1} \\ LB \end{pmatrix}, \begin{pmatrix} -\bar{x} \\ -\bar{y} \\ 1 \end{pmatrix} \right\rangle \\
&= (1 - \lambda(0)) \frac{1}{4|V|} - \lambda(0) \left\langle \begin{pmatrix} \mathbf{0} \\ \mathbf{1} \\ LB \end{pmatrix}, \begin{pmatrix} -\bar{x} \\ -\bar{y} \\ 1 \end{pmatrix} \right\rangle \\
&\geqslant (1 - \lambda(0)) \frac{1}{4|V|} - \lambda(0) \left\| \begin{pmatrix} \mathbf{0} \\ \mathbf{1} \\ LB \end{pmatrix} \right\| \left\| \begin{pmatrix} -\bar{x} \\ -\bar{y} \\ 1 \end{pmatrix} \right\| \\
&\geqslant (1 - \lambda(0)) \frac{1}{4|V|} - \lambda(0) 4|V| = \frac{1}{4|V|} - \lambda(0) \left(\frac{1}{4|V|} + 4|V| \right),
\end{aligned}$$

which in turn yields

$$\lambda(0) \geqslant \frac{\frac{1}{4|V|} - 2\|z\|}{\frac{1}{4|V|} + 4|V|}.$$

We conclude

$$\left\langle \begin{pmatrix} \mathbf{0} \\ \mathbf{1} \end{pmatrix}, \begin{pmatrix} x \\ y \end{pmatrix} \right\rangle - LB \leqslant 2\|z\| \frac{\frac{1}{4|V|} + 4|V|}{\frac{1}{4|V|} - 2\|z\|} = \frac{2\|z\| \left(1 + 16|V|^2\right)}{1 - 8\|z\||V|}.$$

□

Combining Lemma 4.7 and Lemma 4.8, we obtain to following result.

Theorem 4.9. *For every $\varepsilon > 0$, we obtain a feasible solution satisfying*

$$LB \leqslant \max \left\{ \frac{|E(H)|}{|V(H)|} : H \textit{ is a subgraph of } G \right\} \leqslant LB + \varepsilon$$

after $\mathcal{O}\left(\frac{|V|^6}{\varepsilon^2}\right)$ iterations in Algorithm 1.

Proof. Since we aim for an additive error of at most ε, Lemma 4.8 tells us to stop Algorithm 1 if the norm of z satisfies

$$\varepsilon \geqslant \frac{2\|z\| \left(1 + 16|V|^2\right)}{1 - 8\|z\||V|}, \text{ and } \|z\| < \frac{1}{8|V|},$$

which is satisfied if

$$\|z\| < \frac{\varepsilon}{2\left(1 + 16|V|^2\right) + \varepsilon 8|V|}. \tag{4.7}$$

We note that whenever we update LB during the algorithm, we do not increase z. Moreover, the algorithm starts with a vector $z = z_0 = r_0$, satisfying

$$\|z_0\|^2 = |E| + \frac{|E|^2}{|V|^2} \leqslant 2|V|^2. \tag{4.8}$$

Furthermore, $R(0) = 0$ holds after each update of the LB. Therefore, the algorithm either stops in the iteration after an update of LB, or Lemma 4.7 applies in the next iteration. Let t be the number of iterations that we run Algorithm 1 for. For $i \in \{0, 1, \ldots, t\}$, we denote the vector z after the ith iteration of the algorithm, by z_i. Hence, by applying Lemma 4.7 in at least every second iteration,

$$\|z_i\|^2 \leqslant \|z_{i-2}\|^2 - \frac{1}{16|V|^2} \|z_{i-2}\|^4$$

holds for all $i \in \{2, \dots, t\}$. Therefore,

$$\begin{aligned}\frac{1}{\|z_i\|^2} &\geqslant \frac{1}{\|z_{i-2}\|^2 \left(1 - \frac{1}{16|V|^2}\|z_{i-2}\|^2\right)} \\ &\geqslant \frac{1 - \left(\frac{1}{16|V|^2}\|z_{i-2}\|^2\right)^2}{\|z_{i-2}\|^2 \left(1 - \frac{1}{16|V|^2}\|z_{i-2}\|^2\right)} = \frac{1 + \frac{1}{16|V|^2}\|z_{i-2}\|^2}{\|z_{i-2}\|^2} = \frac{1}{\|z_{i-2}\|^2} + \frac{1}{16|V|^2}.\end{aligned}$$

For even t this implies

$$\frac{1}{\|z_t\|^2} \geqslant \frac{1}{\|z_0\|^2} + \frac{t}{32|V|^2}.$$

Since z_0 satisfies (4.8), its norm is less than $32|V|^2$, and we obtain

$$\frac{1}{\|z_t\|^2} \geqslant \frac{t+1}{32|V|^2},$$

if t is even. With $\|z_t\| \leqslant \|z_{t-1}\|$ for $t \in \mathbb{Z}_{\leqslant 1}$, we conclude

$$\|z_t\| \leqslant \frac{\sqrt{32}|V|}{\sqrt{t}}$$

for general t. To obtain the desired error in the approximation, the number of iterations t needs to satisfy the following condition, which we obtain using (4.7),

$$\frac{\sqrt{32}|V|}{\sqrt{t}} < \frac{\varepsilon}{2\left(1 + 16|V|^2\right) + \varepsilon 8|V|}.$$

This is satisfied if

$$t > \left(\frac{\sqrt{32}|V|\left(2\left(1 + 16|V|^2\right) + \varepsilon 8|V|\right)}{\varepsilon}\right)^2,$$

which yields the claim. □

We can use the discussed approximation algorithm to determine the exact density of a densest subgraph. For this, we use the following fact that has been exploited for many exact algorithms, see [Gol84]. The set of different densities that may occur in a subgraph is a finite discrete set. To be more precise, every density d of a subgraph of G lies within the set $\{m/n : 0 \leqslant m \leqslant |E|, 1 \leqslant n \leqslant |V|\}$. Given two distinct densities $m_1/n_1, m_2/n_2$, their difference equals

$$\left|\tfrac{m_1}{n_1} - \tfrac{m_2}{n_2}\right| = \left|\tfrac{m_1 n_2 - m_2 n_1}{n_1 n_2}\right|. \tag{$*$}$$

Since $n_1 n_2 \leqslant |V|^2$ holds, we obtain $(*) \geqslant \frac{1}{|V|^2}$. So within an interval of length $\frac{1}{|V|^2}$ there is at most one value that is a fraction between some number of edges and some number of vertices. Therefore, having an additive error of at most $\frac{1}{|V|^2}$ suffices to obtain the exact optimal value. Due to Theorem 4.9, Algorithm 1 can be used to find the exact optimal value to the densest subgraph problem in $\mathcal{O}(|V|^{10})$ many iterations.

4.5 Open questions

Using the original algorithm with the scaling as in [Dad+22], we only need $\mathcal{O}\left(\frac{|V|^4}{\varepsilon^2}\right)$ iterations in Theorem 4.9. Since the output of the algorithm will always be better than $|E|/|V|$, this yields a multiplicative ε-approximation with $\mathcal{O}\left(\frac{|V|^6}{\varepsilon^2|E|^2}\right)$ iterations. For a very dense graph, this is quite comparable to the existing ε-approximations. Regarding the running time two very natural questions arise. First, from a theoretical perspective we ask:

Question 4.10. *Is the analysis of the running time tight?*

We suppose that the worst case analysis of the progress within one iteration is tight, but we guess that within some iterations the progress will be noticeably larger.

On the practical side, a reasonable next step is to implement the proposed algorithm to provide computational comparison. Besides the already mentioned ε-approximation [BGM14], it may also be interesting to compare against the approximation algorithm in [Boo+20] by Boob, Sawlani, Wang, and others. This algorithm named Greedy++ is not yet proven to be an ε-approximation, but it seems to work quite well in practice. Moreover, it was designed with the goal of developing a simple algorithm such as we did.

Question 4.11. *How does our algorithm perform in practice?*

When performing practical experiments, it will also be interesting to compare the occurrence of primal steps, meaning updating the lower bound, against the amount of dual steps, meaning updating λ to decrease the norm of z. Do they alternate, or does one step dominate in the beginning and the other at the end? Is the amount of steps of one kind noticeably larger than the other or are they about the same?

Although, our algorithm only performs simple updates on values assigned to nodes and edges, we did not achieve what we aimed for. Regarding the suggested LP, the algorithm outputs intermediate primal solutions, but these do not give rise to actual intermediate subgraphs. So our ambitious question from the beginning is still present. Sticking to our proposed algorithm, it would be great to obtain intermediate subgraphs in addition to the intermediate primal solutions.

Question 4.12. *Can we adopt our algorithm in such a way that it outputs intermediate subgraphs whose density hits the current lower bound without involving complicated computations?*

Bibliography

[AR05] D. Aharonov and O. Regev. “Lattice problems in NP $\cap$ coNP”. In: *J. ACM* 52.5 (2005), pp. 749–765.

[Ake22] M. Akemann. “The Densest Subgraph Polytope”. MA thesis. Technical University Munich, 2022.

[AF22] M. Aprile and S. Fiorini. “Regular matroids have polynomial extension complexity”. In: *Mathematics of Operations Research* 47.1 (2022), pp. 540–559.

[Art+16] S. Artmann, F. Eisenbrand, C. Glanzer, T. Oertel, S. Vempala, and R. Weismantel. “A note on non-degenerate integer programs with small sub-determinants”. In: *Operations Research Letters* 44.5 (2016), pp. 635–639.

[AWZ17] S. Artmann, R. Weismantel, and R. Zenklusen. “A strongly polynomial algorithm for bimodular integer linear programming”. In: *Proceedings of the forty-ninth Annual ACM Symposium on Theory of Computing (STOC)*. ACM, New York, 2017, pp. 1206–1219.

[Asa+00] Y. Asahiro, K. Iwama, H. Tamaki, and T. Tokuyama. “Greedily finding a dense subgraph”. In: *Journal of Algorithms* 34.2 (2000), pp. 203–221.

[BGM14] B. Bahmani, A. Goel, and K. Munagala. “Efficient primal-dual graph algorithms for mapreduce”. In: *International Workshop on Algorithms and Models for the Web-Graph*. Springer. 2014, pp. 59–78.

[Bal79] E. Balas. “Disjunctive Programming”. In: *Discrete Optimization II*. Ed. by P. Hammer, E. Johnson, and B. Korte. Vol. 5. Annals of Discrete Mathematics. Elsevier, 1979, pp. 3–51.

[BM86] F. Barahona and A. R. Mahjoub. “On the cut polytope”. In: *Mathematical programming* 36.2 (1986), pp. 157–173.

[Ber57] C. Berge. “Two theorems in graph theory”. In: *Proceedings of the National Academy of Sciences* 43.9 (1957), pp. 842–844.

[Bir46] G. Birkhoff. “Tres observaciones sobre el algebra lineal”. In: *Univ. Nac. Tucuman, Ser. A* 5 (1946), pp. 147–154.

[Bog+15] Y. Bogomolov, S. Fiorini, A. Maksimenko, and K. Pashkovich. “Small extended formulations for cyclic polytopes”. In: *Discrete Comput. Geom.* 53.4 (2015), pp. 809–816.

[Boo+20] D. Boob, Y. Gao, R. Peng, S. Sawlani, C. Tsourakakis, D. Wang, and J. Wang. “Flowless: Extracting densest subgraphs without flow computations”. In: *Proceedings of The Web Conference 2020*. 2020, pp. 573–583.

[BP14] G. Braun and S. Pokutta. “The matching polytope does not admit fully-polynomial size relaxation schemes”. In: *Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM. 2014, pp. 837–846.

[CEN12] E. W. Chambers, J. Erickson, and A. Nayyeri. “Homology flows, cohomology cuts”. In: *SIAM Journal on Computing* 41.6 (2012), pp. 1605–1634.

[Cha+16] S. O. Chan, J. R. Lee, P. Raghavendra, and D. Steurer. “Approximate constraint satisfaction requires large LP relaxations”. In: *J. ACM* 63.4 (2016), pp. 1–22.

[Cha00] M. Charikar. “Greedy approximation algorithms for finding dense components in a graph”. In: *International Workshop on Approximation Algorithms for Combinatorial Optimization*. Springer. 2000, pp. 84–95.

[CS10] J. Chen and Y. Saad. “Dense subgraph extraction with application to community detection”. In: *IEEE Transactions on knowledge and data engineering* 24.7 (2010), pp. 1216–1230.

[Chv75] V. Chvátal. “On certain polytopes associated with graphs”. In: *Journal of Combinatorial Theory, Series B* 18.2 (1975), pp. 138–154.

[CCZ13] M. Conforti, G. Cornuéjols, and G. Zambelli. “Extended formulations in combinatorial optimization”. In: *Ann. Oper. Res.* 204.1 (2013), pp. 97–143.

[Con+20a] M. Conforti, S. Fiorini, T. Huynh, G. Joret, and S. Weltge. “The stable set problem in graphs with bounded genus and bounded odd cycle packing number”. In: *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM. 2020, pp. 2896–2915.

[Con+20b] M. Conforti, S. Fiorini, T. Huynh, and S. Weltge. “Extended formulations for stable set polytopes of graphs without two disjoint odd cycles”. In: *International Conference on Integer Programming and Combinatorial Optimization*. Springer. 2020, pp. 104–116.

[CS99] J. H. Conway and N. J. A. Sloane. *Sphere packings, lattices and groups*. 3. corr. ed. Vol. 290. Grundlehren der Mathematischen Wissenschaften. Springer-Verlag, New York, 1999, pp. lxxiv+703.

[DB15] D. Dadush and N. Bonifas. “Short paths on the Voronoi graph and closest vector problem with preprocessing”. English. In: *Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 2015, pp. 295–314.

[Dad+22] D. Dadush, C. Hojny, S. Huiberts, and S. Weltge. “A simple method for convex optimization in the oracle model”. In: *International Conference on Integer Programming and Combinatorial Optimization*. Springer. 2022, pp. 154–167.

[Dad12] D. N. Dadush. “Integer programming, lattice algorithms, and deterministic volume estimation”. PhD thesis. 2012.

[Dan16] G. Dantzig. “Linear programming and extensions”. In: *Linear programming and extensions*. Princeton university press, 2016.

[DHK11] T. K. Dey, A. N. Hirani, and B. Krishnamoorthy. “Optimal homologous cycles, total unimodularity, and linear programming”. In: *SIAM Journal on Computing* 40.4 (2011), pp. 1026–1044.

[DKS98] I. Dinur, G. Kindler, and S. Safra. “Approximating-CVP to within almost-polynomial factors is NP-hard”. In: *Proceedings 39th Annual Symposium on Foundations of Computer Science (Cat. No. 98CB36280)*. IEEE. 1998, pp. 99–109.

[DH11] N. M. Dunfield and A. N. Hirani. “The least spanning area of a knot and the optimal bounding chain problem”. In: *Proceedings of the Twenty-Seventh Annual Symposium on Computational Geometry (SOCG)*. ACM. 2011, pp. 135–144.

[Edm71] J. Edmonds. “Matroids and the greedy algorithm”. In: *Mathematical programming* 1.1 (1971), pp. 127–136.

[Edm65a] J. Edmonds. “Maximum matching and a polyhedron with 0, 1-vertices”. In: *Journal of research of the National Bureau of Standards B* 69.125-130 (1965), pp. 55–56.

[Edm65b] J. Edmonds. “Paths, trees, and flowers”. In: *Canadian Journal of mathematics* 17 (1965), pp. 449–467.

[EW19] F. Eisenbrand and R. Weismantel. “Proximity results and faster algorithms for Integer Programming using the Steinitz Lemma”. In: *ACM Transactions on Algorithms (TALG)* 16.1 (2019), pp. 1–14.

[Emd81] P. van Emde Boas. “Another NP-complete problem and the complexity of computing short vectors in a lattice”. In: *Technical Report, Department of Mathmatics, University of Amsterdam* (1981).

[EMS04] P. Engel, L. Michel, and M. Sénéchal. *Lattice geometry*. preprint, 2004.

[Erd99] R. M. Erdahl. “Zonotopes, dicings, and Voronoi’s conjecture on parallelohedra”. In: *European J. Combin.* 20.6 (1999), pp. 527–549.

[EN11] J. Erickson and A. Nayyeri. “Minimum cuts and shortest non-separating cycles via homology covers”. In: *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM. 2011, pp. 1166–1176.

[FOS12] Y. Faenza, G. Oriolo, and G. Stauffer. “Separating stable sets in claw-free graphs via Padberg-Rao and compact linear programs”. In: *Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM. 2012, pp. 1298–1308.

[FR19] A. Faragó and Z. R. Mojaveri. “In search of the densest subgraph”. In: *Algorithms* 12.8 (2019), p. 157.

[FPK01] U. Feige, D. Peleg, and G. Kortsarz. “The dense k-subgraph problem”. In: *Algorithmica* 29.3 (2001), pp. 410–421.

[Fio+22] S. Fiorini, G. Joret, S. Weltge, and Y. Yuditsky. “Integer programs with bounded subdeterminants and two nonzeros per row”. In: *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE. 2022, pp. 13–24.

[Fio+13] S. Fiorini, V. Kaibel, K. Pashkovich, and D. O. Theis. “Combinatorial bounds on nonnegative rank and extended formulations”. In: *Discrete Math.* 313.1 (2013), pp. 67–83.

[Fio+12] S. Fiorini, S. Massar, S. Pokutta, H. R. Tiwary, and R. De Wolf. “Linear vs. Semidefinite Extended Formulations: Exponential Separation and Strong Lower Bounds”. In: *Proceedings of the forty-fourth Annual ACM Symposium on Theory of Computing (STOC)*. 2012, pp. 95–106.

[Fio+15] S. Fiorini, S. Massar, S. Pokutta, H. R. Tiwary, and R. D. Wolf. “Exponential lower bounds for polytopes in combinatorial optimization”. In: *J. ACM* 62.2 (2015), pp. 1–23.

[Fra+06] E. Fratkin, B. T. Naughton, D. L. Brutlag, and S. Batzoglou. “MotifCut: regulatory motifs finding with maximum density subgraphs”. In: *Bioinformatics* 22.14 (2006), e150–e157.

[GGT89] G. Gallo, M. D. Grigoriadis, and R. E. Tarjan. "A fast parametric maximum flow algorithm and applications". In: *SIAM Journal on Computing* 18.1 (1989), pp. 30–55.

[GJ79] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. Mathematical Sciences Series. Freeman, 1979. ISBN: 9780716710448. URL: `https://books.google.de/books?id=fjxGAQAAIAAJ`.

[Ger91] A. M. H. Gerards. "Compact systems for T-join and perfect matching polyhedra of graphs with bounded genus". In: *Oper. Res. Lett.* 10.7 (1991), pp. 377–382.

[GKT05] D. Gibson, R. Kumar, and A. Tomkins. "Discovering large dense subgraphs in massive graphs". In: *Proceedings of the 31st international conference on Very large data bases*. Citeseer. 2005, pp. 721–732.

[Goe15] M. X. Goemans. "Smallest compact formulation for the permutahedron". In: *Math. Program.* 153 (2015), pp. 5–11.

[Gol84] A. V. Goldberg. *Finding a maximum density subgraph*. 1984.

[GM03] D. Goldstein and A. Mayer. "On the equidistribution of Hecke points". In: *Forum Mathematicum*. Vol. 15. 2. De Gruyter. 2003, pp. 165–189.

[GJW18] M. Göös, R. Jain, and T. Watson. "Extension complexity of independent set polytopes". In: *SIAM J. Comput.* 47.1 (2018), pp. 241–269.

[GPT13] J. Gouveia, P. A. Parrilo, and R. R. Thomas. "Lifts of convex sets and cone factorizations". In: *Math. Oper. Res.* 38.2 (2013), pp. 248–264.

[GPS18] F. Grande, A. Padrol, and R. Sanyal. "Extension Complexity and Realization Spaces of Hypersimplices". In: *Discrete Comput. Geom.* 59 (2018), pp. 621–642.

[GKS95] J. W. Grossman, D. M. Kulkarni, and I. E. Schochetman. "On the minors of an incidence matrix and its Smith normal form". In: *Linear Algebra and its Applications* 218 (1995), pp. 213–224.

[GLS93] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric algorithms and combinatorial optimization*. 2. corr. ed. Vol. 2. Algorithms Comb. Springer-Verlag, Berlin, 1993.

[Gru07] P. M. Gruber. *Convex and discrete geometry*. Vol. 336. Grundlehren der Mathematischen Wissenschaftens. Springer-Verlag, Berlin, 2007, pp. xiv+578.

[Hat05] A. Hatcher. *Algebraic topology*. Cambridge University Press, 2005.

[HK78] D. Hausmann and B. Korte. "Colouring criteria for adjacency on 0–1-polyhedra". In: *Polyhedral Combinatorics*. Springer, 1978, pp. 106–127.

[Hel57] I. Heller. "On linear systems with integral valued solutions". In: *Pacific J. Math.* 7.3 (1957), pp. 1351–1364.

[Hoo+16] B. Hooi, H. A. Song, A. Beutel, N. Shah, K. Shin, and C. Faloutsos. "Fraudar: Bounding graph fraud in the face of camouflage". In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 2016, pp. 895–904.

[Hun20] C. Hunkenschröder. "New Results in Integer and Lattice Programming". PhD thesis. EPFL, Lausanne, 2020.

[HRS20] C. Hunkenschröder, G. Reuland, and M. Schymura. "On compact representations of Voronoi cells of lattices". In: *Math. Program.* 183.1 (2020), pp. 337–358.

[KW15] V. Kaibel and S. Weltge. “A short proof that the extension complexity of the correlation polytope grows exponentially”. In: *Discrete Comput. Geom.* 53.2 (2015), pp. 397–401.

[Kan87] R. Kannan. “Minkowski’s convex body theorem and integer programming”. In: *Math. Oper. Res.* 12.3 (1987), pp. 415–440.

[Kar72] R. M. Karp. “Reducibility among combinatorial problems”. In: *Complexity of computer computations*. Springer, 1972, pp. 85–103.

[Khi48] A. Y. Khinchin. “A quantitative formulation of the approximation theory of Kronecker”. In: *Izvestiya Rossiiskoi Akademii Nauk. Seriya Matematicheskaya* 12.2 (1948), pp. 113–122.

[KS09] S. Khuller and B. Saha. “On finding dense subgraphs”. In: *International colloquium on automata, languages, and programming*. Springer. 2009, pp. 597–608.

[KZ77] A. N. Korkine and E. I. Zolotarev. “Sur les formes quadratiques positives”. In: *Math. Ann.* 11 (1877), pp. 242–292.

[KV11] B. H. Korte and J. Vygen. *Combinatorial optimization*. Vol. 1. Springer, 2011.

[LRS15] J. R. Lee, P. Raghavendra, and D. Steurer. “Lower bounds on the size of semidefinite programming relaxations”. In: *Proceedings of the forty-seventh Annual ACM Symposium on Theory of Computing (STOC)*. 2015, pp. 567–576.

[Lee+10] V. E. Lee, N. Ruan, R. Jin, and C. Aggarwal. “A survey of algorithms for dense subgraph discovery”. In: *Managing and mining graph data*. Springer, 2010, pp. 303–336.

[LLL82] A. K. Lenstra, H. W. Lenstra, and L. Lovász. “Factoring polynomials with rational coefficients”. In: *Mathematische annalen* 261 (1982), pp. 515–534.

[Len83] H. W. Lenstra Jr. “Integer programming with a fixed number of variables”. In: *Mathematics of operations research* 8.4 (1983), pp. 538–548.

[MM92] A. Malnič and B. Mohar. “Generating locally cyclic triangulations of surfaces”. In: *J. Combin. Theory Ser. B* 56.2 (1992), pp. 147–164. ISSN: 0095-8956. URL: `https://doi.org/10.1016/0095-8956(92)90015-P`.

[Mar91] R. K. Martin. “Using separation algorithms to generate mixed integer model reformulations”. In: *Oper. Res. Lett.* 10.3 (1991), pp. 119–128.

[Mar03] J. Martinet. *Perfect lattices in Euclidean spaces*. Vol. 327. Grundlehren der Mathematischen Wissenschaften. Springer-Verlag, Berlin, 2003, pp. xxii+523.

[McC+21] S. T. McCormick, B. Peis, R. Scheidweiler, and F. Vallentin. “A polynomial time algorithm for solving the closest vector problem in zonotopal lattices”. In: *SIAM Journal on Discrete Mathematics* 35.4 (2021), pp. 2345–2356.

[MG02] D. Micciancio and S. Goldwasser. *Complexity of lattice problems: a cryptographic perspective*. Vol. 671. Springer Science & Business Media, 2002.

[MV13] D. Micciancio and P. Voulgaris. “A deterministic single exponential time algorithm for most lattice problems based on Voronoi cell computations”. In: *SIAM J. Comput.* 42.3 (2013), pp. 1364–1391.

[MT01] B. Mohar and C. Thomassen. *Graphs on surfaces*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, MD, 2001.

[MP92] R. V. Moody and J. Patera. “Voronoi and Delaunay cells of root lattices: classification of their faces and facets by Coxeter-Dynkin diagrams”. In: *J. Phys. A: Math. Gen.* 25.19 (1992), pp. 5089–5134.

[MSW21] S. Morell, I. Seidel, and S. Weltge. “Minimum-cost integer circulations in given homology classes”. In: *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM. 2021, pp. 2725–2739.

[Neu53] J. von Neumann. “A certain zero-sum two-person game equivalent to the optimal assignment problem”. In: *Contributions to the Theory of Games*. Ed. by H. W. Kuhn and A. W. Tucker. Vol. 2. Annals of Mathematics Studies 28. Princeton University Press, 1953, pp. 5–12.

[PR74] M. W. Padberg and M. R. Rao. “The travelling salesman problem and a class of polyhedra of diameter two”. In: *Mathematical Programming* 7.1 (1974), pp. 32–45.

[Pap81] C. H. Papadimitriou. “On the complexity of integer programming”. In: *Journal of the ACM* 28.4 (1981), pp. 765–768.

[PQ82] J.-C. Picard and M. Queyranne. “A network flow solution to some nonlinear 0-1 programming problems, with applications to graph theory”. In: *Networks* 12.2 (1982), pp. 141–159.

[PS93] W. R. Pulleyblank and B. Shepherd. “Formulations for the stable set polytope”. In: *International Conference on Integer Programming and Combinatorial Optimization*. 1993, pp. 267–279.

[Rad57] R. Rado. “Note on independence functions”. In: *Proceedings of the London Mathematical Society* 3.1 (1957), pp. 300–320.

[RS96] D. S. Rajan and A. M. Shende. “A characterization of root lattices”. In: *Discrete Math.* 161.1 (1996), pp. 309–314.

[RC98] F. J. Rispoli and S. Cosares. “A bound of 4 for the diameter of the symmetric traveling salesman polytope”. In: *SIAM Journal on Discrete Mathematics* 11.3 (1998), pp. 373–380.

[Rot13] T. Rothvoß. “Some 0/1 polytopes need exponential size extended formulations”. In: *Math. Program.* 142.1 (2013), pp. 255–268.

[Rot17] T. Rothvoß. “The matching polytope has exponential extension complexity”. In: *J. ACM* 64.6 (2017), pp. 1–19.

[Sah+10] B. Saha, A. Hoch, S. Khuller, L. Raschid, and X.-N. Zhang. “Dense subgraphs with restrictions and applications to gene annotation graphs”. In: *Annual International Conference on Research in Computational Molecular Biology*. Springer. 2010, pp. 456–472.

[Sch98] A. Schrijver. *Theory of linear and integer programming*. John Wiley & Sons, 1998.

[Sch09] A. Schürmann. *Computational Geometry of Positive Definite Quadratic Forms: Polyhedral Reduction Theories, Algorithms, and Applications*. Vol. 48. University lecture series. American Mathematical Society, 2009, pp. xv+162.

[SSW21] M. Schymura, I. Seidel, and S. Weltge. “Lifts for Voronoi cells of lattices”. In: *arXiv:2106.04432, to appear in Discrete & Computational Geometry 2023* (2021).

[Shi14] Y. Shitov. *Sublinear extensions of polygons*. `https://arxiv.org/abs/1412.0728`. 2014.

[Sie45] C. L. Siegel. “A mean value theorem in geometry of numbers”. In: *Ann. Math.* (1945), pp. 340–347.

[Wel15] S. Weltge. “Sizes of Linear Descriptions in Combinatorial Optimization”. PhD thesis. Otto-von-Guericke-Universität Magdeburg, 2015. URL: `http://dx.doi.org/10.25673/4350`.

[Won80] R. T. Wong. “Integer programming formulations of the traveling salesman problem”. In: *Proceedings of the IEEE international conference of circuits and computers*. IEEE Press Piscataway NJ. 1980, pp. 149–152.

[Yan91] M. Yannakakis. “Expressing combinatorial optimization problems by linear programs”. In: *J. Comput. System Sci.* 43.3 (1991), pp. 441–466.