Tessellations for Geometric Optimization

Dissertation zur Erlangung des Doktorgrades

vorgelegt am

Fachbereich Mathematik und Informatik
der Freien Universität Berlin

von

Rafel Jaume Deyà

2014

Institut für Informatik
Freie Universität Berlin
Takustraße 9
14195 Berlin
rafel.jd@gmail.com

Betreuer: Prof. Dr. Günter Rote
Zweitgutachter: Prof. Dr. Michael Joswig
Vorlage zur Begutachtung: Juli 2014
Datum der Disputation: 20. Oktober 2014

Abstract

This thesis is concerned with the study of some tessellations (or subdivisions) of the plane or of the space and their relation to some optimization problems. Several of the results have a combinatorial flavor, whereas others are strongly connected to the geometry underlying the corresponding problems. The work combines theoretical statements with applied implications and related algorithms, making use of linear algebra, convex geometry, graph theory and many other tools from discrete and computational geometry.

Regular subdivisions are tessellations resulting from the projection of the lower faces of a polyhedron. In the first part of this thesis, we generalize regular subdivisions introducing the class of *recursively-regular subdivisions*. Informally speaking, a recursively-regular subdivision is a subdivision that can be obtained by splitting some faces of a regular subdivision by other regular subdivisions (and continue recursively). We also define the *finest regular coarsening* and the *regularity tree* of a subdivision. We derive several properties of these two objects, which reflect certain structure in the class of non-regular subdivisions. In particular, the finest regular coarsening of a subdivision is the regular subdivision that is (in a sense) most similar to it. We show that the class of recursively-regular subdivisions is a proper superclass of the regular subdivisions and a proper subclass of the visibility-acyclic subdivisions (in the sense of an acyclicity theorem by Edelsbrunner). We also show that there exist point sets whose recursively-regular triangulations are not connected by geometric bistellar flips.

We then derive several algorithms related to the studied objects, and point out applications of the main results. In particular, we present relations to tensegrity theory, data visualization, and graph embedding problems. Special attention is paid to the problem of covering the space by placing given floodlights at given points, for which we extend results known since 1981 and discuss two variants of the original problem.

The second part is concerned with the study of optimal partial matchings for pairs of point sets under translations. First, we regard the least-squares cost function. The best approach to this problem so far is to construct (and explore) a particular tessellation of the space of translations. In every tile of the tessellation there is one matching that is optimal for any position of the point sets corresponding to a translation in that tile. We give the first non-trivial bound on the complexity of this tessellation in dimensions two and higher, and study several structural properties that lead to algorithms whose running time is polynomial in the size of the larger set.

We address then the analogous problem under the bottleneck cost function. This cost function assigns to every matching the largest distance defined by a matched pair of points. An associated tessellation is shown to have polynomial complexity. This result, together with graph-theoretical tools, allows us to obtain efficient algorithms for the computation of the corresponding minimum under translations that are sensitive to the size of the smaller of the two sets. The lexicographic variant of the bottleneck cost is analyzed as well.

Finally, we explore natural directions for the generalization of the problems of matching under translations to which many of our results extend.

Zusammenfassung

In dieser Arbeit betrachten wir Parkettierungen (man sagt auch Unterteilungen) der Ebene und des Raumes sowie ihren Zusammenhang mit verschiedenen Optimierungsproblemen. Einige der Resultate sind kombinatorischer Natur, andere wiederum nutzen stark geometrische Eigenschaften der Probleme. Diese Dissertation kombiniert theoretische Ergebnisse mit konkreten Anwendungen und dazugehörigen Algorithmen mithilfe von Methoden der linearen Algebra, konvexen Geometrie, Graphentheorie und vielen weiteren Hilfsmitteln aus der diskreten und algorithmischen Geometrie.

Reguläre Unterteilungen sind Parkettierungen, die durch die Projektion der Seiten eines Polyeders entstehen. Im ersten Teil verallgemeinern wir reguläre Unterteilungen zu *rekursiv-regulären Unterteilungen*: eine rekursiv-reguläre Unterteilung ist eine Unterteilung, die durch rekursives Aufsplitten von Flächen einer regulären Unterteilung in weitere reguläre Unterteilungen konstruiert werden kann. Darauf aufbauend führen wir die *feinste reguläre Vereinfachung* und den *Regularitätsbaum* einer Unterteilung ein. Wir leiten verschiedene Eigenschaften dieser Objekte her, die die zugrunde liegende Struktur der Klasse der nicht-regulären Unterteilungen widerspiegeln. Insbesondere ist die feinste reguläre Unterteilung einer Unterteilung die in gewissem Sinne "ähnlichste" reguläre Unterteilung. Wir zeigen, dass die Klasse der rekursiv-regulären Unterteilungen die Klasse der regulären Unterteilungen echt enthält und andererseits eine echte Teilklasse der azyklischen Sichtbarkeitsunterteilungen (im Sinne des *Acyclicity Theorem* von Edelsbrunner) ist. Wir beweisen außerdem die Existenz von Punktmengen, deren rekursiv-regulären Triangulierungen nicht durch geometrisch bistellare Flips ineinander überführt werden können.

In Zusammenhang mit den theoretischen Ergebnissen entwickeln wir mehrere Algorithmen und stellen verschiedene Anwendungsmöglichkeiten der Hauptergebnisse vor. Insbesondere zeigen wir Verbindungen der betrachteten Objekte zu der Tensigritätstheorie, der Datenvisualisierung und Grapheinbettungsproblemen auf. Im Vordergrund steht dabei das Problem, einen d-dimensionalen Raum mit Flutlichtern auszuleuchten, die nur an bestimmten Punkten positioniert werden können. Wir verallgemeinern verschiedene, bereits seit 1981 bekannte Resultate und gehen im Detail auf zwei verschiedene Varianten des ursprünglichen Problems ein.

Der zweite Teil der Dissertation beschäftigt sich mit optimalen partiellen Matchings von zwei gegebenen Punktmengen, wobei Parallelverschiebungen der ersten Punktmenge erlaubt sind. Zu Beginn betrachten wir das Problem mit der least-squares-Kostenfunktion. Der beste bekannte Ansatz für dieses Problem ist die Konstruktion einer geeigneten Parkettierung des Raumes der Parallelverschiebungen. In jeder Zelle der Parkettierung gibt es ein Matching, das für alle Positionen der Punktmengen, die durch die Parallelverschiebungen der Zelle bestimmt sind, optimal ist. Wir beweisen die erste nichttriviale Komplexitätsschranke für solche Parkettierungen in zwei und mehr Dimensionen und betrachten verschiedene strukturelle Eigenschaften, die benutzt werden können um Algorithmen zu entwickeln, deren Laufzeit polynomial in der Kardinalität der größeren Punktmenge ist.

Darauf aufbauend betrachten wir das analoge Problem mit der bottleneck Kostenfunktion. Diese weist jedem Matching die größte Distanz der gematchten Punktpaare zu. Wir zeigen, dass eine zugehörige Parkettierung polynomielle Komplexität hat. Kombiniert mit verschiedenen Resultaten aus der Graphentheorie kann so ein effizienter Algorithmus hergeleitet werden, der, falls Parallelverschiebungen erlaubt sind, ein Matching mit minimalen Kosten berechnet und dessen Laufzeit sensitiv gegenüber der Größe der kleineren Punktmenge ist. Zuletzt wird auch die lexikographische Variante dieses Problems analysiert.

Abschließend werden natürliche Verallgemeinerungen von partiellen Matchings mit Parallelverschiebungen vorgestellt, für die viele unserer Ergebnisse weiterhin gültig sind.

Acknowledgments

I would like to thank my supervisor Günter Rote for sharing many good ideas with me, and for the opportunity to participate in the EuroGIGA-Voronoi project. I also want to thank him and Michael Joswig for accepting to review this thesis, and to the committee members for their time.

Many thanks to the whole workgroup *Theoretische Informatik* for the nice environment provided throughout these almost three years. In particular, I would like to thank all my office mates: Daniel Werner (for his wise advice), Balázs Keszegh (for fruitful collaboration) and Yannik Stein (for priceless help). Special thanks to Panos Giannopoulos (for empathic encouragement), to Frank Hoffman and Klaus Kriegel (for teaching me every Sprichwort, and for making me feel at home in the office), and to Wolfgang Mulzer (for continuously enlightening and supporting me). I am very grateful to Andrei Asinowski, Matthias Henze (my frequent and excellent coauthor), Heuna Kim and Tillmann Miltzow for proofreading parts of this thesis, and to Paul Seiferth for tirelessly trying to beat me at badminton.

I would also like to thank Ferran Hurtado and Vera Sacristan for teaching me so much inside and outside the classrooms. I want to thank Francisco Santos for his suggestions and encouragement concerning recursive regularity. I am also grateful to all my coauthors from whom I learned during this period.

I would like to acknowledge the Research Training Group *Methods for Discrete Structures* and the *Berlin Mathematical School* for the provided framework and travel funding, and the *Obra Social La Caixa* and the DAAD for the scholarship.

Finally and most of all, I would like to thank my family and Maria, without whose support I could have never finished any project: Moltíssimes gràcies per la vostra ajuda i paciència; aquesta tesi és tan vostra com meva, perquè sou totalment imprescindibles per a mi!

Rafel

Contents

1

Introduction

1.1 Motivation

Tessellations are important in many areas of discrete and computational geometry. Some of them, like Voronoi diagrams, appear also in many other fields. A reason for this fact could be that subdivisions of the space are a constant in nature; from honeycombs to mud cracks, ranging over the formation of crystals and the growth patterns of cells.

It may be said that constructing tessellations in the plane or higher dimensions is as fundamental as sorting points in the real line since the first operation generalizes, in a sense, the latter one. Spatial partitions have obvious applications in point location, motion planning, VLSI design or data visualization. More surprisingly perhaps, they are also useful in other areas such as geometric optimization, tensegrity theory or shape matching, as we will see. Tessellations are often the key to convert a question that does not have an obvious discrete formulation into a finite problem. They can covert geometric problems into combinatorial ones or reduce the number of candidate solutions significantly. Early examples of these facts are the algorithm to compute the minimum enclosing circle via the furthest-point Voronoi diagram [99], and the construction of the Euclidean minimum spanning tree from the Delaunay triangulation [87]. Spatial tessellations appear in clustering problems as well, such as the *k-centroid* [25] and the *k-center* [31] *problems*. A specially relevant instance for this work is the relation between power diagrams and the *constrained least-squares assignment problem*, studied in [13].

This thesis addresses the relation between tessellations and a variety of problems. The initial motivation for this study are two questions whose partial answers take advantage of certain properties of particular tessellations. The first one is whether given a set of floodlights represented by a polyhedral fan and a set of points, there exists a one-to-one assignment of floodlights to points such that the floodlights cover the space when placed with the apex at the corresponding point. The second one is whether the partial matching between two point sets that minimizes the least-squares cost when one of the sets can be freely translated can be found in polynomial time. Although both problems arise naturally and are easy to state, they might not have received the attention they deserve. The first one was solved in 1981 for instances in the plane and for regular fans in higher dimensions [61, 92]. There has been however no progress ever since. Despite being a fundamental question in point matching, the second problem has been efficiently solved only for equally-sized point sets [110] or in one-dimensional scenarios [93]. In contrast, for the bottleneck variant of this question, polynomial algorithms were discovered some time ago [9, 52]. Nevertheless, the solutions found so far were not sensitive to the size of the smaller set. The purpose of this thesis is to study these problems and the tessellations underlying them, in order to counter the aforementioned weaknesses and unknowns of the available solutions.

1.2 Our contribution

The first part of this thesis is concerned with recursively-regular subdivisions and some related problems. A subdivision of a point set is a polyhedral complex whose vertices belong to the point set, and it is regular if it is the projection of the lower faces of a polyhedron. A subdivision is recursively regular if it is regular or it can be coarsened to a regular subdivision that splits it into recursively-regular parts. In other words, recursively-regular subdivisions are the ones that can be obtained by recursively subdividing a point set in a regular manner. We show that, unlike the regular subdivisions, the recursively-regular subdivisions of a point set are not necessarily connected by bistellar flips. We introduce two constructions closely related to recursively-regular subdivisions: the finest regular coarsening and the regularity tree of a polyhedral subdivision. We provide algorithms for the construction of these objects, which have applications in different areas that we explore. As the main application, we address the problem of finding a one-to-one assignment of a set of floodlights to a set of points such that the floodlights cover the space when translated to the assigned points. The given floodlights are assumed to be the cells of a complete polyhedral fan. We say that the fan is *universal* if the floodlights can cover the space regardless of the given point set. We prove that recursively-regular fans are indeed universal, and that having a cycle in visibility is sufficient yet not necessary for a fan to be non-universal. It remains open though to give a characterization of universal fans. We also examine the problem of deciding, given a polyhedral fan in the plane with assigned positions for the apices of its cells, whether the translated cells cover the plane.

The second part is mainly concerned with the study of optimal matchings under translations. Given two point sets $A, B \subset \mathbb{R}^d$ with $k = |B| \leq |A| = n$, we study the minimization diagram of the functions $f_\sigma(t) = \sum_{b \in B} \|b + t - \sigma(b)\|^2$, for all injections $\sigma : B \hookrightarrow A$. We call this polyhedral complex the least-squares partial-matching Voronoi diagram. This polyhedral complex is only known to have polynomial complexity when the position parameter t ranges along a line [93]. We provide structural properties and bounds on the complexity of this complex, which lead to algorithms for its construction and exploration. In particular, for constant k the bounds read $O(n^{2d})$ for dimension $d > 2$ and $O(n^2)$ in the plane, improving the best previous bound of $O(n^{kd})$ for $d \geq 2$. However, the bounds are exponential in k. We study also the bottleneck partial-matching Voronoi diagram, defined analogously for $f_\sigma(t) = \max_{b \in B} \|b + t - \sigma(b)\|^2$. In this case, the diagram is polynomial in both n and k, allowing us to derive efficient, k-sensitive algorithms. We provide procedures for the construction of the diagrams and the computation of the translations attaining the minimum of the lower envelope of the costs of the matchings. In addition, we explore generalizations of these Voronoi diagrams to other transformation spaces and cost functions.

The first part of this thesis is joint work with Rote, available in [75]. The second part contains results from a note with Keszegh and Henze [70], an article with Ben-Avraham, Henze, Raz, Sharir, Keszegh and Tubis [19, 20], and a recent work with Henze [68, 69].

2

Preliminaries

The specific background and literature related to the problems considered is presented at the beginning of the corresponding parts. We introduce here though general definitions and facts that will be used throughout the thesis.

2.1 Notation and conventions

For the clarity of presentation, we will use henceforth the notation $[n]$ to refer to the set of natural numbers $\{1, \dots, n\}$. The d-dimensional Euclidean space will be denoted by $\mathbb{R}^d$ and $\|\cdot\|$ will denote the Euclidean norm. We will use $\mathbb{R}^A$ for a finite set A to denote $\mathbb{R}^{|A|}$ with an orthonormal basis labeled by the elements of A. The symbol $\mathbb{R}^+$ will denote the positive real numbers, while $\mathbb{R}^{\geq 0}$ will denote the non-negative ones. The bars $|\cdot|$ will generally denote cardinality except when applied to a polyhedral complex, where they will denote the union of all its faces.

The term *poset* is used to denote a partially ordered set (with respect to a certain relation). The power set of a set A will be denoted by 2^A. We adopt the usual convention that the result of applying a function f to a set S of elements in its domain is the set $f(S) = \{f(s) : s \in S\}$. In particular, for a set $B \subset \mathbb{R}^d$ and a point $t \in \mathbb{R}^d$, the expression $B+t$ denotes $\{b+t : b \in B\}$, and $B - t$ is defined similarly. The space of real $m \times n$ matrices is denoted by $\mathbb{R}^{m\times n}$. For a graph $G = (V, E)$ on the vertex set V and set of edges E, an edge $\{v, u\} \in E$ will be abbreviated uv.

2.2 Tessellations, polyhedral complexes and subdivisions

This thesis studies subdivisions and tessellations and how they can be used to derive properties of certain objects. In Mathematics and Computer Science, there is no agreement on the definition of tessellation and the term “subdivision” is used in various contexts with different meanings, as well as “diagram”, “decomposition” or “partition”. In some areas, the term “tessellation” is reserved for a tiling of some space by a finite number of prescribed prototiles. In this thesis, we use the following definition.

Definition 2.1. A *tessellation* of a set $S \subset \mathbb{R}^d$ is a finite family of d-dimensional polyhedra (called *cells*) whose union equals S and whose interiors are pairwise disjoint.

Most of our results can be stated using the previous definition of tessellation. However, we restrict our study most of the time (for the ease of presentation) to objects with more structure: polyhedral complexes. We introduce now basic definitions concerning these (and related) objects, referring the reader to the books [3, 45, 47, 109] for more details.

We use the term *polytope* for a bounded polyhedron, and *polyhedral cone* refers to the (possibly translated) intersection of finitely many closed linear halfspaces. A polyhedral cone is *pointed* if it does not contain any line.

Definition 2.2. A *polyhedral complex* is a finite set $\mathcal{S}$ of polyhedra such that

(1) if $Q \in \mathcal{S}$ and F is a face of Q, then $F \in \mathcal{S}$, and

(2) for all $Q, R \in \mathcal{S}$, $Q \cap R$ is a face of both Q and R.

A *polyhedral fan* is a polyhedral complex whose elements are cones. A fan is *pointed* if all of its cones are pointed. A fan is *complete* if the union of all its cones is the whole ambient space.

The polyhedra in a polyhedral complex will be called *faces*. The *dimension* of a polyhedral complex is the dimension of its top-dimensional faces. A polyhedral complex is *pure* if all its maximal faces have the same dimension. A *cell* is a top-dimensional face of a pure polyhedral complex. A *facet* or *wall* is a face of co-dimension one in a pure complex. As usual, *edges* and *vertices* are one and zero-dimensional faces in the complex, respectively. The (unbounded) one-dimensional faces of a polyhedral fan will be called *rays* as well. The *(combinatorial) complexity* of a polyhedral complex is its number of faces. A pure polyhedral complex embedded in $\mathbb{R}^d$ is *full-dimensional* if it has dimension d.

Definition 2.3. A d-dimensional polyhedral complex is *regular* if its faces are the projection of the lower faces of a $(d+1)$-dimensional polyhedron.

In the first part of the thesis, we will mainly deal with subdivisions of point sets. We will not need the "bullet proof" definition from [45]. Instead, we will use the following relaxed version.

Definition 2.4. Let A be a finite set of points. A *polyhedral subdivision* (or *subdivision*, for short) of A is a polyhedral complex whose vertices are a subset of A and the union of whose cells is the convex hull of A. A *polyhedral subdivision* (or *subdivision*, for short) of a finite set V of vectors is a polyhedral fan whose rays have as directions a subset of V and whose union is the positive span of V. A *triangulation* is a subdivision of a point set consisting only of simplices.

We use the standard notion of a (geometric bistellar) *flip* between triangulations, see [45, Section 2.4] and [98]. The set of subdivisions of a point (or vector) set form a poset with the following relation.

Definition 2.5. Given two complexes $\mathcal{S}$ and $\mathcal{S}'$, we say that $\mathcal{S}'$ *refines* $\mathcal{S}$ if every face of $\mathcal{S}'$ is contained in a face of $\mathcal{S}$. We say that $\mathcal{S}'$ is a *refinement* of $\mathcal{S}$ and $\mathcal{S}$ is a *coarsening* of $\mathcal{S}'$.

Our notion of a subdivision and the refinement relation are simpler than the more subtle definitions in [45, Section 2.3] or in [98]. The differences are not relevant to our results.

Most of the first part of the thesis is concerned with the following generalization of regular subdivisions, for which we give here a tentative, informal definition.

Definition 2.6. A polyhedral complex $\mathcal{S}$ is *recursively regular* if it is regular, or it has a regular coarsening $\mathcal{S}'$ such that for each cell $C \in \mathcal{S}'$, the restriction of $\mathcal{S}$ to C is recursively regular.

In the second part of the thesis we extensively use the following polyhedral complexes.

Definition 2.7. Given a finite set H of hyperplanes in $\mathbb{R}^d$, the *arrangement* of H is the polyhedral complex whose cells are the closure of the connected components of the set

$$\mathbb{R}^d \setminus \left(\bigcup_{h \in H} h \right).$$

The duality transformation relating a set of points with a set of hyperplanes via polar reciprocity with respect to the unit paraboloid identifies the convex hull of a point set with the lower envelope of the set of dual hyperplanes.

Definition 2.8. Given a set F of continuous functions having domain $\mathbb{R}^d$, the *lower envelope of F* is the pointwise minimum of the functions in F, that is, the function

$$\mathcal{E}_F(x) = \min_{f \in F} f(x), \text{ for all } x \in \mathbb{R}^d.$$

The *minimization diagram of F* is the partition of $\mathbb{R}^d$ into maximal connected relatively-open sets in each of which the value given by $\mathcal{E}_F$ is attained by a fixed set of functions in F.

Often, notation is abused identifying the lower envelope and the minimization diagram of a set of non-vertical hyperplanes with the lower envelope and the minimization diagram of the functions having them as graphs.

Definition 2.9. Given a set H of non-vertical hyperplanes in $\mathbb{R}^d$ and $k \in [|H|] \cup \{0\}$, a point $x \in \mathbb{R}^d$ is at *level k* with respect to H if it lies above or on exactly k hyperplanes in H. The *k-level* of the arrangement $\mathcal{A}$ of H is the union of the facets of $\mathcal{A}$ whose interior points have level k. The *$(\leq k)$-level* of $\mathcal{A}$ is the polyhedral complex consisting of the faces of $\mathcal{A}$ that lie on or below the k-level.

Voronoi diagrams and their variants are central objects in computational geometry. Details on many Voronoi-type diagrams can be found in a recent book by Aurenhammer, Klein and Lee [14]. In this thesis, we will use mainly classical Voronoi diagrams and power diagrams in the Euclidean space. We will also take advantage of the prolific literature on higher-order Voronoi diagrams and its relation to levels in arrangements of hyperplanes.

Definition 2.10. Given a finite point set $P \subset \mathbb{R}^d$, its Voronoi diagram is the polyhedral complex having cells

$$V_p = \{x \in \mathbb{R}^d : \|x - p\| \leq \|x - q\| \text{ for all } q \in P\}, \text{ for all } p \in P.$$

The *order-k Voronoi diagram* for $k \in [|P|]$ is the polyhedral complex having cells

$$V_S = \{x \in \mathbb{R}^d : \|x - p\| \leq \|x - q\| \text{ for all } p \in S \text{ and all } q \in P \setminus S\}, \text{ for all } S \subset P \text{ with } |S| = k.$$

We will implicitly use later the following observation that can be found, for instance, in [47].

Proposition 2.11. *Given a finite set of points $P \subset \mathbb{R}^d$, consider the linear functions*

$$h_p(x) = \|p\|^2 - 2\langle x, p\rangle, \text{ for all } p \in P.$$

For a fixed $k \in [|P|]$, the k-level of the arrangement of hyperplanes associated to these functions projects onto the order-k Voronoi diagram of P.

2.3 Power diagrams and constrained least-squares assignments

One of the common factors of the main problems considered in this work are least-squares assignments and their relation to power diagrams observed in [13]. We present this connection here in order to let the reader taste the flavor of this thesis, which mixes combinatorial and optimization formulations with more geometric and algorithmic accents.

Definition 2.12. The *power diagram* of a finite set of points $Q \subset \mathbb{R}^d$ (called *sites*) with assigned weights $w : Q \to \mathbb{R}^+$ is the polyhedral complex whose cells are

$$R_q = \{x \in \mathbb{R}^d : \|x - q\|^2 - w(d)^2 \leq \|x - q'\|^2 - w(q')^2 \text{ for all } q' \in Q\}, \text{ for } q \in Q.$$

For every $q \in Q$, the locus R_q is a polyhedron called the *region* of q.

Note that, for every $q \in Q$, the value $\|x-q\|^2 - w(d)^2$ is the power of the point x with respect to a circle centered at q and having radius $w(q)$. This is the reason why the power diagram is often defined for a set of circles instead of weighted points. For more details on this type of diagrams, see the survey in [12].

Given a finite point set S and a set Q of weighted points, we say that an assignment $\sigma : S \to Q$ is *induced by the power diagram* of Q if $\sigma^{-1}(q) \subset R_q$, for all $q \in Q$. These assignments are related to the constrained least-squares assignments, defined as follows.

Definition 2.13. Given a finite set of points Q, a function $c : Q \to \mathbb{N}$, and a set S of $\sum_{q\in Q} c(q)$ points, a *constrained least-squares assignment* for Q and S with *capacities* c is an assignment minimizing $\sum_{q\in Q} \|b - \tau(b)\|^2$ among all $\tau : S \to Q$ satisfying $|\tau^{-1}(q)| = c(q)$ for all $q \in Q$.

The following proposition compiles parts of the work in [13] that are relevant to our study.

Theorem 2.14 (Aurenhammer, Hoffmann and Aronov [13])**.** *Let Q be a finite set of points with weights $w : Q \to \mathbb{R}^+$ and let S be a point set.*

(i) Any assignment $\sigma : S \to Q$ induced by the power diagram of Q is a constrained least-squares assignment for Q and S with capacities $c(q) = |\sigma^{-1}(q)|$, for all $q \in Q$.

(ii) Conversely, if π is a constrained least-squares assignment for Q and S with capacities c, then there exist weights w such that π is induced by the power diagram of Q weighted by w.

The analogous result can be stated replacing S by a continuous measure. In particular, the measure could be uniform in a polytope and the capacities would then be a partition of its volume. As a consequence, the following theorem can be easily derived.

Theorem 2.15 (Aurenhammer, Hoffmann and Aronov [13])**.** *Let Q be a finite set of points.*

(i) For any finite set S of points and function $c : Q \to \mathbb{N}$ such that $\sum_{q\in Q} c(q) = |S|$, there exist weights $w : Q \to \mathbb{R}^+$ such that the power diagram of Q weighted by w induces an assignment $\sigma : S \to Q$ with $|\sigma^{-1}(q)| = c(q)$, for all $q \in Q$.

(ii) For any polytope P and function $c : Q \to \mathbb{R}^+$ such that $\sum_{q\in Q} c(q) = \mathrm{Vol}(P)$, there exist weights $w : Q \to \mathbb{R}^+$ such that the power diagram of Q weighted by w induces an assignment $\sigma : P \to Q$ with $\mathrm{Vol}(\sigma^{-1}(q)) = c(q)$, for all $q \in Q$.

This result is considered a Minkowsky-type theorem due to its relation to Minkowsky's problem for polytopes. This asks whether, given a list of unit vectors $u_1, \ldots, u_k \in \mathbb{R}^d$ and a set of associated values $\alpha_1, \ldots, \alpha_k \in \mathbb{R}^+$, there exist a polytope having facets $f_1, \ldots, f_k$ such that u_i is normal to f_i and f_i has volume α_i, for all $i \in [k]$. The answer is affirmative provided that $\alpha_1 u_1 + \cdots + \alpha_k u_k = 0$ and that the vectors $u_1, \ldots, u_k$ are spanning. Moreover, if there is a polytope satisfying the conditions, it is unique up to translation. Several proofs of this nice result can be found, for instance, in [78] and references therein.

If $k = |Q| \leq |S| = n$, a partition as indicated in Theorem 2.15-(i) can be computed in $O(k^2 n \log n)$ time by an algorithm given in [6]. For the special case $|S| = |Q| = n$ and $c(q) = 1$ for all $q \in Q$, the problem can be formulated as a linear sum assignment problem (described in Section 8.4) and can be solved using the Hungarian method in $O(n^3)$ time.

I

Recursive regularity and related problems

3

Introduction to Part I

In this chapter, we first motivate the study of regular (and recursively-regular) subdivisions and, afterwards, we review some facts that will be used or referred to later.

3.1 Motivation

Regular polyhedral complexes appear in a wide variety of situations. The minimization diagram of a set of linear functions, whose regularity follows almost directly from the definition, is a common instance. Power diagrams are regular complexes as well. It is not hard to see that an arrangement of hyperplanes is a regular subdivision as well; it is the projection of the lower envelope of the dual of a zonotope [47]. Yet another remarkable example is the Delaunay triangulation of a point set. A surprising connection is the Maxwell-Cremona correspondence [82], which relates the regularity of a planar graph to its rigidity as a framework.

Regular subdivisions are quite well-understood even in higher dimensions. Although, as shown by Santos [98], not all the triangulations of a point set in dimension five and higher are connected via flips, regular triangulations are, as reviewed in the following section. Another remarkable result, which holds in any dimension, is that regular subdivisions contain no cycles in the visibility relations in the sense of [48] (see Section 3.3).

On the other hand, not so much is known about non-regular subdivisions. Several generalizations of regularity have been studied in order to better understand them. For instance, the subdivisions induced by the projection of a polytope onto another polytope, introduced by Billera, Filliman and Sturmfels [23], have been extensively studied together with their variants.

Since we will need later several basic results on regular subdivisions, we summarize next the relevant facts and notation. See the book by De Loera, Rambau and Santos [45] for a detailed discussion on this topic.

3.2 Regular subdivisions and the secondary polytope

We present here an alternative definition of regular subdivision of a point set, which will simplify the notation and the proofs in this part of the thesis. In this section, for a point $a \in \mathbb{R}^d$ and a scalar $\lambda \in \mathbb{R}$ we denote by $\binom{a}{\lambda} \in \mathbb{R}^{d+1}$ the tuple (thought as a point) resulting from adding the coordinate λ to a.

Definition 3.1. Let $A \subset \mathbb{R}^d$ be a finite set of points. A subdivision $\mathcal{S}$ of A is *regular* if there exists a *height function* $\omega : A \to \mathbb{R}$ such that each face of $\mathcal{S}$ is the projection of a face in the lower convex hull of

$$A^\omega = \left\{ \begin{pmatrix} a \\ \omega(a) \end{pmatrix} : a \in A \right\}.$$

The function ω will be identified with the vector $\omega \in \mathbb{R}^A$. The notation A^ω will be used as a function of a point set A and a height function or vector ω. Given a cell $C \in cells(\mathcal{S})$, we will also use the notation

$$A^\omega|_C = \left\{ \begin{pmatrix} a \\ \omega(a) \end{pmatrix} : a \in A \cap C \right\}.$$

The following proposition gives local conditions for testing the regularity of a subdivision.

Proposition 3.2. *(Folding form [45]) Let $A \subset \mathbb{R}^d$ be a finite set of points. A polyhedral subdivision $\mathcal{S}$ of A is regular if there exists a height function $\omega : A \to \mathbb{R}$ such that:*

(1) for every cell $C \in cells(\mathcal{S})$, the points of $A^\omega|_C$ lie in a hyperplane (coplanarity condition), *and*

(2) for every wall $W = C \cap D$, where $C, D \in cells(\mathcal{S})$, the point $\binom{a}{\omega(a)}$ lies strictly above the hyperplane containing $A^\omega|_D$, for all $a \in A \cap (C \setminus D)$ (local folding condition).

We next analyze the conditions in Proposition 3.2 and show how they translate into linear equations and inequalities. We refer to [45] for more details.

Note that the coplanarity condition for a cell can be translated into a set of linear homogeneous equations in the heights of its vertices. Indeed, it is enough to choose an affine basis for each cell and require that each set resulting from extending this basis with a vertex in the cell is affinely dependent. Hence, all the coplanarity conditions together restrict the set of possible height functions ω to a linear subspace of $\mathbb{R}^n$.

Consider now the local folding condition for a wall $W = C \cap D$ with $C, D \in cells(\mathcal{S})$. Let $B = \{b_1, \ldots, b_{d+1}\}$ be a set of spanning vertices of D, and let $a \in A \cap (C \setminus D)$. The local folding condition for W can be expressed as

$$\begin{vmatrix} 1 & \ldots & 1 \\ b_1 & \ldots & b_{d+1} \end{vmatrix} \begin{vmatrix} 1 & \ldots & 1 & 1 \\ b_1 & \ldots & b_{d+1} & a \\ \omega(b_1) & \ldots & \omega(b_{d+1}) & \omega(a) \end{vmatrix} > 0. \tag{3.1}$$

By developing the last row of the second determinant, it becomes clear that this condition is a linear homogeneous strict inequality in the heights of the lifted points. Therefore, the local folding conditions for all the walls of a subdivision $\mathcal{S}$ define together a relatively open cone in the subspace determined by the coplanarity conditions.

Definition 3.3. The *regularity system* of a subdivision is the collection of equations and inequalities resulting from its coplanarity and local folding conditions. The *weak regularity system* of a subdivision is the system resulting of replacing the strict inequality in (3.1) with a weak inequality. The *secondary cone* is the set of solutions of the weak regularity system.

Note that the regularity system can be defined for coarsenings of polyhedral complexes, even if they are not polyhedral complexes (that is, if the "faces" fail to be convex or the tessellation is not face-to-face). Most of the definitions and statements presented here can be easily generalized to the case where the initial object A is a set of vectors instead of points. In such a case, the cells of the complex are cones forming a polyhedral fan whose 1-faces are rays with directions taken from A.

The local folding (and coplanarity) conditions lose then the row of ones of both determinants appearing in (3.1) and also one column each, since affine bases are replaced with linear bases. We will use the term subdivision in an ambiguous manner to stress this fact and focus on point-set subdivisions in the proofs. Details on this more general framework can be found in [45].

Regular subdivisions were first studied by Gelfand, Kapranov and Zelevinsky [63], who introduced the secondary fan and the secondary polytope. These two objects encode the combinatorics of the refinement poset of the regular subdivisions of a point set. We next give the necessary definitions to state their main results.

Definition 3.4. The GKZ-*vector* $\alpha(\mathcal{T})$ of a triangulation $\mathcal{T}$ of a finite point set A is the vector $\alpha(\mathcal{T}) \in \mathbb{R}^A$ whose a-th component is

$$\sum_{\substack{C \in cells(\mathcal{T}) \\ C \ni a}} \mathrm{Vol}(C).$$

The convex hull $\Sigma(A) \subset \mathbb{R}^A$ of all vectors $\alpha(\mathcal{T})$ over all triangulations $\mathcal{T}$ of A is an $(n-d-1)$-dimensional polytope called the *secondary polytope of* A.

Theorem 3.5 (Gelfand, Kapranov and Zelevinsky [63])**.** *The secondary cones of the regular triangulations of a d-dimensional point set A define a* $(n{-}d{-}1)$*-dimensional complete polyhedral fan (called the* secondary fan of A*). The secondary fan of A is the normal fan of* $\Sigma(A)$*.*

As a consequence, the vertices of $\Sigma(A)$ correspond to regular triangulations of A and the edges of $\Sigma(A)$ correspond to flips between regular triangulations. This proves, in particular, that the regular triangulations of A are connected in the graph of flips.

3.3 Edelsbrunner's acyclicity theorem

We state in this section a theorem by Edelsbrunner on regular subdivisions. The theorem asserts that a regular subdivision must be acyclic according to the *in-front* relation, which we define next.

Definition 3.6. Let x be a point in $\mathbb{R}^d$ and $S, T \subset \mathbb{R}^d$ be two disjoint convex sets. We say that S is *in front of* T with respect to x if there is an open halfline ℓ starting at x so that $S_0 = \ell \cap S \neq \varnothing$, $T_0 = \ell \cap T \neq \varnothing$ and every point of S_0 lies between x and any point of T_0.

This relation is called the *in-front* relation (from x), which is well-defined and antisymmetric because of the convexity of S and T. The relation can be defined for a direction as well, when x is considered to lie at infinity. It can be extended to all the faces of a polyhedral complex. To do that, the relation between faces is inherited from the relation between their relative interiors, which are pairwise-disjoint. The definition is similarly extended to polyhedral fans.

Definition 3.7. A polyhedral complex is said to be *cyclic* in a direction v (or from a point x) if the in-front relation induced by v (or x) on its open cells contains a cycle. The complex is called *acyclic* if it is not cyclic from any point or direction.

A cyclic polyhedral complex is shown in Figure 3.1, together with a point x from which it is cyclic. The arrows represent the half-lines certifying that C_1 is in-front of C_2, C_2 is in-front of C_3 and C_3 is in-front of C_1, if the relation is taken with respect to x.

We are finally ready to reproduce the theorem from Edelsbrunner.

Theorem 3.8 (Acyclicity Theorem [48])**.** *Regular polyhedral complexes are acyclic.*

In fact, Edelsbrunner proved something stronger: that the in-front relation is acyclic for all relatively-open faces of a regular polyhedral complex.

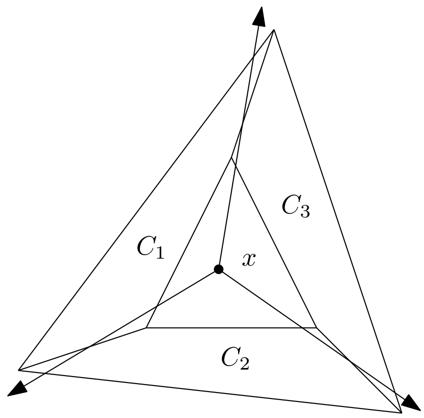


Figure 3.1: A cyclic polyhedral subdivision.

3.4 The Maxwell-Cremona correspondence

We provide in this section an introduction to tensegrity theory in the plane. Tensegrity theory studies the rigidity properties of frameworks made of bars, cables and struts from a formal point of view. The Maxwell-Cremona correspondence maps every equilibrium stress of a planar framework to a 3-dimensional polyhedral surface. We next develop the necessary concepts to formalize this correspondence.

An *abstract framework* $G = (V; B, C, S)$ is a graph on the vertex set $V = \{v_1, \dots, v_n\}$ whose edge set E is partitioned into sets B, C and S. The edges in B are called *bars*, the ones in C are called *cables* and the ones in S are called *struts*. They represent links supporting any stress, non-negative stresses and non-positive stresses, respectively. A *(tensegrity) framework* (in $\mathbb{R}^2$) is an abstract framework together with an embedding of the vertices $p : V \to \mathbb{R}^2$ where we put $p(v_i) = p_i$, for $i \in [n]$. The framework will be denoted by $G(p)$ and p will be thought of as a point $(p_1, \dots, p_n) \in \mathbb{R}^{2n}$. We can consider the configuration space of $G(p)$ to be

$$\begin{aligned} X(p) = \{(x_1, \dots, x_n) \in \mathbb{R}^{2n} : & \|x_i - x_j\| = \|p_i - p_j\|, \text{ for all } v_iv_j \in B; \\ & \|x_i - x_j\| \leq \|p_i - p_j\|, \text{ for all } v_iv_j \in C; \\ & \|x_i - x_j\| \geq \|p_i - p_j\|, \text{ for all } v_iv_j \in S\}. \end{aligned} \tag{3.2}$$

That is, $X(p)$ is the set of embeddings of G preserving the length of the bars, making the lengths of the cables no longer and the lengths of the struts no shorter than their lengths induced by p.

Definition 3.9. A tensegrity framework $G(p)$ is *rigid* in $\mathbb{R}^d$ if there exists an open neighborhood $U \subset \mathbb{R}^{2n}$ of p such that $X(p) \cap U = M(p) \cap U$, where

$$M(p) = \{(x_1, \dots, x_n) \in \mathbb{R}^{2n} : \|x_i - x_j\| = \|p_i - p_j\|, \text{ for all } i, j \in [n]\}$$

is the manifold of rigid motions associated to p.

In other words, a framework is rigid if its only motions respecting the constraints (3.2) are the motions that rigidly move the whole framework. The study of the quadratic constraints in the definition of $X(p)$ can be complicated. Because of this, the notion of infinitesimal rigidity was introduced, which captures the rigidity constraints up to the first order. Consider the system of linear equations and inequalities obtained by differentiating the constraints in (3.2). If the solutions of the system correspond only to differentials of motions in the Euclidean group, the framework is *infinitesimally rigid*. It is known that infinitesimal rigidity implies rigidity and that the converse is in general not true.

Definition 3.10. Given a framework $G(p)$, we say that $\omega : E \to \mathbb{R}$ is a *proper (equilibrium) stress* for $G(p)$ if the following conditions hold:

(1) $\omega(v_iv_j) = 0$ if $v_iv_j \notin E$.

(2) $\omega(v_iv_j) \geq 0$ if $v_iv_j \in C$.

(3) $\omega(v_iv_j) \leq 0$ if $v_iv_j \in S$.

(4) Every $v_i \in V$ is *in equilibrium*. That is, $\sum_{v_j \in V} \omega(v_iv_j)(p_j - p_i) = 0$.

We say that ω is *strictly proper* if the stresses on all cables and struts are non-zero.

Intuitively, ω is a proper equilibrium stress for $G(p)$ if the forces exerted by the edges (represented by ω) on the vertices add up to zero, taking into account that cables can support only non-negative stresses and struts can support only non-positive ones. Clearly, the stress assigning zero to all the edges is proper. This stress is called the *trivial stress*.

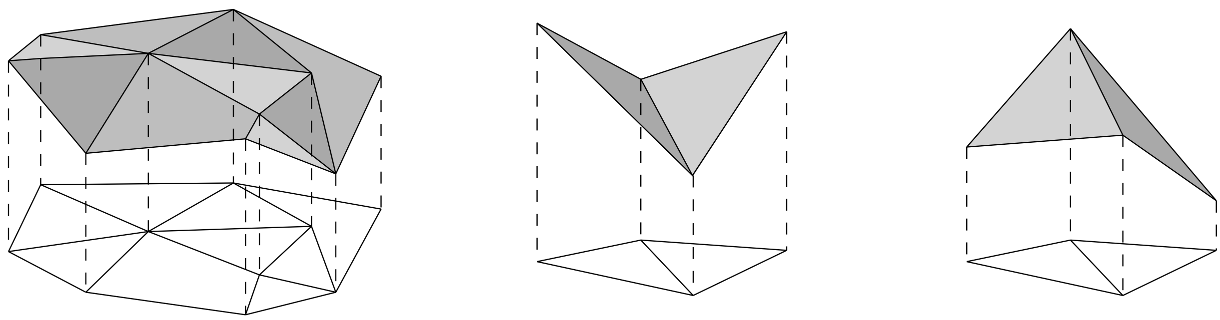


Figure 3.2: A polyhedral surface projecting onto a framework (left), a valley (center) and a mountain (right).

We state now a the Maxwell-Cremona correspondence. We refer to Figure 3.2 for a visual aid, and to [41] for more details.

Theorem 3.11 (Maxwell-Cremona correspondence)**.** *Let G be an abstract framework and $G(p)$ be a planar straight-line realization of G. There is a bijection between proper stresses for $G(p)$ and polyhedral terrains (with one arbitrarily chosen but fixed face at height zero) projecting on $G(p)$, where positive stress values correspond to valleys, negative stress values correspond to mountains and zero stress values correspond to flat edges in the lifting.*

Definition 3.12. A *spider web* is a framework (in $\mathbb{R}^2$) whose graph is connected, consisting only of cables, and with the vertices in the convex hull pinned down (that is, in equilibrium by definition).

The two following results relate equilibrium stresses of a framework with its rigidity and infinitesimal rigidity.

Lemma 3.13 (Connelly [40])**.** *If a spider web has a strictly proper stress, then it is rigid.*

Lemma 3.14 (Roth and Whiteley [95])**.** *If a tensegrity framework is infinitesimally rigid, then it has a strictly proper stress.*

3.5 Outline

The refinement of a regular subdivision is not necessarily regular. We will prove that, even if the restriction of the refinement to each of the cells of the original subdivision is regular, the resulting subdivision may be non-regular. However, the result of iterating regular refinement operations is a *recursively regular subdivision*. These subdivisions are the object of study of a big portion of Chapter 4. They arise from the observation that some theorems and algorithms exploit regularity in a way that can be transformed into a recursive scheme. We show that recursively-regular subdivisions generalize regular subdivisions while remaining acyclic in Edelsbrunner's sense. However, we prove the existence of recursively-regular triangulations belonging to different connected components of the flip graph of a point set, which suggests that the class is, in some sense, meaningfully larger than the class of regular subdivisions. We also show that one level of recursion (in the definition of recursive regularity) is not enough to capture all recursively-regular subdivisions.

Next, we address the problem of finding a simple characterization for recursively-regular subdivisions. We give an algorithm that decides whether a polyhedral subdivision is recursively-regular by computing a sequence of *finest regular coarsenings*. We believe this notion is of independent interest, since it maps every subdivision to the regular subdivision which is, in a specific sense, the most similar one.

Chapter 5 studies two illumination problems, providing new results in dimensions three and higher thanks to the theory developed in the preceding chapters.

In Chapter 6, we present other applications of the derived results.

4

The finest regular coarsening and the regularity tree

In this chapter, we study the finest regular coarsening of a subdivision, which we will use afterwards to define the regularity tree. Finally, we will introduce the class of recursively-regular subdivisions and analyze some of its properties.

Roughly speaking, the finest regular coarsening of a subdivision is the finest among all the coarsenings of the subdivision that are regular. One should note that it is not obvious whether this object is well-defined. We show first that this is indeed the case. We do it observing that merging two cells of a subdivision corresponds to converting a local folding condition into a coplanarity condition and, furthermore, this transformation can be done by simply replacing the strict inequality by an equation with the same coefficients. In other words, we are looking for the smallest set of inequalities we need to “relax” in order to make a given system compatible.

The first section of this chapter is concerned with a (we assume) well-known fact of linear algebra, for which we could not find a reference. We include it for completeness and because it definitely provides an insight into the problem. Later, in Proposition 4.18, we will give an algorithm to compute the finest regular coarsening (or a point in the relative interior of a polyhedral cone given by its hyperplane description), whose correctness will be implied be the following discussion.

4.1 A detour through linear algebra

We start by introducing some notation regarding systems of equations and inequalities.

Definition 4.1. Let $M \in \mathbb{R}^{m\times n}$ be a matrix with row vectors $s_1, \dots, s_m \in \mathbb{R}^n$. The *system of* M, denoted by $S(M)$, is the system

$$S(M) : \begin{cases} Mx > 0 \\ x \in \mathbb{R}^n. \end{cases} \tag{4.1}$$

Given $E \subset [m]$, we use $S^{\geq}(M, E)$ to denote the system

$$S^{\geq}(M,E) : \begin{cases} \langle s_i, x\rangle \geq 0, \text{ for all } i \in E \\ \langle s_j, x\rangle > 0, \text{ for all } j \in [m] \setminus E \\ x \in \mathbb{R}^n. \end{cases}$$

Given $E \subset [m]$, the *system of* M *relaxed by* E, denoted by $S(M, E)$, is the system

$$S(M,E) : \begin{cases} \langle s_i, x\rangle = 0, \text{ for all } i \in E \\ \langle s_j, x\rangle > 0, \text{ for all } j \in [m] \setminus E \\ x \in \mathbb{R}^n. \end{cases}$$

Literally, the adjective "relaxed" would better fit $S^{\geq}(M,E)$ but the following proposition shows that the two systems are equivalent in the cases we are interested in. The purpose of this section is to show that, given a matrix $M \in \mathbb{R}^{m\times n}$, there is a unique set $E \subset [m]$ of minimum cardinality such that $S^{\geq}(M,E)$ has a solution, and that this set can be easily found. If $S(M)$ is already compatible, it is clear that $E = \varnothing$ is the unique such set. Otherwise, we show that the problem can be transformed into an equivalent one.

Proposition 4.2. *Let $M \in \mathbb{R}^{m\times n}$ be such that $S(M)$ is incompatible, and let $E \subset [m]$ be a set of minimum cardinality such that $S^{\geq}(M,E)$ is compatible. Then, $S^{\geq}(M,E)$ and $S(M,E)$ have the same set of solutions.*

Proof. It is clear that the set of solutions of $S^{\geq}(M,E)$ contains the set of solutions of $S(M,E)$. Assume x_0 is a solution of $S^{\geq}(M,E)$ and is not a solution of $S(M,E)$. This means that at least one of the inequalities indexed by E is strictly satisfied by x_0. If $E_0 \neq \varnothing$ is the set of such inequalities, then x_0 is a solution of $S(M,E\setminus E_0)$. Since $E_0 \subset E$, this contradicts the assumed minimality of E. □

The previous observations motivate the following definition.

Definition 4.3. Given $M \in \mathbb{R}^{m\times n}$, the *minimum relaxation set* of the system $S(M)$, denoted by $E(M)$, is the intersection of all the sets $E \subset [m]$ such that $S(M,E)$ is compatible. The *minimum relaxation* of the system $S(M)$ is the system $S(M,E(M))$

We will prove that the system of M relaxed by $E(M)$ is compatible. Hence, it will be clear that it is the (unique) set of minimum cardinality that needs to be relaxed in $S(M)$ in order to make the system compatible.

For our purposes it is easier to argue in terms of the dual problem. The two following lemmas will be later used to prove the theorem of this section.

Definition 4.4. Given $M \in \mathbb{R}^{m\times n}$, the *dual system of* $S(M)$, denoted by $S^*(M)$, is the system

$$S^*(M) : \begin{cases} M^\top y = 0 \\ y \geq 0,\ y \neq 0 \\ y \in \mathbb{R}^m. \end{cases}$$

A system and its dual are related by the following special case of the Farkas Lemma.

Lemma 4.5 (Gordan's Theorem)**.** *Given $M \in \mathbb{R}^{m\times n}$, the system $S(M)$ is compatible if and only if the dual system $S^*(M)$ is incompatible.*

This result can be read in the following way: there is no solution for the original system if and only if there exists a non-zero non-negative linear combination y_0 of some inequalities leading to the contradiction "$0 > 0$". Such a combination y_0 is called a *contradiction cycle* and can be interpreted as a solution to the dual system.

It is convenient to prove first the following lemma, which translates Gordan's Theorem to the case where also linear homogeneous equations are included in the system. Before stating it, we need one more definition.

Definition 4.6. Given $M \in \mathbb{R}^{m\times n}$, and $E \subset [m]$, the *dual system of* $S(M,E)$, denoted by $S^*(M,E)$, is the system

$$S^*(M,E) : \begin{cases} M^\top y = 0 \\ y = (y_1,\dots,y_m) \in \mathbb{R}^m \\ y_i \geq 0\,, \text{ for all } i \in [m]\setminus E, \text{ and} \\ \text{there exists } \ j \in [m]\setminus E \text{ such that } y_j > 0. \end{cases}$$

This definition coincides with Definition 4.4 for $E = \varnothing$. We prove the previous lemma by reducing it to Gordan's Theorem. It is also possible to prove it by linear programming duality.

Lemma 4.7 (Extension of Gordan's theorem)**.** *Given $M \in \mathbb{R}^{m\times n}$, and $E \subset [m]$, the system $S(M,E)$ is compatible if and only if the dual system $S^*(M,E)$ is incompatible.*

Proof. If $E = \varnothing$, the statement of the theorem is Gordan's theorem. Hence, assume without loss of generality that $E = [j]$ for some $j \in [m]$. We will reduce this case to Gordan's theorem. Assume further that the span L of the first j row vectors is k-dimensional and that the first k rows span L. Let M' be the matrix resulting of excluding from M the rows indexed by $[j] \setminus [k]$. The set of solutions of $S(M',[k])$ is exactly the same as the set of solutions of $S(M,[j])$. On the other hand, we show next that the system $S^*(M,[j])$ has a solution if and only if the system $S^*(M',[k])$ has. Although the dimension of the domain of the linear map associated to $(M')^\top$ is smaller than the one of $M^\top$, a solution to $S^*(M',[k])$ can be extended to a solution of $S^*(M,[j])$ by setting the remaining coordinates to zero. That is, the dimension of the kernels of the linear maps associated to both matrices are the same.

Let now y_0 be a solution to $S^*(M,[j])$. We can obtain another solution having the coordinates indexed by $[j] \setminus [k]$ equal to zero, by expressing the columns of M indexed by $[j] \setminus [k]$ as linear combinations of the columns indexed by $[k]$, and modifying the coefficients in $[k]$ accordingly. In this way, ignoring the coordinates indexed by $[j] \setminus [k]$, we obtain a solution to $S^*(M',[k])$.

Henceforth, we will assume then that $E = [k]$ and that the first k rows of M are linearly independent. The set of equations in $S(M,[k])$ restricts then the variables to an $(n-k)$-dimensional linear subspace of $\mathbb{R}^n$, for $0 \le k < n$. We can then find an invertible $n \times n$ matrix T such that the space defined by the equations of $S(MT,[k])$ is the one having the first k coordinates equal to zero. Since T is invertible, $S(M,[k])$ has a solution if and only if the system $S(MT,[k])$ has.

On the other hand, $S^*(M,[k])$ and $S^*(MT,[k])$ have the same set of solutions because T is invertible and, thus, $M^\top y = 0$ if and only if $T^\top M^\top y = 0$ for all $y \in \mathbb{R}^m$.

Assume now that

$$MT = \begin{pmatrix} \mathrm{Id}_{k\times k} & 0_{k\times(n-k)} \\ R & N \end{pmatrix},$$

with $R \in \mathbb{R}^{k'\times k}$, $N \in \mathbb{R}^{k'\times(n-k)}$, and $k' = m - k$. The systems $S(MT,[k])$ and

$$S(N) : \begin{cases} Ny > 0 \\ y \in \mathbb{R}^{n-k} \end{cases}$$

have both a solution or none of them have. This is because a solution to the first must have the first k coordinates equal to zero and, thus, the last $n-k$ coordinates must be a solution of the second. Conversely, a solution to the second system can be extended to a solution of the first by just adding k zero coordinates.

Similarly, the system $S^*(MT,[k])$ and the system

$$S^*(N) : \begin{cases} N^\top y = 0 \\ y \in \mathbb{R}^{k'} \\ y \ge 0,\ y \ne 0 \end{cases}$$

have both a solution or none of them has. Indeed, the restriction of a solution of $S^*(MT,[k])$ to the last k' coordinates must be a solution of $S^*(N)$. For the other direction, let

$$z_0 = \begin{pmatrix} -R^\top y_0 \\ y_0 \end{pmatrix},$$

where y_0 is a solution of $S^*(N)$. Since

$$(MT)^\top z_0 = \begin{pmatrix} \mathrm{Id}_{k\times k} & R^\top \\ 0_{(n-k)\times k} & N^\top \end{pmatrix} \begin{pmatrix} -R^\top y_0 \\ y_0 \end{pmatrix} = \begin{pmatrix} -R^\top y_0 + R^\top y_0 \\ 0 + N^\top y_0 \end{pmatrix} = 0,$$

the vector z_0 is a solution of $S^*(MT,[k])$. We finish the proof by applying Gordan's theorem to $S(N)$ and $S^*(N)$. □

Theorem 4.8. *Let $M \in \mathbb{R}^{m\times n}$ be a matrix. The system $S(M, E(M))$ is compatible.*

Proof. If $S(M)$ is compatible, then $E(M) = \varnothing$, since $S(M,\varnothing) = S(M)$ is compatible. Assume, then, that $S(M)$ is not compatible. Lemma 4.5 provides a solution y_0 of the dual system $S^*(M)$. Let $E_0 \neq \varnothing$ be the set of positive coordinates of y_0. We will show that $E_0 \subset E$ for any $E \subset [m]$ with $S(M,E)$ compatible. Indeed, if we assume the contrary, then y_0 is also a solution of $S^*(M,E)$, and we can derive that $S(M,E)$ is not compatible, forcing the contradiction. As any set E making $S(M,E)$ compatible must contain E_0, we focus now on the system $S(M,E_0)$, which has strictly fewer inequalities than S. If it is compatible, then obviously $E(M) = E_0$. Otherwise, we keep transforming inequalities into equations iterating the previous arguments (using Lemma 4.7) until a compatible system is found. The process finishes because $S(M,[m])$ is compatible. Since all the elements we introduce in our relaxation set must be necessarily in any set making the system compatible, it is clear that the set obtained at the end is $E(M)$. □

We can interpret the previous lemma in terms of contradictions as well. Assume that $S(M)$ is not compatible and, therefore, there exists a solution y_0 of its dual $S^*(M)$. The vector y_0 represents a positive linear combination of the inequalities in $S(M)$ leading to the contradiction "$0 > 0$". If all the inequalities involved in the contradiction (that is, corresponding to positive coordinates of y_0) are converted into equations, the same combination will lead to the valid equation "$0 = 0$". In other words, y_0 will no longer be a solution of the dual system. However, the equations we created in the relaxation process may be now used to derive other contradictions. Indeed, they confer more freedom to the dual system, since the corresponding dual variables are now allowed to take negative values. The corresponding intuition is that the relation "$>$" is maintained if one adds two strict inequalities. The relation is however not necessary preserved if one inequality is subtracted from another. Nevertheless, equations can be either added to or subtracted from a strict inequality and the a strict inequality can still be derived. Note also that if we do not relax all the inequalities involved, the contradiction cannot be avoided: the linear combination of equations will give an equation, to which we can add a strict inequality to derive "$0 > 0$" again. In other words, if the system contains a contradiction, all the involved inequalities must be relaxed in order to obtain a compatible system.

An intuitive explanation of why the minimum relaxation is unique can be easily obtained if one looks at the complementary problem. That is, given a system of weak homogeneous linear inequalities, decide which is the maximum number of constraints that can be satisfied strictly. The set of solutions of the system is a closed polyhedral cone K. If x_0 is a point in the relative interior of K, then the desired maximal set of constraints consists exactly of those constraints that are strictly satisfied by x_0. That is, finding this minimum relaxation is equivalent to finding a point in the relative interior of a (possibly not full-dimensional) polyhedral cone given by a set of weak (possibly redundant) inequalities.

The results of this section can be generalized to systems of non-homogeneous inequalities. The main difference would be that $S(M,[m])$ is not necessarily compatible in this case and, therefore, there may be no relaxation at all. Nevertheless, whenever there exists a compatible relaxed system, the minimum relaxation is well-defined and can be computed in the same way as in the homogeneous case.

4.2 The finest regular coarsening of a subdivision

The algebra in the previous section will make it very easy to show that there exists a (well-defined) finest regular coarsening of a polyhedral subdivision. We next introduce some additional terminology concerning coarsenings.

Definition 4.9. Given a polyhedral subdivision $\mathcal{S}$, and a coarsening $\mathcal{S}'$ of $\mathcal{S}$, the *coarsening function* (from $\mathcal{S}$ to $\mathcal{S}'$) is the function $\kappa : \mathit{cells}(\mathcal{S}) \to \mathit{cells}(\mathcal{S}')$ such that $C \subset \kappa(C)$, for all $C \in \mathit{cells}(\mathcal{S})$. Given two coarsenings $\mathcal{S}_1$ and $\mathcal{S}_2$ of $\mathcal{S}$, we say that $\mathcal{S}_1$ is *finer* than $\mathcal{S}_2$ if $\mathcal{S}_2$ is a coarsening of $\mathcal{S}_1$. A coarsening is *proper* if it has strictly fewer cells than the original subdivision. The *trivial coarsening* is the one merging all the cells into a single one.

Using the definitions in [45], the refinement relation induces a partial order on the set subdivisions. Furthermore, the restriction of this partial order to regular subdivisions is a lattice. This lattice is isomorphic to the face lattice of the secondary polytope of the point set. However, as far as we know, not much work has been done concerning coarsenings of non-regular subdivisions. The finest regular coarsening goes in that direction, and permits to map every non-regular subdivision to a regular one which is, in a specific sense, the most similar to it.

Definition 4.10. The *finest regular coarsening* of a subdivision $\mathcal{S}$ of a point set A is the subdivision obtained by the projection of the lower hull of A^{ω_0}, where ω_0 is a solution of the minimum relaxation of the regularity system of $\mathcal{S}$.

The next theorem justifies the name in the previous definition.

Theorem 4.11. *Let $\mathcal{S}$ be a polyhedral subdivision, and $\mathcal{S}_0$ be the finest regular coarsening of $\mathcal{S}$. Then, $\mathcal{S}_0$ is a regular coarsening of $\mathcal{S}$ and all the regular coarsenings of $\mathcal{S}$ are coarsenings of $\mathcal{S}_0$.*

Proof. Observe first that the relaxation of a constraint corresponding to a local folding condition in the regularity system of $\mathcal{S}$ converts this condition into a coplanarity condition (up to a non-zero scalar factor) for the two cells incident to the wall. Thus, the new system is equivalent to the regularity system of the polyhedral complex resulting from merging the two cells of $\mathcal{S}$ involved in the constraint. That is, coarsenings of $\mathcal{S}$ have regularity systems that are relaxations of the regularity system of $\mathcal{S}$. In addition, a coarsening is regular if and only if its regularity system has a solution. Hence, $\mathcal{S}_0$ is a coarsening, is regular and it is the regular coarsening that merges the minimum number of cells, that is, the finest one. □

It will come in handy later to say that a subdivision is *completely non-regular* if its finest regular coarsening is its trivial coarsening. This implies, in particular, that every wall of the subdivision can appear in a contradiction cycle of its regularity system.

Relation to the secondary polytope. Note that, once we are convinced that the finest regular coarsening is well-defined for any subdivision of a finite point set A, it is easy to derive an alternative definition in terms of the secondary polytope $\Sigma(A)$ of a point set A. Considering the definitions of subdivision and refinement used in [45], the faces of $\Sigma(A)$ correspond to regular subdivisions of A and their inclusion relations correspond to coarsening relations. The vertices of $\Sigma(A)$ are the GKZ-vectors of all regular triangulations of A. Non-regular triangulations however have GKZ-vectors that are not vertices of $\Sigma(A)$. Moreover, for non-regular triangulations the function mapping a triangulation to its GKZ-vector may not even be injective [45]. In any case, the normal cone $\alpha(\mathcal{T})$ of a triangulation $\mathcal{T}$ in $\Sigma(A)$ is isomorphic to the secondary cone of $\mathcal{T}$ [63]. It is then not surprising that the finest regular coarsening of a triangulation $\mathcal{T}$ corresponds to the subdivision associated to the smallest face in $\Sigma(A)$ containing $\alpha(\mathcal{T})$. As stated in [45], the secondary cone can be similarly defined for general subdivisions (not only for triangulations). This cone will be contained in the linear subspace L of the height-functions space defined by the coplanarity conditions. Of course, the cone will be also contained in the affine hull H of the secondary fan, which is $(n-d-1)$-dimensional. If the dimension of $L \cap H$ is $k < n-d-1$, the subdivision is regular if its secondary cone is k-dimensional as well. Then, any height function in the relative interior of the secondary cone will certainly produce the subdivision. If the subdivision $\mathcal{S}$ is not regular, this cone will not be full-dimensional with respect $L \cap H$.

4.3 The regularity tree and recursively-regular subdivisions

In this section we introduce the class of *recursively-regular subdivisions* of a point set. Roughly speaking, recursively-regular subdivisions are subdivisions that can be decomposed, via a regular coarsening, into recursively-regular pieces. First, we define a partial order on the power set of cells of a given subdivision, which will be called the *regularity tree*. This tree decomposes the subdivision in terms of its (recursive) finest regular coarsenings. This object reflects some structure in the set of non-regular subdivisions of a given point set and hints an algorithmic procedure to decide whether a subdivision is recursively regular. We derive later some properties of the recursively-regular subdivisions and leave to Chapter 6 the motivation for the study of these objects.

Definition 4.12. A polyhedral subdivision $\mathcal{S}$ is *recursively regular* if either

(a) $\mathcal{S}$ is regular, or

(b) There exists a proper non-trivial coarsening $\mathcal{S}'$ of $\mathcal{S}$ with coarsening function κ such that

  (1) $\mathcal{S}'$ is a regular subdivision, and
  (2) for each cell $C \in \mathcal{S}'$, $\kappa^{-1}(C)$ is recursively regular.

Note that the previous definition can be extended to polyhedral fans. We will use the notation $\mathfrak{R}(A)$ to refer to the set of recursively-regular subdivisions of a point configuration A. The class of all recursively-regular subdivisions of any point set will be denoted by $\mathfrak{R}$. We will show that $\mathfrak{R}$ is larger than the class of regular subdivisions and that the regularity tree can even have arbitrary depth.

To proceed, we need to introduce some notation and technical definitions. Recall that according to our conventions in Section 2.1, given a subset $\mathcal{C}$ of *cells*$(\mathcal{S})$, we denote by $|\mathcal{C}|$ the ground set $\cup_{C \in \mathcal{C}} C$ covered by these cells. Similarly, if $\mathcal{S}$ is a subdivision, $|\mathcal{S}|$ will denote the union of the cells of $\mathcal{S}$.

Definition 4.13. A *subdivision tree* of a subdivision $\mathcal{S}$ of a point set A is a rooted tree whose vertices are subsets of *cells*$(\mathcal{S})$, whose root is *cells*$(\mathcal{S})$, and such that if the children of $\mathcal{C}$ are $\mathcal{C}_1, \ldots, \mathcal{C}_l$, then $|\mathcal{C}_1|, \ldots, |\mathcal{C}_l|$ are the cells of a polyhedral subdivision of $A \cap |\mathcal{C}|$. A subdivision tree is called *regular* if the subdivisions of $A \cap |\mathcal{C}|$ used to split the nodes of the tree are all regular.

Note that a subdivision is recursively-regular if and only if it has a regular subdivision tree. However, a subdivision can have many subdivision trees, and even many regular subdivision trees. We next define a canonical one, which will be later used to decide if a subdivision is recursively regular.

Definition 4.14. The *regularity tree* of the subdivision $\mathcal{S}$ is the subdivision tree created by the following recursion.

(a) If a subdivision $\mathcal{S}$ is regular or its finest regular coarsening is trivial, its regularity tree is the tree whose single node is $|\mathcal{S}|$.

(b) The regularity tree of a non-regular subdivision $\mathcal{S}$ with a non-trivial finest regular coarsening $\mathcal{S}_0$ is obtained by appending to its trivial coarsening the regularity tree of $\kappa^{-1}(C)$, for each cell $C \in \mathcal{S}_0$.

Figure 4.1 exhibits an example of a regularity tree. The figure shows a triangulation in $\mathfrak{R}$ which needs two levels of recursion to fit the definition of recursively-regular subdivision. The coordinates of this example and a proof that the finest regular coarsening of the depicted subdivision is the subdivision defined by the second level of the tree are provided in Appendix A. Note that the example consists of a "pinwheel" triangulation (refining the "mother of all examples" in [45]) inserted into a triangle of a bigger copy of the pinwheel triangulation. The insertion procedure can be repeated recursively to obtain a triangulation whose regularity tree has a number of levels linear in the number of vertices.

Note that the leaves of the regularity tree of $\mathcal{S}$ are a partition of *cells*($\mathcal{S}$). We say that a *leaf* $\mathcal{C}$ *is regular*, respectively *completely non-regular*, if the subdivision induced by $\mathcal{S}$ on $\mathcal{C}$ is regular, respectively completely non-regular. By our definition, there are two possibilities for the leaves of the the regularity tree: they are either regular or completely non-regular.

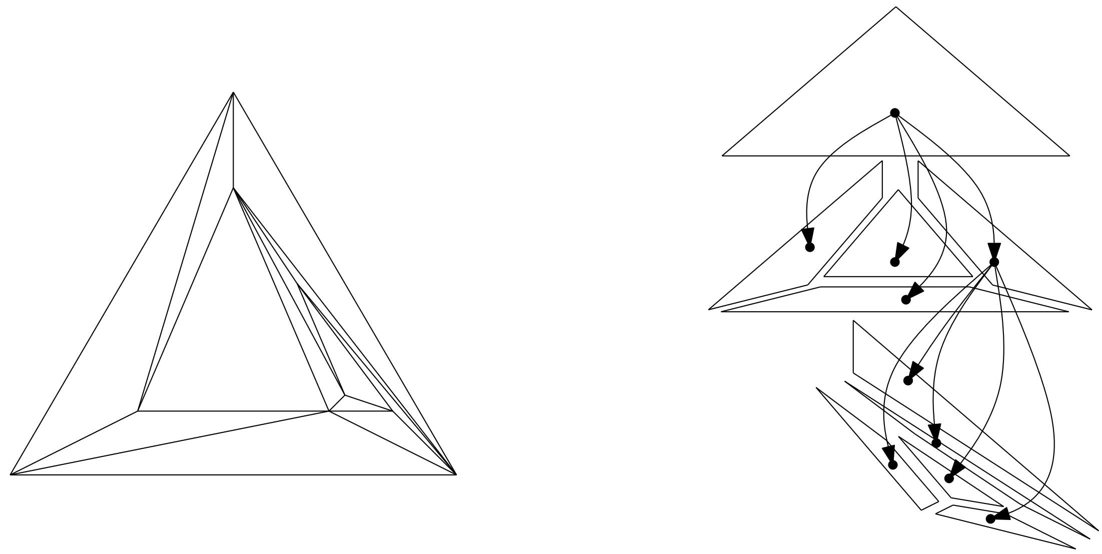


Figure 4.1: A recursively-regular subdivision and a sketch of its regularity tree.

The basic result of this section is the following theorem, which relates the regularity tree and the recursive regularity of a subdivision.

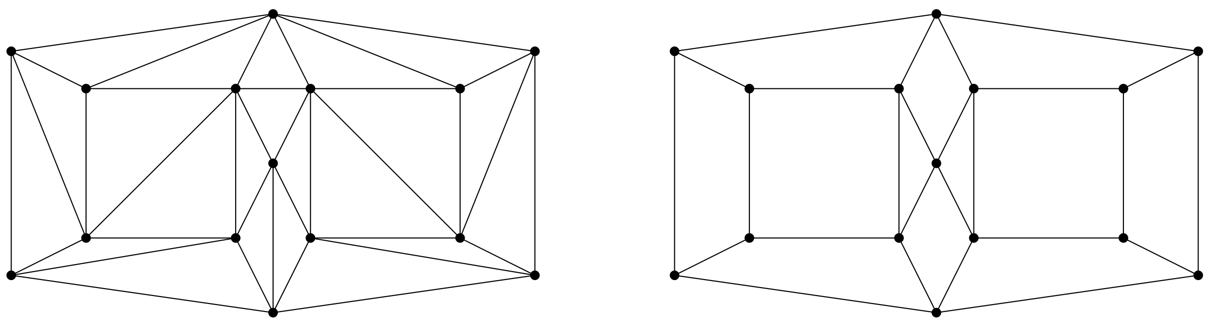
Theorem 4.15. *A polyhedral subdivision $\mathcal{S}$ is recursively regular if and only if the leaves of its regularity tree are regular.*

Proof. If all leaves are regular, the regularity tree itself certifies the recursive regularity of $\mathcal{S}$, proving the *if* direction.

For the *only if*, it will be proved that the leaves of the regularity tree of any subdivision in $\mathfrak{R}$ are regular. We do this by induction on the number of cells of the subdivision. The base case is when the subdivision consists of a single cell C. In this case, the only leaf of its regularity tree is C, which is regular.

For the inductive step, let $\mathcal{S}$ be in $\mathfrak{R}$, and assume that the regularity tree of any recursively-regular subdivision with fewer cells than $\mathcal{S}$ has regular leaves. Let $\bar{\mathcal{S}}$ be a regular coarsening with coarsening function $\bar{\kappa}$ splitting $\mathcal{S}$ into smaller recursively-regular subdivisions. Indeed, by definition, there is a regular subdivision tree of $\mathcal{S}$ representing a set of coarsenings certifying that it is recursively regular. We want to show that the regularity tree is a valid certificate as well. The second part of Theorem 4.11 asserts that $\bar{\mathcal{S}}$ is a coarsening of the finest regular coarsening $\mathcal{S}_0$ of $\mathcal{S}$. This implies that each cell $C \in \mathcal{S}_0$ is contained in some cell $C' \in \bar{\mathcal{S}}$, such that $\mathcal{S}$ restricted to C' is recursively regular. Note that refinement relations and regularity behave well with respect to restrictions to polyhedra. That is, the subdivision obtained by intersecting all the faces of a regular subdivision with a polyhedron is regular as well, and the intersection of a coarsening with a polyhedron is a coarsening of the original subdivision, intersected with the polyhedron. Hence, recursive regularity behaves well with respect to restriction to polyhedra and it follows that $\mathcal{S}$ restricted to $C \subset C'$ is recursively regular. By induction hypothesis, the leaves of the regularity tree of $\mathcal{S}$ restricted to C are regular, for every $C \in \mathit{cells}(\mathcal{S}_0)$. Since the leaves of the regularity tree of $\mathcal{S}$ are the leaves of the regularity trees of its children, this completes the proof. □

We present now some properties of the recursively-regular subdivisions. Recall that a subdivision $\mathcal{S} \subset \mathbb{R}^d$ said to be *acyclic* if there exists no point $q \in \mathbb{R}^d$ such that the visibility relation defined on the faces of $\mathcal{S}$ (detailed in Section 3.3) from q contains a cycle.



(a) Non-regular triangulation

(b) Finest regular coarsening

Figure 4.2: Two-dimensional recursively-regular and non-regular triangulation.

Proposition 4.16. *Let A be a finite point set. Every regular subdivision of A is recursively-regular. Every recursively regular subdivision of A is acyclic. The converse of the previous statements is in general not true.*

Proof. Note first that regular subdivisions are in $\mathfrak{R}(A)$ by directly applying the definition. We will prove that any $\mathcal{S}$ in $\mathfrak{R}$ must be acyclic by induction on its number of cells. For the base case, we use that a single-cell subdivision is always acyclic. If $\mathcal{S}$ has more than one cell, we distinguish two cases. If $\mathcal{S}$ itself is regular, then Theorem 3.8 shows that it must be acyclic. Otherwise, there exists a regular coarsening $\mathcal{S}'$ of $\mathcal{S}$ with coarsening functions κ. Assume for the sake of contradiction that $\mathcal{S}$ contains a cycle and consider the image by κ of the involved faces. If this image contains more than one cell, the cycle induces another one in $\mathcal{S}'$ leading to a contradiction with its assumed regularity. So the cycle must be contained in $\kappa^{-1}(C)$ for a single cell $C \in \mathcal{S}'$. But $\kappa^{-1}(C)$ is a recursively-regular subdivision having strictly fewer cells than $\mathcal{S}$ and, hence, acyclic by the induction hypothesis.

Figure 4.2(a) shows a non-regular triangulation that belongs to $\mathfrak{R}$. A certificate for its non-regularity is included in Appendix B, while that it belongs to $\mathfrak{R}$ is straightforward after observing that the coarsening in Figure 4.2(b) is regular. For the properness of the second inclusion, we

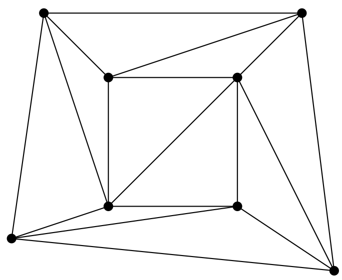


Figure 4.3: An acyclic triangulation that is not recursively regular.

refer to the example shown in Figure 4.3, which shows an acyclic subdivision that does not belong to $\mathfrak{R}$. Its acyclicity and that it does not belong to $\mathfrak{R}$ will be certified in Appendix C. □

The next proposition illustrates that $\mathfrak{R}$ includes some "pathological" triangulations. More precisely, we will show that there are triangulations in $\mathfrak{R}$ that are not connected in the graph of flips of its vertex set. To prove this, we will simply show that the non-regular triangulations used by Santos in [98] are indeed in $\mathfrak{R}$.

Proposition 4.17. *There exists a point set $A \subset \mathbb{R}^5$ whose recursively-regular triangulations are not connected by flips.*

Proof. Santos constructs in [98] a set of triangulations $\mathfrak{T}$ of a five-dimensional point set A that are pairwise disconnected in its graph of flips. We show that all the triangulations in $\mathfrak{T}$ are recursively regular. The convex hull of A is a prism over a polytope Q called the 24-cell. The polytope Q is four-dimensional and has 24 facets, which are regular octahedra. All the triangulations in $\mathfrak{T}$ are refinements of the prism $\mathcal{P}$ (in the sense of [45, Definition 4.2.10]) over a subdivision $\mathcal{B}$ of $A \cap Q$. The subdivision $\mathcal{B}$ is a central subdivision of Q, it is thus regular (see [45, Section 9.5]) and consists of 24 pyramids over octahedra. Therefore, the prism $\mathcal{P}$ is regular as well (because the prism over a regular subdivision is regular [45, Lemma 7.2.4]). Each cell of $\mathcal{B}$ is triangulated in a specific way for every triangulation in $\mathfrak{T}$. However, since a triangulation of a pyramid is regular if and only if the triangulation induced on its base is regular (see [45, Observation 4.2.3]), and the bases of the pyramids are regular octahedra (which are known to have only regular triangulations), the restriction any triangulation in $\mathfrak{T}$ to any cell of $\mathcal{B}$ is regular. Hence, the restriction of any triangulation in $\mathfrak{T}$ to every cell of $\mathcal{P}$ is regular as well, since a triangulation of a prism over a simplex is regular ([45, Section 6.2]). Thus, every triangulation in $\mathfrak{T}$ is recursively regular. Indeed, each triangulation in $\mathfrak{T}$ is a refinement of a regular subdivision $\mathcal{P}$, and its restriction to any cell of $\mathcal{P}$ is regular. □

In fact, the previous proposition shows that there is a point set A with at least 12 triangulations in $\mathfrak{R}(A)$ that are pairwise disconnected and disconnected from any regular triangulation in the graph of flips of A, as observed in [98].

In the following section, we construct algorithms which are of interest to apply the theoretical results described so far.

4.4 Algorithms

We study how the problem of finding the minimum relaxation of a system, which is equivalent to finding a point in the relative interior of polyhedral cone given by a set of inequalities.

This problem has been rediscovered several times and an approach to it can be found for instance in [15]. We give an algorithm starts from a compatible dual system and moves towards compatible primal using the machinery introduced in Section 4.1.

Proposition 4.18 (*folklore*)**.** *Let $M \in \mathbb{R}^{m\times n}$. The minimum relaxation set $E(M)$ of system $S(M)$ (consisting of m linear inequalities on n variables) can be computed solving at most m linear programs in m variables and n constraints.*

Proof. In the proof of Theorem 4.8 we show that if a coordinate can take a positive value in a solution of $S^*(M)$, then $E(M)$ must include the corresponding index. It is also argued that the minimal relaxation of the system can be obtained by incrementally applying this criterion. We will convert here this incremental procedure into an algorithm that uses linear programming. We start setting $E = \varnothing$ and we insert into the set E the indices that must belong to $E(M)$. The compatibility of $S(M, E)$ is related to its dual system $S^*(M, E)$ in the sense of Definition 4.6. To check whether $S^*(M, E)$ has a solution, we solve the linear program

$$\mathit{maximize} \sum_{i\in[m]\setminus E} y_i,$$

subject to the linear constraints given by the system $S^*(M, E)$ plus the condition $\sum_{i\in[m]} y_i \leq 1$, which ensures that the maximum is bounded. For the ease of argumentation, we add a linear inequality in order to make the dual feasible region bounded. If the optimum value is zero, then none of the variables in $[m]\setminus E$ can attain a positive value under the constraints of $S^*(M, E)$, and thus it is incompatible. Consequently, the system $S(M, E)$ is compatible and E is the minimal relaxation (because we have only added an index to E if we know that the index must be in any relaxation set making the system compatible). The converse is also true: if the function takes a positive value, a non-empty set of variables $E_0 \subset [m] \setminus E$ take positive values. Therefore, as argued in the proof of Theorem 4.8, we know that $E_0 \subset E(M)$. Hence, we add the indices of E_0 to E and iterate the process. At each iteration, we discover at least one new index that belongs to $E(M)$ and, thus, at most m iterations are needed. □

Observe that a primal variant of the algorithm can be easily devised adding slack variables. That is, the strict inequality $\langle s_i, x\rangle > 0$ of $S(M)$ is transformed into an equation $\langle s_i, x\rangle - \lambda_i = 0$ and a weak inequality $\lambda_i \geq 0$, for all $i \in [m]$. Then, we add all the slack variables λ_i in the objective function. Easy arguments (dual to the ones we used in Theorem 4.8) would ensure that if a slack variable cannot be positive for an intermediate problem, then it can never be, and thus we can set it to zero and maximize the remaining ones similarly.

With help of the previous theorem, it becomes easy to prove that the finest regular coarsening of a subdivision can be efficiently computed. The best known bound for linear programming is polynomial only if the total number of bits L needed to encode the coefficients is counted as input size (as in the Turing machine model). Alternatively, we can say that a linear program can be solved in time polynomial in the number of variables and L. We choose this second option to formalize the following bound.

Corollary 4.19. *Let $\mathcal{S}$ be subdivision of a point set A in any fixed dimension and let L be the total number of bits necessary to encode the coordinates of A. The finest regular coarsening of $\mathcal{S}$ can be computed in time polynomial in $|A|$ and L.*

Proof. It follows from the definition of the finest regular coarsening that it can be determined by finding a point ω_0 in the relative interior of the secondary cone of $\mathcal{S}$, computing the point set A^{ω_0} and its convex hull to finally project its lower faces. However, it is easier to iteratively construct it following the algorithm in Proposition 4.18 to find the minimum relaxation set of its regularity system. Whenever a constraint is relaxed (a dual variable is unrestricted), we merge the cells sharing the corresponding wall. We perform the merge operation symbolically by giving a common label to the merged cells. When the iteration ends, we construct the cells of the finest regular coarsening by computing the convex hull of the vertices of the cells with the same label. Since we assume that the dimension is constant and the vertices of the finest regular coarsening are a subset of A, the construction of the cells can be done in polynomial time.

Note that the coefficients of the linear program come from d-dimensional determinants on the coordinates of points in A. Therefore, the number of bits needed to encode them is polynomial in L. In each iteration, a linear program with a number of constraints proportional to $|A|$ and as many variables as walls in $\mathcal{S}$ is solved. Therefore, the whole algorithm takes polynomial time in $|A|$ and L. □

Some improvements can probably be done when computing the finest regular coarsening by taking into account the special structure of the regularity system of $\mathcal{S}$. In particular, the matrix M associated to the system is sparse and its structure is related to the combinatorics of the subdivision. Each row, corresponding to a wall, has at most $d + 2$ non-zero coefficients. In addition, d of the involved vertices can be taken to be an affine basis for the corresponding wall. Then, the corresponding d coefficients are positive while the other two are negative. If $\mathcal{S}$ is a triangulation, this means that each vertex that is involved in a folding condition appearing in a contradiction cycle must be involved in another condition of the contradiction cycle. Moreover, if a vertex belongs to the wall corresponding to a condition in a contradiction cycle, it must appear in another condition of the cycle associated to a wall that does not contain it (because the contributions to a vertex in a dual solution must add up to zero). If $\mathcal{S}$ is not a triangulation, a similar combinatorial property still holds.

The statement in the previous corollary is not trivial because there exist subdivisions, even in the plane, with a linear number of simultaneous flips [62]. That is, a linear number of pairs of cells that can be independently merged or not. Consequently, these subdivisions have an exponential number of minimal coarsenings that one might need to test for regularity. The scenario seems even worse when it comes to recursive regularity. Fortunately, as a consequence of Theorem 4.15, this can indeed be decided in polynomial time using the procedure in Corollary 4.19.

Proposition 4.20. *Let $\mathcal{S}$ be subdivision of a point set A in fixed dimension and let L be the total number of bits necessary to encode the coordinates of A. Whether $\mathcal{S}$ is recursively regular can be decided in time polynomial in $|A|$ and L.*

Proof. Theorem 4.15 ensures that we only need to compute the regularity tree of $\mathcal{S}$ to decide whether $\mathcal{S}$ belongs to $\mathfrak{R}$ or not. This is done by computing the finest regular coarsening of subdivisions of some subsets of A. Each time we go down a level in the tree, there is one wall in the finest regular coarsening that was not in any previous finest regular coarsenings. Therefore, if we charge the computation of the finest regular coarsening to this wall, we can conclude that the number of computations is bounded by the number of walls in $\mathcal{S}$, which is polynomial if d is considered to be a constant. □

5

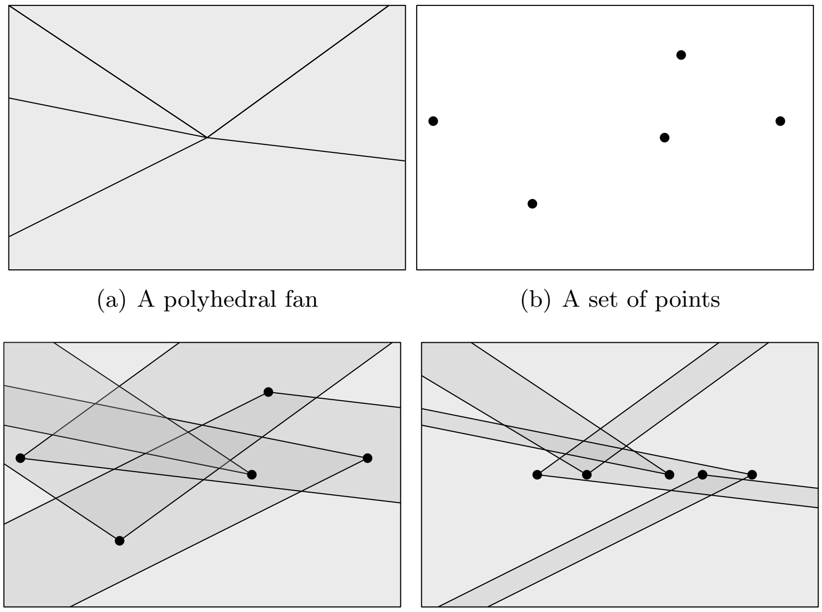
Illumination by floodlights in high dimensions

In the last decades, a wide collection of problems have been studied concerning illumination or guarding of geometric objects. The first Art Gallery problem posed by Klee asked simply how many guards are necessary to guard a polygon. Since then, considerable research has addressed several variants of this problem, such as finding watchman routes or illuminating sets of objects. A remarkable group of problems arises when the light sources (or the surveillance devices) do not behave in the same way in all directions. In the major part of the literature, these problems are studied only in the plane. A compilation of results on this type of problem can be found in [105]. The problem we are interested in assumes that a light source can illuminate only a convex unbounded polyhedral cone. We are given the polyhedral cones available and a set of points representing the allowed positions for their apices. We can then choose the assignment of the floodlights to the points in order to cover some target set. In this chapter, the assignment will be required to be one-to-one and the floodlights will not be permitted to rotate.

The first problem we look at in this section is the *space illumination problem* in three or higher dimensions. Informally speaking, the problem asks if given a set of floodlights and a set of points there is a placement of the floodlights on the points such that the whole space is illuminated. Afterwards, we study the generalization to higher dimensions of the *stage illumination problem*, introduced by Bose, Guibas, Lubiw, Overmars, Souvaine and Urrutia [26].

5.1 Illuminating space

The results presented here use recursively-regular polyhedral fans. These objects are analogous to recursively-regular subdivisions of a point set with vectors instead of points as base elements. Definitions and basic properties concerning polyhedral fans, also called subdivisions of vector configurations, can be found in [45]. We next introduce some new definitions specific to this problem. For basic definitions, see Section 2.2. The *ground set* of a polyhedral fan $\mathcal{F}$, denoted by $|\mathcal{F}|$, is the union of all its cells. We say that a d-dimensional polyhedral fan is *complete* if its ground set is the whole space and that it is *conic* if the ground set is a pointed d-dimensional cone. Similarly, we will talk about the *complete case* and the *conic case* to refer to instances of the problem where the given fan is complete or conic, respectively. A facet of a fan will be called *interior* if it is not contained in the boundary of the ground set of the fan. A cone K is said to *contain a direction (or vector) v* if it contains the ray ρ_v starting at the apex of K and having direction (or direction vector) v. We will say that the direction is *interior to a cone* if ρ_v intersects the boundary of K only in its apex.



(a) A polyhedral fan (b) A set of points

(c) A covering assignment (d) Assignment for points on a line

Figure 5.1: Covering the plane by floodlights.

Definition 5.1. Let P be a polyhedron

$$P = \bigcap_{i \in I} \Pi_i^+,$$

where Π_i are the hyperplanes supporting the facets of P, for $i \in I$. The *reverse polyhedron* of P, denoted by P^-, is defined as

$$P^- = \bigcap_{i \in I} \Pi_i^-.$$

The *reverse fan* of a polyhedral fan $\mathcal{F}$ is the fan obtained by reversing all its faces. The *reverse cone* of a conic fan is the reversed set of its ground set.

Note that if P is a cone with apex at the origin, then $P^- = -P$.

Definition 5.2. Given a d-dimensional complete polyhedral fan $\mathcal{F}$ with n cells and a set of n points $P \subset \mathbb{R}^d$, we say that an assignment $\sigma : \mathit{cells}(\mathcal{F}) \to P$ is *covering* if it is one-to-one and

$$\bigcup_{C \in \mathit{cells}(\mathcal{F})} (C + \sigma(C)) \supset |\mathcal{F}|.$$

Note that the floodlights are only translated to the corresponding points and not rotated, as in other variants of the problem. Figure 5.1 contains illustrations of a two-dimensional polyhedral fan, a set of points, the translated cells associated to a covering assignment for the shown point set and for a set of points on a line.

We are now ready to state formally the *space illumination problem.* Given a d-dimensional polyhedral fan and a set of points in $\mathbb{R}^d$ we would like to know whether there is a covering assignment for that fan and the point set in the sense of Definition 5.2. Galperin and Galperin [61] proved that a covering assignment can be found if the fan is complete and regular, regardless of the given point set and in any dimension. In particular, the there is a covering assignment for a fan in the plane and any point set of the right cardinality.

Theorem 5.3 (Galperin, Galperin [61]; Rote [92])**.** *Let $\mathcal{F} \subset \mathbb{R}^d$ be a full-dimensional regular polyhedral fan consisting of n cells and $P \subset \mathbb{R}^d$ be a set of n points. There is a covering assignment for $\mathcal{F}$ and P.*

This last statement was rediscovered with a small variation in the formulation of the problem in [26], where an $O(n \log n)$ algorithm for finding a covering assignment is given as well. The conic case in the plane has also been considered with the extra assumption that the points are contained in the reverse cone of the fan. In this case, a covering assignment can be always found as well. However, if the points are not required to lie in the reverse cone, deciding the existence of a covering assignment becomes NP-hard even in the plane, since the problem is equivalent to the *wedge illumination problem* studied in [32]. It is worth mentioning the problem of illumination disks with a minimum number of points in the plane, studied by Fejes Tóth [56]. The notion of illumination in that work is not the usual one and the lights can be placed anywhere. Surprisingly enough, he used the properties of power diagrams to prove an upper bound on the number of needed points, the same diagrams used by Rote [92] to provide an alternative proof of Theorem 5.3.

We generalize first the conic case to higher dimensions and prove that it is sufficient for the fan to be recursively-regular to ensure the existence of a covering assignment for any point set in the reverse cone of the fan. Afterwards, we use this result to prove that Theorem 5.3 can be extended to recursively-regular fans in the complete case as well. Both generalizations are synthesized in the following statement (note that $(\mathbb{R}^d)^- = \mathbb{R}^d$ and there is thus no restriction for P in the complete case).

Theorem 5.4. *Let $\mathcal{F} \subset \mathbb{R}^d$ be a full-dimensional recursively-regular polyhedral fan consisting of n cells and $P \subset |\mathcal{F}|^-$ be a set of n points. There is a covering assignment for $\mathcal{F}$ and P.*

We prove first two simple technical lemmas. Recall that the *restriction* of a fan to a polyhedral cone K (having the same apex) is the fan obtained by intersecting every face of $\mathcal{F}$ with K.

Lemma 5.5. *A conic full-dimensional fan $\mathcal{F} \subset \mathbb{R}^d$ with $|\mathcal{F}| = K$ is regular if and only if $\mathcal{F}$ is the restriction to K of a complete regular fan.*

Proof. For the *only if* direction, assume that $\mathcal{F}$ is regular and, hence, there is a cone $\widetilde{K} \subset \mathbb{R}^{d+1}$ whose lower convex hull projects on $\mathcal{F}$. This cone can be written as

$$\widetilde{K} = \left(\bigcap_{i \in I^+} \Pi_i^+ \right) \cap \left(\bigcap_{i \in I^-} \Pi_i^- \right),$$

where Π^+ refers to the closed halfspace above the hyperplane Π and Π^- refers to the closed halfspace below Π; and I^+ is the set of indices such that $\widetilde{K} \subset \Pi_i^+$ and I^- is the set of indices such that $\widetilde{K} \subset \Pi_i^-$. By convention, $\widetilde{K}$ will be considered to lie below the vertical hyperplanes, and thus the indices of these hyperplanes are considered as part of I^-. Note that $\bigcap_{i \in I^+} \Pi_i^+$ is a cone whose faces project onto a complete fan $\mathcal{G}$, since the vertical direction is interior to it. Moreover, its restriction to $|\mathcal{F}|$ is $\mathcal{F}$.

To prove the *if* direction, assume that $\widetilde{L} \subset \mathbb{R}^{d+1}$ is a cone whose lower hull projects onto a complete fan $\mathcal{G} \subset \mathbb{R}^d$ and let $K = \bigcap_{i \in I} \Pi_i^+ \subset \mathbb{R}^d$ be a polyhedral cone. For every $i \in I$, let $\widetilde{\Pi}_i$ be the vertical hyperplane in $\mathbb{R}^{d+1}$ containing Π_i. Clearly the set $\widetilde{L} \cap (\bigcap_{i \in I} \widetilde{\Pi}_i^+)$ is a cone whose lower hull projects onto the restriction of $\mathcal{G}$ to K. □

The following technical lemma will be useful to extend Theorem 5.3 to the conic case and to recursively-regular fans. Given a complete fan $\mathcal{G}$ and a pointed cone K, the lemma relates the existence of a covering assignment for $\mathcal{G}$ to a covering property of the restriction $\mathcal{F}$ of $\mathcal{G}$ to K. More precisely, we show that the cells of $\mathcal{F}$ can cover a polyhedron Q resulting from shifting the hyperplanes defining K provided that the given point set lies in Q^- and that there is a covering assignment for this point set and $\mathcal{G}$.

Lemma 5.6. *Let $Q = \bigcap_{i\in I}(\Pi_i^+ + t_i)$ be a full-dimensional polyhedron, where $t_i \in \mathbb{R}^d$ for all $i \in I$. Let $\mathcal{G} \subset \mathbb{R}^d$ be a full-dimensional complete fan consisting of n cells, whose restriction $\mathcal{F}$ to $K = \bigcap_{i\in I}\Pi_i^+$ consists of n cells as well. If there is a covering assignment for $\mathcal{G}$ and a set $P \subset Q^-$ of n points, then the cells of $\mathcal{F}$ translated by the corresponding assignment cover Q.*

Proof. Let $\theta : cells(\mathcal{F}) \to cells(\mathcal{G})$ be the map such that $C = \theta(C) \cap K$ for all $C \in cells(\mathcal{F})$, and let $\sigma : cells(\mathcal{G}) \to P$ be a covering assignment. We want to show that

$$\bigcup_{C\in cells(\mathcal{F})} (C + \sigma(C)) \supset Q.$$

By hypothesis,

$$\bigcup_{C\in cells(\mathcal{F})} (\theta(C) + \sigma(C)) = \mathbb{R}^d \supset Q.$$

We are done if we can prove that $(C + p) \cap Q \supset (\theta(C) + p) \cap Q$ for all $p \in Q^-$ and for all $C \in cells(\mathcal{F})$. Note that $Q \subset \Pi_i^+ + p$ for any $p \in Q^-$ and for all $i \in I$, by definition of reverse polyhedron. Since

$$C = \theta(C) \cap K = \theta(C) \cap \left(\bigcap_{i\in I}\Pi_i^+\right),$$

it follows that

$$(C+p) \cap Q = \left[(\theta(C)+p) \cap \left(\bigcap_{i\in I}(\Pi_i^+ + p)\right)\right] \cap Q \supset (\theta(C)+p) \cap Q.$$

Therefore, the cells of $\mathcal{F}$ translated according to $\sigma \circ \theta$ cover Q. □

The following proposition is now easy to prove.

Proposition 5.7. *Let $\mathcal{F} \subset \mathbb{R}^d$ be a full-dimensional conic regular fan with $|\mathcal{F}| = K$ consisting of n cells and $P \subset K^-$ be a set of n points. There is a covering assignment for $\mathcal{F}$ and P.*

Proof. Lemma 5.5 provides us with a fan $\mathcal{G}$ whose restriction to K coincides with $\mathcal{F}$ and has the same number of cells. Theorem 5.3 applies to $\mathcal{G}$ and P. It only remains to invoke Lemma 5.6 to show that any covering assignment for $\mathcal{G}$ and P can trivially be translated into a covering assignment for $\mathcal{F}$ and P. □

We can now prove Theorem 5.4, which generalizes the result (and the proof) in [92].

Proof of Theorem 5.4. The proof proceeds recursively splitting the set of cells of $\mathcal{F}$, the space in $|\mathcal{F}|$ and the points of P into smaller problems. Before detailing the recursion, we introduce some notation and include a proof of a lemma by Rote.

Let $\mathcal{F}_0$ be the finest regular coarsening of $\mathcal{F}$, and $\kappa : cells(\mathcal{F}) \to cells(\mathcal{F}_0)$ be the associated coarsening function. Let

$$K = \bigcap_{C\in cells(\mathcal{F}_0)} \Pi_C^+$$

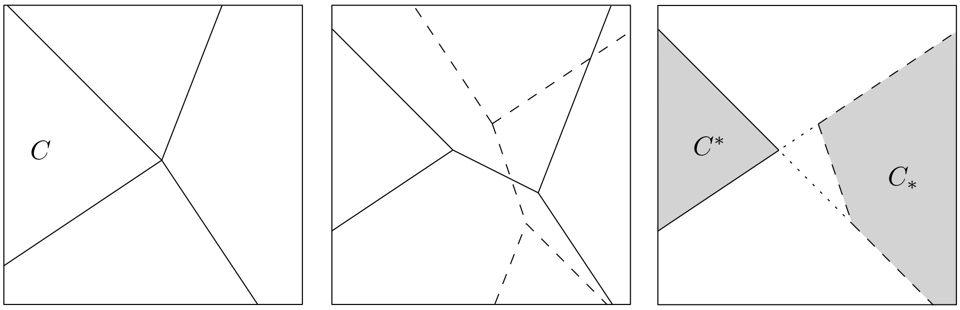


Figure 5.2: A fan $\mathcal{F}$ (left). An instance of $\varphi^*(\mathcal{F}, w)$ and $\varphi_*(\mathcal{F}, w)$ (center). A cell C_* contained in the reverse cone of C^*.

be a $(d+1)$-dimensional cone projecting onto $\mathcal{F}_0$, where the hyperplane Π_C supports the facet of K that projects onto C, for all $C \in \mathit{cells}(\mathcal{F}_0)$. Given a function $\omega : \mathit{cells}(\mathcal{F}_0) \to \mathbb{R}$, let the power diagram $\varphi^*(\mathcal{F}_0, \omega)$ be the (projection of the) upper envelope of the hyperplane arrangement obtained by vertically shifting the hyperplane Π_C by $\omega(C)$, for all $C \in \mathit{cells}(\mathcal{F}_0)$. Similarly, let $\varphi_*(\mathcal{F}_0, \omega)$ denote the lower envelope of these hyperplanes. Both power diagrams have as many cells as $\mathcal{F}_0$ and all of them are unbounded. In addition, the cells in these diagrams can be paired in a natural way with the hyperplane they come from. For simplicity of notation, let C^* denote the cell of $\varphi^*(\mathcal{F}_0, \omega)$ corresponding to C, and by C_* the corresponding cell of $\varphi_*(\mathcal{F}_0, \omega)$. These pairs of cells satisfy the following property, which is illustrated in Figure 5.2.

Lemma 5.8 (Rote [92])**.** *Every cell $C_* \in \varphi_*(\mathcal{F}, \omega)$ is contained in the reverse polyhedron of $C^* \in \varphi(\mathcal{F}, \omega)$.*

Proof. Choose an arbitrary cell C_* of $\varphi_*(\mathcal{F}, \omega)$. Let D_* be an adjacent cell and $W_* = C_* \cap D_*$ be their common wall. Consider also the wall $W^* = C^* \cap D^*$. Note that both W_* and W^* are supported by the hyperplane h, which is the projection of $(\Pi_C + \omega(C)) \cap (\Pi_D + \omega(D))$. Clearly $\Pi_C + \omega(C)$ is above $\Pi_D + w_D$ in one side of h while $\Pi_D + \omega(D)$ is above $\Pi_C + \omega(C)$ in the other side and, hence, C_* is contained in one side of h while D_* is contained in the other. Putting together the analogous observations for all other cells adjacent to C_*, we derive the desired statement for this (arbitrarily chosen) cell C. □

We continue the proof of Theorem 5.4. The main idea is to apply Theorem 2.15-(i) and find ω such that $\varphi_*(\mathcal{F}_0, \omega)$ leaves in C_* exactly $|\kappa^{-1}(C)|$ points of P, for every cell $C \in \mathit{cells}(\mathcal{F}_0)$. Then, we will cover each cell C^* of $\varphi^*(\mathcal{F}_0, \omega)$ with the floodlights of $\mathcal{F}$ contained in C and the points of P in C_*. If $\mathcal{F}_C = \kappa^{-1}(C) \cap \mathcal{F}$ is regular, we proceed as in the proof of Proposition 5.7, using Lemma 5.6 to construct an assignment that covers C^* with the points in $P \cap C_*$ and the floodlights of $\mathcal{F} \cap C$. If $\mathcal{F}_C = \kappa^{-1}(C) \cap \mathcal{F}$ is not regular but recursively-regular, we repeat the process recursively. That is, we split the points of $P \cap C^*$ with a power diagram associated to the finest regular coarsening $\mathcal{G}_0$ of $\mathcal{F}_C$. For each cell D of $\mathcal{G}_0$, we get points in $C_* \cap D_*$, which is contained in the reverse polyhedron of $C^* \cap D^*$. Hence, the recursion proceeds until the base case, where we can cover the target polyhedron with a regular fan from points in its reverse polyhedron using Lemma 5.6. □

The following definition will come in handy later.

Definition 5.9. Let $\mathcal{F} \subset \mathbb{R}^d$ be a full-dimensional polyhedral fan with n cells. $\mathcal{F}$ is *universally covering* if for any point set $P \subset \mathbb{R}^d$ of n points there exists a covering assignment for $\mathcal{F}$ and P.

After showing that all recursively regular fans are universally covering, one could imagine that all fans are so. We prove that this is not the case in dimension three and higher by showing that if a fan is cyclic in the sense described in Section 3.3, there is a point set for which there is no covering assignment. This statement will easily follow from Theorem 5.14. Before proving this theorem, we need to introduce a definition and state a technical lemma.

Definition 5.10. Let $\mathcal{F} \subset \mathbb{R}^d$ be a full-dimensional polyhedral fan, let $W \in \mathcal{F}$ be a facet incident with $C, D \in \mathit{cells}(\mathcal{F})$, and let v be a vector normal to W pointing from C to D. We say that an assignment $\sigma : \mathit{cells}(\mathcal{F}) \to \mathbb{R}^d$ satisfies the *overlapping condition* for W if $\langle(\sigma(C) - \sigma(D)), v\rangle \geq 0$.

Note that the previous condition is satisfied for a facet and an assignment if and only if the copies of the two cells sharing the facet translated to the assigned points have non-empty intersection. We provide now a proof for the following well-known facts.

Lemma 5.11. *Let $K \subset \mathbb{R}^d$ be a full-dimensional polyhedral cone.*

(i) Any line with direction interior to K has unbounded intersection with K.

(ii) Any line with direction not contained in K has bounded intersection with K.

Proof.

(i) Let $\ell = \{p + \lambda v : \lambda \in \mathbb{R}\}$ be a line passing through $p \in \mathbb{R}^d$ and with direction $v \in \mathbb{R}^d$ interior to $K = \{q + \sum_{i\in I} \alpha_i v_i : \alpha_i \geq 0 \text{ for all } i \in I\}$, where v_i for $i \in I$ are the extreme rays of K. Since v is interior to K, we can express $v = \sum_{i\in I} \gamma_i v_i$ with $\gamma_i > 0$ for all $i \in I$. The set of vectors $\{v_i : i \in I\}$ spans $\mathbb{R}^d$ because its positive span K is not contained in any proper affine subspace. Thus, we can express $p - q = \sum_{i\in I} \delta_i v_i$, where $\delta_i \in \mathbb{R}$ for all $i \in I$. Note now that

$$p + \lambda v = q + (p - q) + \lambda \sum_{i\in I} \gamma_i v_i = q + \sum_{i\in I} (\delta_i + \lambda\gamma_i) v_i,$$

which lies in K for all $\lambda \geq \max_{i\in I} \frac{-\delta_i}{\gamma_i}$.

(ii) We triangulate K into a finite number of simplicial cones and show that the intersection of ℓ with each cone is bounded. For a fixed simplicial cone K' with extreme rays $v_1, \ldots, v_d$, the direction v of the line can be expressed in a unique way as $v = \sum_{i\in[d]} \gamma_i v_i$ with $\gamma_i \in \mathbb{R}$ for all $i \in [d]$.

We will prove the contrapositive. Assume that there exists $\lambda_0 \in \mathbb{R}$ such that $p + \lambda v \in K'$, for all $\lambda \geq \lambda_0$. Since

$$p + \lambda v = p + \lambda \sum_{i\in[d]} \gamma_i v_i = q + \sum_{i\in[d]} (\lambda\gamma_i - \delta_i) v_i,$$

we have that $\lambda\gamma_i - \delta_i \geq 0$ for all $i \in [d]$ and for all $\lambda \geq \lambda_0$. Thus, $\gamma_i \geq 0$ for all $i \in [d]$, which implies that the direction of v is contained in K'. □

The next lemma follows easily.

Lemma 5.12. *Let $\mathcal{F} \subset \mathbb{R}^d$ be a full-dimensional polyhedral fan. A covering assignment for $\mathcal{F}$ must satisfy the overlapping condition for every interior facet of the fan.*

Proof. If the condition is not satisfied for the facet $H = C \cap D$, we consider a ray in a direction interior to H (for instance, the barycenter of its rays) and placed at the point $(\sigma(C) + \sigma(D))/2$. In view of Lemma 5.11-(ii), no cell of $\mathcal{F}$, except for C and D, can cover an unbounded part of this ray. In addition, none of these two cells intersect it. Therefore, since the ray is unbounded and we have finitely many cones, the ray cannot be completely covered. If the fan is complete, the proof is finished. Otherwise, we should note that the ray will eventually enter $|\mathcal{F}|$, since the direction of the ray is interior to an interior facet of $\mathcal{F}$ and, hence, interior to $|\mathcal{F}|$. □

The previous condition is not sufficient in general, not even in the plane. An exception is the case where all the points lie on a line, which is studied in the following lemma. Figure 5.1(d) shows a sketch of a covering assignment for the case under consideration.

Lemma 5.13. *Let σ : cells($\mathcal{F}$) $\to P$ be an assignment for a full-dimensional polyhedral fan $\mathcal{F} \subset \mathbb{R}^d$ and a point set $P \subset \ell \cap |\mathcal{F}|^-$, where ℓ is a line. If σ satisfies the overlapping condition, then it is a covering assignment.*

Proof. We prove first the complete case. Fix an orientation for ℓ and let v be a direction vector for it. In addition, we can assume without loss of generality that ℓ goes through the apex of $\mathcal{F}$. Consider any oriented line ℓ' with direction v. At infinity, ℓ' is covered by some (untranslated) cell C of $\mathcal{F}$. Hence, when C is translated to its assigned point of P, it still covers ℓ' at infinity because the translation is only in the direction v. Let $q \in \ell'$ be the point where ℓ' leaves C. If q is in the relative interior of (the translation of) a facet $W = C \cap D$, where $C, D \in$ *cells*($\mathcal{F}$), the overlapping condition for W (and the special position of P) ensures that ℓ' enters D before leaving C. Iterating this argument, we eventually reach a cell containing the direction $-v$ that covers the unbounded remainder of ℓ'. Thus, any line ℓ' with direction v and that intersects only d- and $(d-1)$-dimensional faces of the translated cells is completely covered. The union $\mathcal{U}$ of the remaining lines with direction v (that is, the lines intersecting some $(d-2)$-dimensional face of some translated cell) is a nowhere-dense set and thus is covered as well. Indeed, for every line $\hat{\ell} \in \mathcal{U}$ we can find a line not in $\mathcal{U}$ with direction v (and, hence, covered) arbitrarily close to $\hat{\ell}$. Since the cells are closed sets, the limit of a sequence of covered lines must be covered as well, and thus $\mathcal{U}$ is covered. Since any line with direction v is covered, $\mathbb{R}^d$ is completely covered.

Assume now that $\mathcal{F}$ is a conic fan with $K = |\mathcal{F}|$. Consider a line ℓ' with direction v that enters K through a facet. Let $C \in$ *cells*($\mathcal{F}$) be the cell containing this facet. Since $P \subset \ell \cap K^-$, the line ℓ' should enter the cell C translated to the corresponding point before entering K. The arguments for the complete case carry over until the line crosses (the translation of) a facet W of a cell D such that $W \subset \partial K$. Then, again the fact that $P \subset \ell \cap K^-$ implies that the ℓ had left K before. Therefore, if ℓ' is a line with direction v that avoids $(d-2)$-dimensional faces of the translated cells (and of K), then $\ell' \cap K$ is covered. A limit argument as in the complete case ensures that then all the lines with direction v has the portion intersecting K covered, and thus K is covered. □

We are now in a position to construct examples consisting of a fan and a point set for which there is no covering assignment.

Theorem 5.14. *Given a full-dimensional polyhedral fan $\mathcal{F} \subset \mathbb{R}^d$ with n cells and set of n points $P \subset \ell \cap |\mathcal{F}|^-$, where ℓ is a line, there is a covering assignment for $\mathcal{F}$ and P if and only if $\mathcal{F}$ is acyclic in the direction of ℓ.*

Proof. Provided that $\mathcal{F}$ is acyclic in the direction v of ℓ, we can construct a directed acyclic graph having the cells of $\mathcal{F}$ as vertices and an edge from D to C if the vector u normal to $W = C \cap D$ pointing from C to D satisfies $\langle u, v\rangle \geq 0$. If the order as the points $\sigma(C)$ appear on ℓ (for all $C \in$ *cells*($\mathcal{F}$)) respects the partial order represented by such a directed graph, then

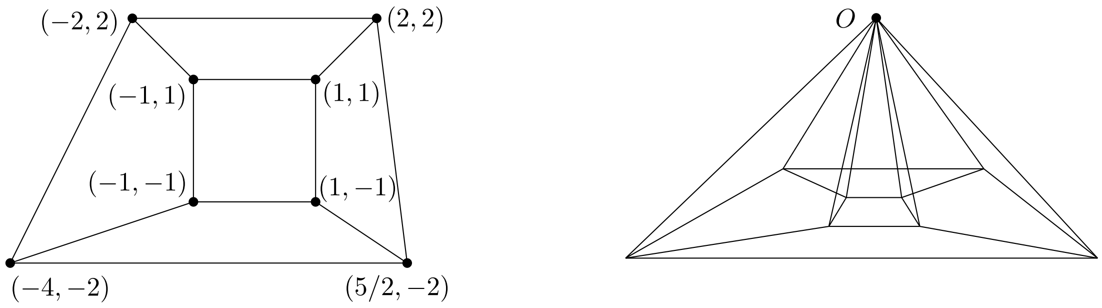


Figure 5.3: Acyclic, not recursively-regular subdivision (left) and the corresponding fan (right).

the overlapping condition holds for σ. Lemma 5.13 ensures that this condition is sufficient for the assignment to be covering.

We prove the other direction by contrapositive. If there is a visibility cycle $\tau = (C_1 \dots C_k)$ in the direction v (that is, C_i is in front of C_{i+1}, for all $i \in [k-1]$, and C_k is in front of C_1), there is a cycle in the order the points $\sigma(C_1), \dots, \sigma(C_k)$ should appear in the line, preventing the overlapping condition to be satisfied for all the facets of the fan. This has been proven to be necessary for the assignment to be covering. $\square$

If a covering assignment exists for a given point set in a line and a given fan, it can be computed in $O(n^2)$ time by performing a topological sort on the graph described in the proof of Theorem 5.14. Since the number of facets is bounded by n^2, the algorithm runs in the claimed time. Afterwards, it only remains to sort the points, which can be done in $O(n \log n)$ time. Moreover, the topological sort algorithm would detect if the graph has a cycle and, therefore, there is no covering assignment.

After understanding the previous theorem, one might be tempted to conjecture that being acyclic is equivalent to being universally covering. We exhibit next an example to show that this is not the case.

Proposition 5.15. *There exists full-dimensional polyhedral fan $\mathcal{F} \subset \mathbb{R}^3$ consisting of n cells, and a set of n points $P \subset |\mathcal{F}|^-$ for which there is no assignment satisfying the overlapping conditions. The fan has no cycle in any direction.*

Proof. We will provide a three-dimensional fan $\mathcal{F}$ with five cells and a point set $P \subset \mathbb{R}^3$ for which there is no covering assignment. More precisely, it can be shown that for each of the 5! possible assignments, one of the eight overlapping conditions is violated. To construct $\mathcal{F}$, take the subdivision sketched in Figure 5.3 (left) and embed it in the plane $\{(x, y, z) \in \mathbb{R}^3 : z = -1/8\}$. Take then the cones from the origin to each of the cells of this subdivision forming the fan displayed in Figure 5.3 (right).

Let P be the point set consisting of the points

$$p_1 = (29, 95, 89),\ p_2 = (55, 19, 92),\ p_3 = (54, 10, 82)$$
$$p_4 = (78, 2, 68), p_5 = (15, 40, 92).$$

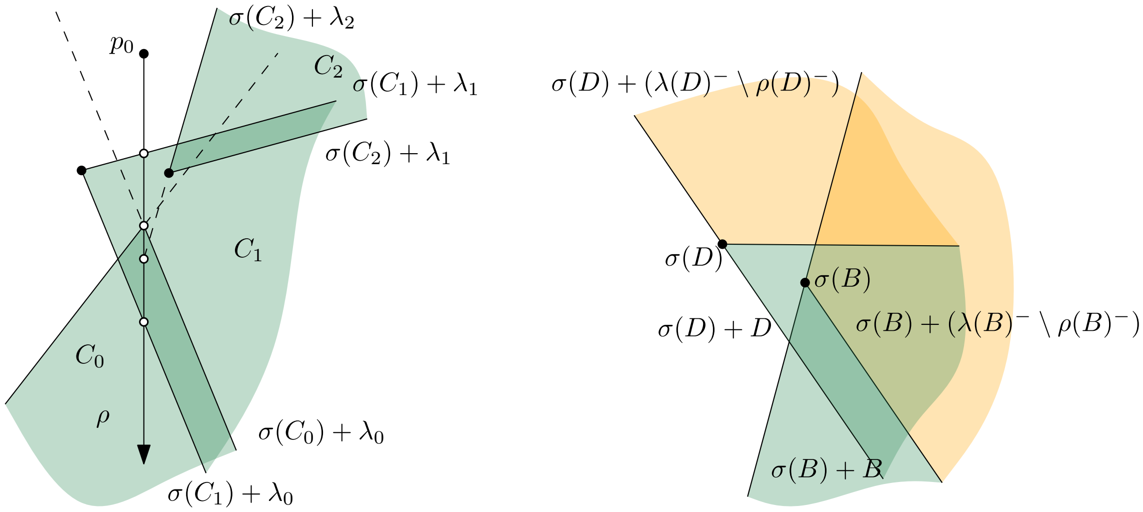
There is no assignment for this point set fulfilling all the overlapping conditions, as proved in Appendix D. The last statement together with Lemma 5.13 allow us to derive that there is no covering assignment for the given fan and the given point set. That there is no direction in which $\mathcal{F}$ is cyclic is also proven in Appendix D. $\square$

The point set P in the previous proof was found with the help of a computer. We generated many pseudo-random samples of five points in $\mathbb{R}^3$ trying different precisions for the coordinate generator and several parameters for the distribution.

This last example motivates the conjecture that a fan is covering if and only if it is recursively regular. Note that a fan that is not recursively-regular must have a completely non-regular convex region, and this fact could perhaps be used to construct a point set for which no covering assignment exists.

5.2 Testing assignments in the plane

In this section, we address the problem of, given a polyhedral fan in the plane, a set of points and an assignment, decide whether the assignment is covering. As seen in the previous section, the overlapping conditions are necessary for an assignment to be covering. In the plane, the overlapping conditions already imply that the uncovered region is bounded. We develop below some other properties of the uncovered region that lead to an efficient algorithm to compute it.



(a) Illustration for Lemma 5.16 (b) Illustration for Lemma 5.18

Figure 5.4: Drawings for the proofs in Section 5.2.

Lemma 5.16. *Let $\mathcal{F}\subset\mathbb{R}^2$ be a full-dimensional complete fan with n cells, and let $P\subset\mathbb{R}^2$ be a set of n points. If an assignment $\sigma : cells(\mathcal{F})\to P$ satisfies the overlapping condition for every facet, then the set $\sigma(C)+C^-$ is covered, for all $C\in cells(\mathcal{F})$.*

Proof. Assume for the sake of contradiction that there is a cell $C_0\in cells(\mathcal{F})$ and a point $p_0\in\sigma(C_0)+C_0^-$ that is not covered. See Figure 5.4(a) for a drawing. Since the uncovered region is an open set, we can further assume that $\sigma(C_0)-p_0$ and $p_0-\sigma(C_0)$ are not directions of any ray of $\mathcal{F}$. Consider the ray ρ with apex at p_0 and direction $\sigma(C_0)-p_0$. Without loss of generality, we take the origin to be p_0 and ρ to be vertical and pointing "downwards". Consider now the set of cells $\mathcal{C}=\{C_1,\dots,C_m\}\subset cells(\mathcal{F})$ completely contained in the right halfplane. Assume that the indices of the cells in $\mathcal{C}$ correspond to their counterclockwise order from ρ, and let $\lambda_0,\dots,\lambda_m$ be the (closed) rays bounding the cells of $\mathcal{C}$, counterclockwise ordered as well.

We will now prove by induction in k that the line supporting $\sigma(C_k)+\lambda_k$ intersects ρ, for all $k\in[m]$. For $k=0$, the ray $\sigma(C_0)+\lambda_0$ intersects ρ by construction. Assume that the statement is true for k, that is, the line ℓ supporting $\sigma(C_k)+\lambda_k$ intersects ρ. Then, the point $\sigma(C_{k+1})$ lies below ℓ, since the corresponding overlapping condition for λ_k is satisfied. In addition, since the line supporting $\sigma(C_{k+1}) + \lambda_{k+1}$ is non-vertical, it either intersects ρ or it intersects the interior of $-\rho$. In the second case, the set $\sigma(C_{k+1}) + C_{k+1}$ covers p_0, leading to a contradiction. Hence, the inductive step is proven.

We can perform the symmetric argument for the set $\mathcal{C}'$ of cells completely contained in the left halfplane. Then, the overlapping conditions for cell $\bar{C}$ containing the direction of $-\rho$ imply that $p_0 \in \sigma(\bar{C}) + \bar{C}$, which contradicts the assumption that p_0 is uncovered. $\square$

Definition 5.17. Let $\mathcal{F} \subset \mathbb{R}^2$ be a full-dimensional complete fan with n cells, and let $P \subset \mathbb{R}^2$ be a set of n points. For a cell $C \in \mathit{cells}(\mathcal{F})$, let $\lambda(C)$ denote its clockwise-minimum ray and $\rho(C)$ denote its clockwise-maximum ray. We denote by $\lambda(C)^-$ the open halfspace bounded by the line supporting $\lambda(C)$ and disjoint from C, and by $\lambda(C)^+$ the set $\mathbb{R}^2 \setminus \lambda(C)^-$. We denote by $\rho(C)^-$ the open halfspace bounded by the line supporting $\rho(C)$ and disjoint from C, and by $\rho(C)^+$ the set $\mathbb{R}^2 \setminus \rho(C)^-$.

Lemma 5.18. *Let $\mathcal{F} \subset \mathbb{R}^2$ be a full-dimensional complete fan with n cells, and let $P \subset \mathbb{R}^2$ be a set of n points. If an assignment $\sigma : \mathit{cells}(\mathcal{F}) \to P$ satisfies the overlapping condition for every facet, then the region that is not covered by σ is either*

$$\bigcap_{C \in \mathit{cells}(\mathcal{F})} \left(\sigma(C) + \rho(C)^-\right) \qquad \text{or} \qquad \bigcap_{C \in \mathit{cells}(\mathcal{F})} \left(\sigma(C) + \lambda(C)^-\right).$$

Proof. Let B and D be two cells of $\mathcal{F}$ sharing the ray $\beta = \lambda(B) = \rho(D)$. The overlapping condition for β is satisfied if and only if $\sigma(D) + \rho(D)^- \subset \sigma(B) + \lambda(B)^+$. See the drawing in Figure 5.4(b). Lemma 5.16 states that the uncovered region is contained in

$$\bigcap_{C \in \mathit{cells}(\mathcal{F})} \left(\sigma(C) + \left((\rho(C)^- \setminus \lambda(C)^-) \cup (\lambda(C)^- \setminus \rho(C)^-)\right)\right).$$

Assume that there is an uncovered point $x \in \sigma(B) + (\lambda(B)^- \setminus \rho(B)^-)$ and note that

$$\left(\sigma(D) + \left(\rho(D)^- \setminus \lambda(D)^-\right)\right) \cap \left(\sigma(B) + \left(\lambda(B)^- \setminus \rho(B)^-\right)\right) = \varnothing,$$

since $\sigma(D) + \rho(D)^- \subset \sigma(B) + \lambda(B)^+$, which has empty intersection with $\sigma(B) + \lambda(B)^-$. Thus, we have that $x \in \sigma(D) + (\lambda(D)^- \setminus \rho(D)^-)$ and, iterating the argument, it follows that

$$x \in \bigcap_{C \in \mathit{cells}(\mathcal{F})} \left(\sigma(C) + \lambda(C)^-\right).$$

Symmetrically, if there is an uncovered point $x \in \sigma(B) + (\rho(B)^- \setminus \lambda(B)^-)$, then x belongs to $\sigma(C) + \rho(C)^-$ for all $C \in \mathit{cells}(\mathcal{F})$.

We prove now that indeed at least one of the two intersections must be empty. Let us assume that $H = \bigcap_{C \in \mathit{cells}(\mathcal{F})} (\sigma(C) + \rho(C)^-)$ is non-empty. Since H is bounded (because σ satisfies all the overlapping conditions), we have that $\widetilde{H} = \bigcap_{C \in \mathit{cells}(\mathcal{F})} (\sigma(C) + \rho(C)^+) = \varnothing$. In addition, we have that

$$\bigcap_{C \in \mathit{cells}(\mathcal{F})} \left(\sigma(C) + \lambda(C)^-\right) \subset \widetilde{H}.$$

Indeed, $\sigma(B) + \lambda(B)^- \subset \sigma(D) + \rho(D)^+$ for all $B \in \mathit{cells}(\mathcal{F})$, where D is the cell counterclockwise next to B, since the overlapping conditions give $\sigma(D) + \rho(D)^- \subset \sigma(B) + \lambda(B)^+$. The symmetrical argument finishes the proof of the claim. $\square$

As a consequence of Lemma 5.18, the region that is not covered by an assignment is either unbounded (if some overlapping condition is violated) or a convex polygon. An algorithmically interesting implication is stated below.

Theorem 5.19. *Let $\mathcal{F} \subset \mathbb{R}^2$ be a full-dimensional complete fan with n cells, and let $P \subset \mathbb{R}^2$ be a set of n points. If the adjacency information is in the input, it can be checked in $O(n)$ time whether an assignment $\sigma : cells(\mathcal{F}) \to P$ is covering.*

Proof. We know that the overlapping conditions are necessary for an assignment to be covering, as stated in Section 5.1. These conditions can be checked in $O(n)$ time. If the assignment satisfies them, then we check whether $H_1 = \bigcap_{C \in cells(\mathcal{C})}((\sigma(C) + \rho(C)^-))$ is empty using linear programming. Indeed, since the slopes of the lines bounding the halfplanes we want to intersect are sorted (because the fan is given as input), we can decide if H_1 is empty just by computing the lower and the upper envelope of the lover and upper halfplanes and testing whether they intersect by a sweeping-line algorithm. We analogously check if $H_2 = \bigcap_{C \in cells(\mathcal{C})}((\sigma(C)+\lambda(C)^-))$ is empty. Lemma 5.18 assures that the assignment is covering if and only if both H_1 and H_2 are empty. □

5.3 Illuminating a stage

The problem of illuminating a pointed cone using floodlights is closely related to the problem of illuminating a stage. Informally, the problem in the plane asks whether given n angles and n points, floodlights having the required angles can be placed on the points in a way that a given segment (the stage) is completely illuminated.

This problem was introduced in [26], where a partial answer for the problem in the plane is given. In that work, a floodlight can be freely rotated before its placement is determined. It is claimed [26] that if the sum of the angles is at least the angle of the smallest cone defined by two lines through the endpoints of the stage and containing all the points, then the stage can be illuminated. This result triggered a variant of the problem called *optimal illumination of the stage*. In this problem, one tries to illuminate a stage placing floodlights at prescribed points such that the sum of the angles used is minimized. This problem was solved in [43], where an $O(n \log n)$ algorithm is provided. However, it is also shown that, in the optimal solution, at least one of the points receives more than one floodlight and the points which do not belong to the convex hull of the point set do not receive any. If we require each point to hold exactly one floodlight, the problem can be solved in $O(n \log n)$ time as well, as shown in [46]. In contrast, if the floodlights are already assigned to points and they are only allowed to rotate, the decision problem of whether the stage can be illuminated is NP-complete [73].

We consider here a version of the stage illumination problem in higher dimensions. We require the assignment to be one-to-one and we do not allow the floodlights to rotate.

Definition 5.20. Let $\mathcal{F} \subset \mathbb{R}^d$ be a full-dimensional polyhedral fan consisting of n cells, $P \subset \mathbb{R}^d$ be a set of n points and Π be a hyperplane. The $(d-1)$-dimensional polyhedron $Q = |\mathcal{F}| \cap \Pi$ is called the *stage*. An assignment $\sigma : cells(\mathcal{F}) \to P$ *covers Q with $\mathcal{F}$ from P* if it is one-to-one and

$$\bigcup_{C \in cells(\mathcal{F})} (C + \sigma(C)) \supset Q.$$

The following proposition summarizes the relation bounding the stage illumination and the pointed cone illumination problems.

Proposition 5.21. *Let $\mathcal{F} \subset \mathbb{R}^d$ be full-dimensional a polyhedral fan with n cells.*

(i) If $\mathcal{F}$ is recursively-regular, then, for any hyperplane Π and any point set $P \subset |\mathcal{F}|^-$ of n points, there exists an assignment that covers $Q = |\mathcal{F}| \cap \Pi$ with $\mathcal{F}$ from P.

(ii) If $\mathcal{F}$ is cyclic in some direction, then, for every hyperplane Π with normal vector interior to $|\mathcal{F}|$, there exists a set of n points $P \subset |\mathcal{F}|^-$ such that there is no assignment covering $Q = |\mathcal{F}| \cap \Pi$ with $\mathcal{F}$ from P.

Proof.

(i) As a direct consequence of Theorem 5.4, there is a covering assignment for $|\mathcal{F}|$ and P. In particular, this assignment covers Q with $\mathcal{F}$ from P.

(ii) Assume, without loss of generality, that the apex of $\mathcal{F}$ is the origin of $\mathbb{R}^d$. Let v be a direction in which $\mathcal{F}$ is cyclic, and let ℓ be a line having direction v. Consider a set P of n points in ℓ. As observed in the proof of Theorem 5.14, any assignment must violate the overlapping condition for at least one interior facet of $\mathcal{F}$. However, if the points on the line are not carefully chosen, the stage Q could still be covered by some assignment. Even more, for every assignment σ there is an uncovered ray $r_\sigma = \{x_\sigma + \lambda v_\sigma : \lambda \in \mathbb{R}^+\}$ with $x_\sigma \in \mathbb{R}^d$ and $v_\sigma \in \mathbb{R}^d$ interior to $|\mathcal{F}|$. By Lemma 5.11-(i), the intersection $r_\sigma \cap |\mathcal{F}|$ is an unbounded ray $\tilde{r}_\sigma = \{y_\sigma + \lambda v_\sigma : \lambda \in \mathbb{R}^+\}$ with $y_\sigma \in \mathbb{R}^d$. Given $\alpha \in \mathbb{R}^+$, let $h_\alpha : \mathbb{R}^d \to \mathbb{R}^d$ be the uniform scaling of the coordinates of $\mathbb{R}^d$ (with center at the origin) by a factor of α, and let α_σ be the smallest α such that $h_\alpha(y_\sigma)$ lies above Π (that is, in the halfspace bounded by Π and containing the origin). Note that h_α fixes both $|\mathcal{F}|$ and $|\mathcal{F}|^-$, and that if $h_\alpha(y_\sigma)$ is above Π then $h_\beta(y_\sigma)$ is too, for all positive $\beta < \alpha$. Let $\bar{\alpha}$ be the minimum α_σ among all the assignments $\sigma : \mathit{cells}(\mathcal{F}) \to P$, and let $\bar{P} = h_{\bar{\alpha}}(P)$. The assignment $\bar{\sigma} : \mathit{cells}(\mathcal{F}) \to \bar{P}$ induced by $\sigma : \mathit{cells}(\mathcal{F}) \to P$ fails to cover $Q \cap h_{\bar{\alpha}}(\tilde{r}_\sigma)$, which finishes the proof. □

The converse of Proposition 5.21-(i) is, in general, not true. That is, an assignment can cover a stage Q, yet fail to cover the whole cone induced by Q. This fact makes the proof of the following statement more involved.

Proposition 5.22. *There exists an acyclic non-recursively-regular fan $\mathcal{F} \subset \mathbb{R}^3$, a plane Π and a set of n points $P \subset |\mathcal{F}|^-$ such that there is no assignment covering $Q = |\mathcal{F} \cap \Pi|$ with $\mathcal{F}$ from P.*

Proof. Consider the example exhibited in the proof of Proposition 5.15. The fan $\mathcal{F}$ is the one sketched in the right part of Figure 5.3. Assume, without loss of generality, that the apex of $\mathcal{F}$ apex is the origin and it is contained in the lower halfspace of $\mathbb{R}^3$. From the arguments in the proof of Proposition 5.15, every assignment for $\mathcal{F}$ and P violates the overlapping condition for at least one facet. Thus, every assignment σ fails to cover a ray $r_\sigma = \{x_\sigma + \lambda v_\sigma : \lambda \in \mathbb{R}^+\}$ with $x_\sigma \in \mathbb{R}^3$ and $v_\sigma \in \mathbb{R}^3$ interior to $|\mathcal{F}|$ and, by Lemma 5.11-(i), $r_\sigma \cap |\mathcal{F}|$ is an unbounded ray $\tilde{r}_\sigma = \{y_\sigma + \lambda v_\sigma : \lambda \in \mathbb{R}^+\}$ with $y_\sigma \in \mathbb{R}^3$. Let Π be an horizontal plane intersecting $\tilde{r}_\sigma$ for all the assignments σ, which can be found because all the v_σ have negative vertical coordinate. Then, $Q = |\mathcal{F} \cap \Pi| \neq \varnothing$ cannot be covered with $\mathcal{F}$ from P. □

6

Other applications and related problems

In this chapter we describe applications of the theoretical results introduced before.

6.1 Redundancy in spider webs

In this section, we present a problem related to the finest regular coarsening of subdivisions in $\mathbb{R}^2$. The results in this section belong to the field of tensegrity theory. The necessary background for this topic is summarized in Section 3.4.

Let G be an abstract spider web on the vertex set V, and let $p : V \to \mathbb{R}^2$ be an embedding corresponding to a non-crossing straight-line realization of G. Assume that the vertices lying on the convex hull of $p(V)$ are fixed (they are, therefore, in equilibrium by definition). Note that the straight-line realization of $G(p)$ can be thought of as a polyhedral subdivision of the convex hull of $p(V)$ in the plane. Throughout this section, this subdivision will be denoted by $\mathcal{S} = \mathcal{S}(G(p))$ and called the subdivision *associated to* the spider web.

The Maxwell-Cremona correspondence states that $G(p)$ has a strictly-proper stress if and only if $\mathcal{S}$ is regular. From this fact and Lemma 3.14, it is easy to derive the following proposition.

Proposition 6.1. *Let $\mathcal{S}$ be the subdivision associated to a planar spider web $G(p)$.*

(i) Only the cables of G corresponding to edges of the finest regular coarsening of $\mathcal{S}$ support a positive stress in any equilibrium stress of $G(p)$.

(ii) If $\mathcal{S}$ is recursively regular, then $G(p)$ is rigid.

Proof.

(i) Since we showed that the edges omitted in the finest regular coarsening are lifted into a plane by any convex lifting (Theorem 4.11), the Maxwell-Cremona correspondence indicates that the corresponding cables will receive no stress in any proper equilibrium.

(ii) The finest regular coarsening of the subdivision corresponds to a set of cables such that there is an equilibrium stress assigning positive values to all of them. Therefore, the spider web defined by this set of cables is rigid by Lemma 3.13. For each of the subsubdivisions defined by the finest regular coarsening, we can assume that the vertices in the corresponding convex hull are now fixed and apply the previous argument recursively. □

Figure 6.1 illustrates the previous result. The spider web represented in it is constructed from a triangulation appearing in [4]. The edges omitted in the picture to the right, which do not belong to the finest regular coarsening of the associated subdivision, support no stress in any equilibrium. Therefore, they can be considered redundant.

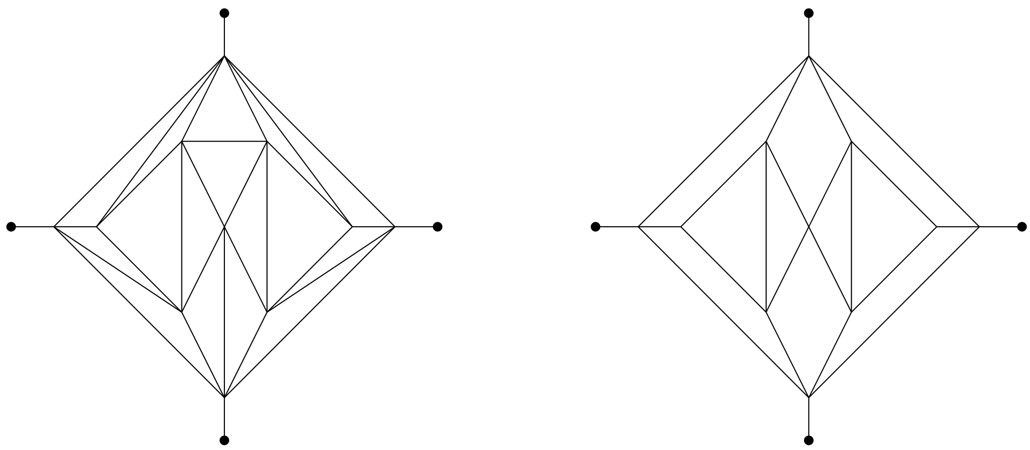


Figure 6.1: A spider web (left) and the result of removing redundant cables (right).

Note that even thought recursively-regular subdivisions are associated to rigid spider webs, these might be far from infinitesimally rigid. For instance, if a regular subdivision is refined by adding an edge whose endpoints are interior to previous edges, the result is recursively regular but obviously not infinitesimally rigid. We next translate a well-known fact of infinitesimal rigidity to the language of finest regular coarsenings.

Corollary 6.2. *The subdivision associated to a infinitesimally rigid spider web is its own finest regular coarsening (hence, it is regular).*

Proof. As Lemma 3.14 states, if a framework is infinitesimally rigid, it has a strictly-proper stress. The edges omitted in the finest regular coarsening of the associated subdivision cannot participate in such stress. Therefore, none of the edges are omitted in the finest regular coarsening of the subdivision. □

6.2 Voronoi Tree-maps

Mainly because of the big amount of data generated and used by people and companies nowadays, Data Visualization has become a wide field in Computer Science. In this area of research, *tree-maps* were studied for first time, to best of our knowledge, by Johnson and Shneiderman [76]. These objects are planar tessellations representing a hierarchical structure and whose cells have prescribed areas. Hence, the result stated in Theorem 2.15-(ii) offers a nice starting point for the creation of a tree-map. This connection has been exploited to construct so-called *Voronoi tree-maps* [16, 84]. These tree-maps are created by partitioning an initial convex polygon by overlaying on it a power diagram, after adjusting the weights such that each region has the area prescribed for it. Figure 6.2 shows a tree-map made by computing the Voronoi diagram of the blue points, refining the resulting cells by the diagrams of the red, purple and yellow points, and finally splitting one of the cells corresponding to a purple point by the Voronoi diagram of the green points. Then, the lower levels of the hierarchy are constructed in a similar way.

The Voronoi tree-map approach does not take into account any relation between different levels of the hierarchy. Every node in the hierarchy is partitioned using an independent (and, to some extent, arbitrary) power diagram. We will next analyze this kind of constructions in terms of recursive regularity. This will allow us to transform an initial subdivision to meet the prescribed volume requirements, while preserving some features of the tessellation.

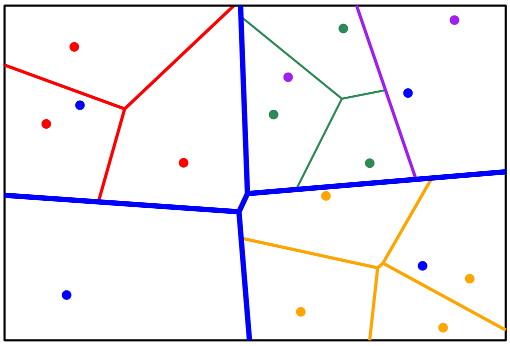


Figure 6.2: A Voronoi tree-map.

For completeness, we reproduce below a simple proof of a well-known fact [85, 50, 79, 12], which reveals the relation between power diagrams and regular polyhedral complexes.

Lemma 6.3 (folklore)**.** *The set of regular polyhedral complexes $\mathcal{S}$ with $|\mathcal{S}| = \mathbb{R}^d$ is the set of power diagrams in $\mathbb{R}^d$.*

Proof. Let $q_1, \dots, q_k \in \mathbb{R}^d$ be sites with associated weights $w_1, \dots, w_k \in \mathbb{R}^+$. Note that, for all $j \in [k]$, we have that

$$\|x - q_i\|^2 - w_i^2 \leq \|x - q_j\|^2 - w_j^2 \quad \text{if and only if} \quad 2\langle x, q_i\rangle - \|q_i\| + w_i^2 \leq 2\langle x, q_j\rangle - \|q_j\| + w_j^2.$$

Thus, the power diagram of the given weighted sites is the vertical projection of the lower envelope of a set of hyperplanes and, hence, it is a regular polyhedral complex.

To see the converse, one only needs to take the polyhedron which projects onto the given regular subdivision, and regard the hyperplanes supporting its (say k) facets as functions

$$f_i(x) = q_i \cdot x + b_i = -(\|x - q_i\|^2 - \|x\|^2 - \|q_i\|^2)/2 + b_i, \text{ for } i \in [k].$$

Note that, without loss of generality, we can assume $b_i > 0$ for all $i \in [k]$ by adding a constant to all the functions without changing the subdivision they induce. The maximization diagram of the functions $-f_i(x)$ for $i \in [k]$ is the same as the minimization diagram of

$$\tilde{f}_i(x) = \|x - q_i\|^2 - 2b_i - \|q_i\|^2, \text{ for } i \in [k],$$

which can be thought of as the power of x with respect the circle centered at q_i with radius $\sqrt{2b_i + \|q_i\|^2}$, for all $i \in [k]$. □

As a consequence of the previous lemma, we can assign to any regular subdivision $\mathcal{S}$ a (possibly non-uniquely determined) set of weighted sites, denoted by $sites(\mathcal{S})$, whose power diagram is $\mathcal{S}$. Then, we can apply Theorem 2.15-(ii) to re-weight the sites in a way that the prescribed area requirements are fulfilled. The adjacency relations between cells are in general not preserved. However, the obtained tessellation resembles the original one in some specific sense. For instance, the facets appearing in the tessellation have directions corresponding to bisectors of $sites(\mathcal{S})$. We formalize some of other properties next.

Proposition 6.4. *Given a full-dimensional recursively regular subdivision $\mathcal{S}$ in $\mathbb{R}^d$, a polyhedron $P \subset \mathbb{R}^d$ and a function $v : cells(\mathcal{S}) \to \mathbb{R}^+$ such that $\sum_{C \in cells(\mathcal{S})} v(C) = \mathrm{Vol}(P)$, there is a tessellation $\mathcal{Q}$ of P satisfying the following properties.*

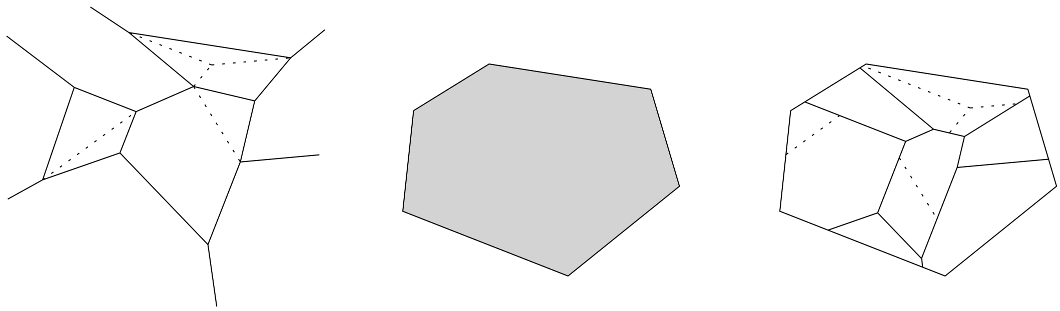


Figure 6.3: A representation of a regular subdivision tree (left), a polygon (center) and a partition of this polygon by the subdivision tree.

(1) The cells of $\mathcal{Q}$ are polyhedra, and they are in correspondence with the cells of $\mathcal{S}$.

(2) The region corresponding to a cell C has volume $v(C)$.

(3) If a pair of cells share a wall in $\mathcal{S}$ and in $\mathcal{Q}$, the (affine hulls of) the walls are parallel.

Proof. We can use the recursively-regular structure of $\mathcal{S}$ in combination with Lemma 6.3 to recursively apply the partition result of the Theorem 2.15-(ii) using a hierarchy of power diagrams encoded in the regularity tree of $\mathcal{S}$. To meet the volume requirements, we first partition P using a power diagram of the finest regular coarsening of $\mathcal{S}$ in a way that each region has volume equal to the sum of the volumes that have been prescribed for the regions it contains. We repeat this procedure recursively. Statements (1) and (2) follow by construction. Let C and D be cells sharing the wall W in $\mathcal{S}$ and W' in $\mathcal{Q}$. If C and D are both in the same (regular) leaf $\mathcal{C}$ of the regularity tree, the associated cells in $\mathcal{Q}$ are created by shifting the hyperplanes of the polyhedron projecting on the (regular) subdivision of $A \cap |\mathcal{C}|$ induced by $\mathcal{S}$. Hence, the wall W' is the projection of the intersection of the shifted hyperplanes, which is parallel to the intersection of the original ones. If C and D are not in the same leaf, let $\mathcal{C}'$ be the lowest common ancestor of the leaves containing C and D. The ground set of this node is split by a regular subdivision $\mathcal{S}'$ and leaving C and D in different children. Then, W must be a wall of $\mathcal{S}'$. The final cells of $\mathcal{Q}$ associated to C and D will be contained in the polytopes resulting from the partition via $\mathcal{S}'$ of the cell corresponding to $\mathcal{C}'$. These polytopes share a wall parallel to W (and only this one). Thus, any point common to the cells in $\mathcal{Q}$ corresponding to C and D must be contained in that wall. This fact completes the proof of (3). □

Note that the coarsening structure used in the previous proposition can be altered, as far as the used coarsenings are regular. That is, if we have a regular subdivision tree having as leaves the cells of the subdivision, we can use the associated hierarchy of coarsenings to partition the given polytope. We would also like to remark that the discrete counterpart of the previous result (analogous to Theorem 2.15-(i)) applies as well. Note also that for $d = 2$, if the graph induced by $\mathcal{S}$ has n vertices and m edges, then $\mathcal{Q}$ has at most $3(m - n)$ edges and, thus, the complexity of $\mathcal{Q}$ is not significantly larger than the complexity of $\mathcal{S}$.

Figure 6.3 shows an example a tessellation as the ones described in Proposition 6.4, where the dotted edges correspond to the second level of the hierarchy.

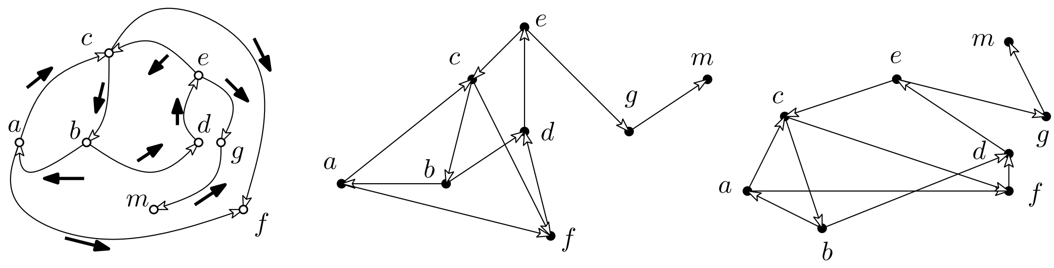


Figure 6.4: A directional graph (left), a drawing (center), and an embedding (right).

6.3 Embeddings of directional graphs

As shown in Chapter 5, for the existence of a covering assignment it is necessary that there is an assignment satisfying the overlapping condition for every interior facet of the fan. Moreover, the examples we have found so far of polyhedral fans and point sets for which there is no covering assignment fail to fulfill the second condition. Hence, it could be that this condition is also sufficient. In any case, we think that it is of independent interest to study this condition alone, which is connected to a problem on graph embedding.

Note first that the overlapping condition for a facet can be expressed as a requirement on the order in which the two involved points are swept by a hyperplane parallel to the facet. That is, we want to know which of two points "appears first" in a specific direction. The problem we study here asks whether, given set of relations of this type (stated on labels) and a point set, we can find a one-to-one labeling of the point set such that every relation is satisfied. We next describe the problem formally.

Definition 6.5. A *directional graph* is a tuple $\overrightarrow{G} = (V, h)$, where V is a set and $h : V \times V \to \mathbb{R}^d$ is a function such that $h(v, u) = -h(u, v)$, for all $v, u \in V$. The elements of V are called *vertices.* We say that $u, v \in V$ are connected by an *edge* if $h(v, u) \neq 0$. The *dimension* of $\overrightarrow{G}$ is d.

We may regard this structure as a directed graph with a non-zero direction associated to every edge. Such a graph will be called the *underlying graph* of the directional graph. Note that the condition in the definition already implies that $h(v, v) = 0$, for all $v \in V$.

Definition 6.6. An *embedding* of a d-dimensional directional graph $\overrightarrow{G} = (V, h)$ on a point set $P \subset \mathbb{R}^d$ is a one-to-one assignment $\sigma : V \to P$ such that

$$\langle h(v, u), \sigma(v) - \sigma(u) \rangle \geq 0, \text{ for all } v, u \in V.$$

If such an embedding exists, we say that $\overrightarrow{G}$ is *embeddable* in P. A *drawing* of a directional graph $\overrightarrow{G} = (V, h)$ is a bijection $\pi : V \to S \subset \mathbb{R}^d$ such that for all $u, v \in V$ with $h(u, v) \neq 0$ we have that $\pi(v) - \pi(u) = \lambda_{uv} \cdot h(v, u)$ for some $\lambda_{uv} > 0$. The *projection* of a d-dimensional directional graph $\overrightarrow{G}$ into a k-dimensional linear subspace $L \subset \mathbb{R}^d$ is the k-dimensional directional graph obtained by projecting the vector $h(u, v) \in \mathbb{R}^d$ onto $L \cong \mathbb{R}^k$, for all $v, u \in V$.

A directional graph is illustrated in Figure 6.4, together with a drawing and an embedding. The arrows near the edges indicate the directions associated with them. Observe that the embedding condition for an edge restricts its direction to a halfspace, while the drawing condition fixes its direction completely. Note also that the lengths of the vectors assigned by h are irrelevant for the existence of an embedding or a drawing of a directional graph. Therefore, we will consider two directional graphs (V, h) and (V, h') *equivalent* if $h(u, v)$ is a positive scalar multiple of $h'(u, v)$ for all $u, v \in V$.

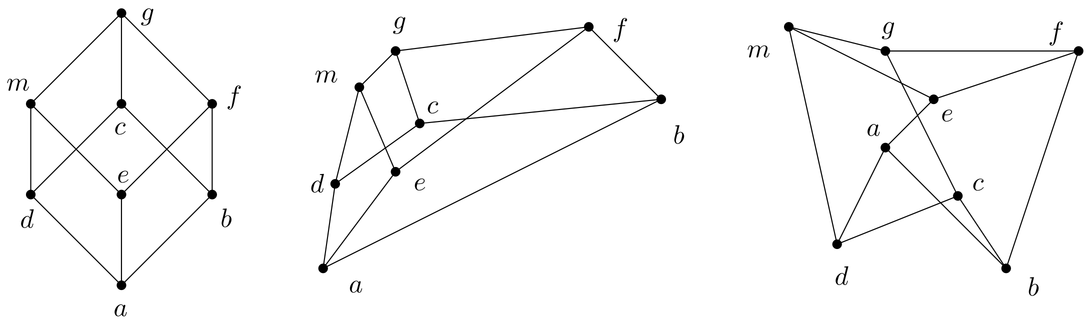


Figure 6.5: Projected prism and valid embeddings into two different point sets.

Definition 6.7. A d-dimensional directional graph $\overrightarrow{G} = (V, h)$ is *universally embeddable* if it is embeddable on any point set $P \subset \mathbb{R}^d$ with $|P| = |V|$. It is *drawable* if it has a drawing.

Polytopes and polyhedral fans are related to certain directional graphs in a natural way.

Definition 6.8. The *directional graph of a polytope* is the set of its vertices, together with the function $h(u, v) = v - u$ if u and v are endpoints of an edge of the polytope, and $h(u, v) = 0$ otherwise. The *normal graph* of a polyhedral fan is set of its cells with the function $h(C, D)$ being a vector normal to the facet common to C and D and pointing "from C to D " if they share a facet, and $h(u, v) = 0$ otherwise.

Figure 6.5 shows a projection of the graph of a 3-prism into the plane and embeddings of it into two different point sets. Note that the directional graph of a polytope and the graph of its normal fan are embedding-equivalent. This is a consequence of the duality between a polytope at its normal fan.

The following proposition shows that there is a surprisingly large family of universally embeddable directional graphs.

Proposition 6.9. *If a directional graph is drawable, then it is universally embeddable. In particular, a directional graph $\overrightarrow{G} = (V, h)$ with underlying graph being a tree is universally embeddable regardless of h. The directional graph of a polytope is universally embeddable.*

Proof. Given a drawable directional graph $\overrightarrow{G} = (V, h)$ and an arbitrary point set P with $|P| = |V|$, consider a drawing π of $\overrightarrow{G}$. Let μ be the least-squares optimal matching between $\pi(V)$ and P. We will show that $\mu \circ \pi$ is an embedding of $\overrightarrow{G}$. Assume that it is not the case. Then, there must be a pair $u, v \in V$ such that $\langle h(v, u), \mu(\pi(v)) - \mu(\pi(u))\rangle < 0$. Since $\pi(u) - \pi(v) = \lambda_{uv} \cdot h(v, u)$, for some $\lambda_{uv} \in \mathbb{R}^+$, we have that $\langle \pi(u) - \pi(v), \mu(\pi(v)) - \mu(\pi(u))\rangle < 0$, which contradicts the optimality of μ because swapping the images of $\pi(u)$ and $\pi(v)$ would improve the matching. Directional graphs having a tree as underlying graph are trivially drawable and directional graphs of polytopes have the 1-skeleton of the polytope as a drawing. □

It is not hard to see that if there is a sequence of vertices $v_1, \dots, v_l, v_{l+1} = v_1$ in V and a vector $\delta \in \mathbb{R}^d$ such that $\langle h(v_i, v_{i+1}), \delta\rangle > 0$, for all $i \in [l]$, then the graph is not drawable. Such a cycle is called a *(δ-)forcing cycle.* However, the converse is not true in general: for instance, the normal graph of the subdivision in Figure 5.3 has no forcing cycle but it is also non-drawable.

The following proposition summarizes some relations of recursive regularity to drawability and embeddability of directional graphs.

Proposition 6.10.

(i) The projection of a universally embeddable directional graph is universally embeddable.

(ii) Normal graphs of recursively-regular fans are universally embeddable.

(iii) Universally embeddable graphs are not necessarily drawable.

(iv) Graphs with forcing cycles are not universally embeddable.

(v) There are graphs with no forcing cycles that are not universally-embeddable.

Proof.

(i) Let $\overrightarrow{G} = (V, h)$ be a d-dimensional universally-embeddable directional graph, and let L be a k-dimensional linear subspace of $\mathbb{R}^d$ with a basis $\{l_1, \ldots, l_k\}$. Let $\bar{G} = (V, \bar{h})$ be the projection of $\overrightarrow{G}$ onto L, which is identified with $\mathbb{R}^k$ through the bijection

$$\begin{aligned} i : \mathbb{R}^k &\to L \subset \mathbb{R}^d \\ (x_1, \ldots, x_k) &\longmapsto \sum_{j \in [k]} x_j l_j. \end{aligned}$$

Consider any set of $|V|$ points $\bar{P} \subset \mathbb{R}^k$, and the associated point set $P = i(\bar{P}) \subset \mathbb{R}^d$. If $\sigma : V \to P$ is an embedding of $\overrightarrow{G}$ on P, then $\bar{\sigma} = i^{-1} \circ \sigma$ is an embedding of $\bar{G}$ in $\bar{P}$, where i^{-1} denotes the inverse of i on L. Indeed, $\langle h(v,u), \sigma(v) - \sigma(u)\rangle = \langle \bar{h}(v,u), \bar{\sigma}(v) - \bar{\sigma}(u)\rangle$ for all $u, v \in V$, because $\sigma(u) - \sigma(v) \in L$ and thus only the projection of $h(u,v)$ onto L contributes to the scalar product.

(ii) Let $\mathcal{F} \subset \mathbb{R}^d$ be a full-dimensional polyhedral fan consisting of n cells. Theorem 5.4 ensures that there is a covering assignment for $\mathcal{F}$ and any set P of n points. This assignment must satisfy the overlapping condition for each facet of the fan, which is equivalent to the embedding condition for the corresponding edge.

(iii) The normal graph of a fan is drawable if and only if the fan is regular (see, for instance, [11]). Thus, the normal graph of a recursively-regular non-regular fan is not drawable and it is, however, universally embeddable, as shown in (ii).

(iv) Consider a δ-forcing cycle $v_1, \ldots, v_l, v_{l+1} = v_1$. Take a set of different points in a line having direction vector δ and label them increasingly with respect to their scalar products with δ. For any embedding σ, $\sigma(v_{i+1})$ must have a label larger than $\sigma(v_i)$, for all $i \in [l]$, which is obviously impossible.

(v) The normal graph of the fan obtained by taking cones from the subdivision in Figure 4.3 has no forcing cycle, since it is acyclic (in the visibility sense). However, we have given a set of points for which all the assignments violate an overlapping condition. Hence, there is no embedding of its normal graph into this point set. □

7

Concluding remarks and open problems

We have shown that the finest regular coarsening of a subdivision, which can be seen as the regular subdivision that is closest to it, can be used to define a structure called the regularity tree. The leaves of this tree define a partition of the subdivision in sub-subdivisions that are either regular or completely non-regular. The regularity tree reflects thus some of the structure of non-regular subdivisions and measures, in some sense, the degree of regularity. As a consequence, the class of recursively-regular subdivisions arises in a natural way. We have shown that this class goes beyond regular subdivisions while excluding cyclic ones. Because of this, it maintains several good properties of regular subdivisions, although it does not avoid all the pathologies of non-regular ones. We have reported on some of such features and have also related the new concepts to previously known results.

In addition, we have studied a collection of related applications. These results belong to a wide range of different areas, some of them more theoretical than others. We expect to find even more applications of the developed theory, since any theorem or algorithm based on the regularity of a subdivision and admitting a recursive scheme can probably be extended to apply for the larger set of recursively-regular subdivisions.

We studied the problem of illuminating the space by floodlights. It was known that regular fans are universal and our aim was to study the problem for the other fans. We have proved that not only regular fans are universal and that not only cyclic ones are non-universal. It makes then sense to ask what is the complexity class of the general problem of deciding whether the space can be covered by a given fan from a given point set (in dimensions bigger than two). It remains open as well to precise the limits of universality, that is, to characterize the polyhedral fans that can cover the space from any point set. A reasonable candidate is recursive-regularity. Indeed, the fact that a non-recursively-regular subdivision has a convex sub-subdivision which is completely non-regular could be the first step towards a proof for this fact. For the problem in the plane, we prove that the region that an assignment leaves uncovered is either unbounded or a convex polygon. This property is then used to give a linear algorithm to decide if a given assignment for a given fan is covering. Our results on covering the space by floodlights have implications for a three-dimensional version of the *stage illumination problem*, on which we report. The problem of embedding directional graphs is in a similar situation. A natural and easy to state question is whether deciding if a directional graph can be embedded in a given point set is NP-hard.

Concerning algorithmic issues, we proved that the finest regular coarsening and the regularity tree of a subdivision can be computed in polynomial time. We have used these facts to prove that recursive regularity of a subdivision can be decided in polynomial time as well, which is relevant for the algorithmic version of the aforementioned problems.

As pointed out in Chapter 2, recursive regularity can be defined for tessellations as in Definition 2.1. Almost all the presented results carry over this more general class of objects, albeit some of the arguments might need special care. On the other hand, finding examples in the class of tessellations (as recursively-regular but non-regular objects, or non-recursively regular objects) is easier, which is another reason to have restricted the study polyhedral complexes.

II

Voronoi diagrams for partial matching

8

Introduction to Part II

The process of point set registration, known as point matching as well, basically seeks a transformation that aligns two point sets. This problem is among the most fundamental challenges in computer vision and pattern recognition. It is widely used in areas such as medical imaging [65], molecular biology [107], object recognition [18], or object tracking [108]. An important variant of the problem searches for an occurrence of a “small” pattern in a “big” point set. This will be the focus of this part, providing approaches that are sensitive to the size of such pattern.

We present first some initial considerations on point matching and review the most relevant literature. Afterwards, we will introduce some notation and facts concerning matchings in graphs. Section 8.3 and Section 8.4 study two variants of the assignment problem for bipartite graphs that will be translated into a geometric setting in the main chapters of this part.

8.1 Point matching

Point-matching approaches are closely related to distances for point sets. In this context, the word distance should not be taken in the mathematical sense because the considered functions usually fail to satisfy the triangle inequality. This is the reason why they are often called similarity measures instead. Formally, a *similarity measure* or a *distance for point sets* is simply a function mapping pairs of point sets to (usually) the non-negative real numbers. In fact, this kind of functions are often called *dissimilarity measures*, since the larger the value they take, the more different the point sets are supposed to be. Some of these distances are only defined when the two sets are equally sized, while others make sense for arbitrary sets as well.

An extensively used distance is the *directed Hausdorff distance*, which matches each point of a set to its nearest neighbor in the other set, taking afterwards the maximum of the (Euclidean) distances between matched points. Observe that only one pair of points is actually relevant for the distance value: the distance between the point sets would not change if one removes all of the other points. This effect is partially avoided if one imposes that the points are matched in an injective way. The resulting distance is called the *(Euclidean) bottleneck distance*.

For several distances a “dynamic” version has been defined, meaning that one of the sets is allowed to be transformed according to some restricted set of transformations. The value of the dynamic distance is then the minimum of the distances attained between the fixed set and each of the instances of the transformed set. This approach is specially useful to decide to what extent a pattern set occurs in a larger set, modulo some kind of equivalence relation (induced by the set of transformations).

Different types of point-matching algorithmic questions are found in the literature. The *exact decision problem* aims to decide whether the distance between two point sets is zero. The *ε-decision problem* asks whether the distance between two sets is smaller than a fixed ε, and the *optimization problem* is equivalent to computing the actual value of the distance. Depending on the field, some authors are interested only in the properties of the distance, while others require it to be efficiently computable. A big part of the literature deals with points in the plane, while a different sector prefers the distance to be meaningful and efficiently computable also in higher dimensions.

In this thesis, we study mainly three types of point-set distances under translations. All of them are evaluated on matchings; that is, injections from the smaller of the two sets into the larger one. The pairs induced by the matching are called its edges, and the length of an edge is the distance between the points defining it. Then, the cost of a matching is the maximum length of the edges it uses (*bottleneck distance*), the decreasingly-ordered vector of lengths of its edges (*lexicographic-bottleneck distance*) or the sum of the squares of the lengths (*least-squares distance*). The value of the distance is then the minimum cost attained by a matching if the small point set can be freely translated.

The bottleneck distance was introduced for equally-sized point sets in the plane by Alt, Mehlhorn, Wagener and Welzl [9]. They gave an $O(n^6 \log n)$ algorithm for the optimization problem under translations and an $O(n^8)$ one for the ε-decision problem under isometries, among many other variants. For translations, the bound was improved to $O(n^5 \log n)$ for the ε-decision and $O(n^5 \log^2 n)$ for the optimization problem by Efrat, Itai and Katz [52]. Rezende and Lee [91] studied later the exact decision problem under congruences in higher dimensions. The problem in the plane under translations was revisited more recently by Bishnu, Bhattacharya, Das and Nandy [24]. A family of different static distances defined not only on matchings was studied by Eiter and Mannila [53]. For a survey on these and other related distances, see the work of Alt and Guibas [8]. The lexicographic bottleneck distance was studied for instance by Burkard and Rendl [28] and revisited by Aneja and Sokkalingam [103] in terms of weighted bipartite graphs. See the work of Croce, Paschos and Tsoukias [42] for another approach, and the general discussion on lexicographic orders by Fishburn [57].

Least-squares matchings have been proven to be relevant for many applications such as pattern recognition, registration and clustering (see, for instance, [97, 67, 5]). In addition, they are related to power diagrams [13]. An early study of these matchings by Silberberg and Zikan appeared in [110]. An approximation algorithm for the problem under rigid motions in the plane was given by Agarwal and Phillips [86]. As shown by Rote [93], the least-squares matchings under translations induce a subdivision of the positions of the translated point set according to its optimal matching. Rote showed that this subdivision is a polyhedral complex and that a line can intersect only polynomially many cells.

Another interesting distance with applications in image retrieval is the Earth Mover's Distance, defined for distributions or weighted points [96]. A FPTAS for its computation under rigid motions in the plane was provided by Cabello, Giannopoulos, Knauer and Rote in [30]. A variant of the distance was studied in [39]. The minimum Euclidean bipartite matching in the plane was studied by Vaidya [106] and improved by Agarwal, Efrat and Sharir as an application in [2]. An interesting example of a negative result is that deciding whether a point set is congruent with a subset of another point set is probably not fixed parameter tractable taking the dimension as a parameter, as shown by Cabello, Giannopoulos and Knauer [29].

A common strategy to find a local minimum of a distance under translations is the *Iterative Closest Point* (or ICP, for short) algorithm, introduced by Besl and McKay [22] and analyzed by Efrat, Ezra and Sharir [55]. Some variants of the algorithm and their efficiency in practice are reviewed in [97, 64]. The algorithm takes an optimal matching for an initial position and translates the point set to a minimum of the cost function of this matching. Then, the optimal matching for this new position is computed. This process is iterated and, under reasonable assumptions, it stops at a local minimum of the distance function. See [19] for a recent alternative approach to find a local minimum of the least-squares distance function under translations.

8.2 Definitions and notation for matching in graphs

In the following sections, we study two problems on graph matching that will be used later in a more geometric setting. We will deal throughout this part with two finite point sets A and B in $\mathbb{R}^d$ and we will be interested in the (Euclidean) distances between points from one set and points from the other. Therefore, we will be thinking implicitly of a complete bipartite graph. The vertices of the graph will be the points of A and B, and the edges will be the pairs $(a, b) \in A \times B$, which will be denoted by ab for short. The edges of this graph will be weighted by the Euclidean distance between the corresponding points. This value will be called *weight* or *length* of the edge. Furthermore, the weight of the edges will be often considered as a function of a parameter $t \in \mathbb{R}^d$, which encodes the position of B. Because of this close relation between point matching and matchings in weighted graphs, we introduce here some basic definitions and facts about them.

Definition 8.1. Let $G = (A, B; E)$ be a bipartite graph with edge set $E \subset A \times B$. A *matching* in G is a set $\sigma \subset E$ such that every vertex in $A \cup B$ is incident to at most one edge of σ. A *maximum matching* is a matching of maximum cardinality. A vertex belonging to an edge of σ is called a *matched vertex*. If a vertex is not matched, it is called *exposed* or *free*. The set of matched vertices is called the *matched set* of σ.

In the following chapters, we will often assume that matchings are maximum. In such case, we will also identify a matching with the injection from B to A that it induces.

Definition 8.2. Given two sets of edges $\sigma, \tau \subset E$, their *symmetric difference* is the set

$$\sigma \oplus \tau = (\sigma \setminus \tau) \cup (\tau \setminus \sigma) = (\sigma \cup \tau) \setminus (\sigma \cap \tau).$$

The connected components of (the graph induced by) $\sigma \oplus \tau$ will be called its *components*.

Note that, if σ and τ are maximum matchings, the components of $\sigma \oplus \tau$ are paths or cycles, since every vertex has degree at most two in $\sigma \cup \tau$.

Definition 8.3. Let $\sigma \subset E$ be a matching. An *alternating path* for σ is a path in G (without repeated vertices) whose even edges are in σ and whose odd ones are in $E \setminus \sigma$. An *alternating cycle* for σ is a cycle in G (without repeated vertices) whose even edges are in σ and whose odd ones are in $E \setminus \sigma$. An *augmenting path* for σ is an alternating path starting and ending at exposed vertices.

Note that, if γ is an augmenting path for σ, the matching $\tau = \sigma \oplus \gamma$ has one more matched vertex than σ. Observe also that given two matchings σ, τ the components of $\sigma \oplus \tau$ are alternating for σ and for τ. If γ is a component of $\sigma \oplus \tau$, then $\sigma \oplus \gamma$ is a matching as well. We call the transition from σ to $\sigma \oplus \gamma$ the *flip* of σ along γ. We also say that if we flip γ in σ we get $\sigma \oplus \gamma$.

Definition 8.4. A *cost function* is a map $f : 2^E \times \mathbb{R}^s \to \mathbb{R}$, where $s \in \mathbb{N}$. The *cost of a set of edges* E' is the function $f(E', \cdot) : \mathbb{R}^s \to \mathbb{R}$ that maps a parameter $t \in \mathbb{R}^s$ to $f(E', t)$. The *cost of an alternating path or alternating cycle* γ with respect to a matching σ is the difference between the cost of $\gamma \cap \sigma$ and the cost of $\gamma \setminus \sigma$. Given two matchings σ, τ, the *cost of a component* γ of $\sigma \oplus \tau$ will be the absolute value of the difference between the cost of $\gamma \cap \sigma$ and the cost of $\gamma \cap \tau$.

The cost functions in this thesis will typically be the sum of the (squared) weights of the edges in the set or the maximum weight over the edges in the set.

8.3 Bottleneck assignments problems

Let A and B be two sets with $k = |B| \leq |A| = n$, and $G = (A, B; E)$ be a bipartite graph with edge set $E \subset A \times B$. Let $w : E \to \mathbb{R}^{\geq 0}$ be a function giving weights to the edges. The bottleneck cost of a matching π in $G = (A, B; E)$ with respect to w is the maximum w-value attained by its edges. The problem of finding a maximum matching of minimum bottleneck cost (referred henceforth to as a *bottleneck matching*) for the complete and balanced case ($k = n$ and $E = A \times B$) has been widely studied in the last decades under the name of the *bottleneck assignment problem*. The most prominent approaches for this problem are the *threshold methods* and the *augmenting path methods*. Details and related issues can be found in the book dedicated to assignment problems from Burkard, Dell'Amico and Martello [27]. The threshold algorithms conduct a binary search on the possible values for the longest edge of a bottleneck matching. At each stage, the edges that are longer than the threshold are ignored, and a maximum matching computation in the modified graph is performed. One of the best-known algorithms to find a maximum matching in a bipartite graph is due to Hopcroft and Karp [71]. It runs in $O(|E|\sqrt{\nu(G)})$ time, where $\nu(G)$ is the size of the maximum matchings in G. The algorithm from Alt, Blum, Mehlhorn and Paul [7] finds a maximum matching more efficiently if the graph is "dense". This fact was exploited by Punnen and Nair [89] to develop an alternative algorithm for the bottleneck assignment problem. The pure threshold method is preferable for dense graphs, the method from Gabow and Tarjan [59] is better for sparse graphs and the approach of Punnen and Nair covers the range in between.

A variant of the bottleneck assignment problem is the *lexicographic bottleneck assignment problem*, introduced in [28] and revisited in [103]. For this problem, the cost of a matching $\sigma : B \hookrightarrow A$ is the result of sorting decreasingly the values $w(b\sigma(b))$ for all $b \in B$. Then, a *lexicographic bottleneck matching* is an matching minimizing the cost, when the corresponding vectors are compared lexicographically. Note that a lexicographic bottleneck matching is necessarily a bottleneck matching.

In the unbalanced case, as long as the graph is complete, the maximum matchings have size k. In addition, we can take advantage of a property similar to the one in Hall's marriage theorem. To state it, we introduce the following definitions and notation.

Definition 8.5. Let $G = (A, B; E)$ be a bipartite graph with $k = |B| \leq |A|$, and $w : E \to \mathbb{R}^{\geq 0}$ be a function giving weights to the edges. Given $b \in B$, a set M_b is a *b-minimal set* if $|M_b| = k$ and no edge $ab \in E \setminus M_b$ has strictly smaller weight than any edge in M_b. The *candidate set of* b is the set E_b with $|E_b| \geq k$ and such that every edge $ab \in E \setminus E_b$ has strictly larger weight than every edge in E_b.

Lemma 8.6. *Let $G = (A, B; E)$ be a bipartite graph with $k = |B| \leq |A|$, and $w : E \to \mathbb{R}^{\geq 0}$ be a function giving weights to the edges.*

(i) If a set $M \subset E$ contains a b-minimal set for each $b \in B$, then M contains a lexicographic bottleneck matching.

(ii) Every lexicographic bottleneck matching is contained in the union of candidate sets $Z = \cup_{b \in B} E_b$.

Proof.

(i) Let $\mu \subset E$ be a lexicographic bottleneck matching. If $\mu \subset M$, nothing is left to be proven. Otherwise, let $ab \in E$ be an edge in $\mu \setminus M$, and let $M_b \subset M$ be a b-minimal set. Since μ matches only k points of A and $ab \notin M \supset M_b$, there must be an edge $a'b \in M_b \setminus \mu$. The matching $(\mu \setminus \{ab\}) \cup \{a'b\}$ is lexicographically at least as good as μ and it uses one more edge of M. Repeating the process we end up with a lexicographic matching contained in M.

(ii) Assume that μ is a lexicographic matching and it contains an edge $ab \in E \setminus Z$. Let $a'b$ be an edge in $E_b \setminus \mu \neq \varnothing$. The matching $(\mu \setminus \{ab\}) \cup \{a'b\}$ is lexicographically better than μ, which contradicts its optimality. □

As a consequence of Lemma 8.6, we can select a set of k^2 edges which is guaranteed to contain a lexicographic bottleneck matching. This pruning of the graph can be done in $O(kn)$ time using selection algorithms. Although we do not know whether the graph will be dense or sparse after pruning the non-relevant edges and isolated vertices, we have that both $|B| + |A|$ and $|E|$ are $O(k^2)$. Thus, the best worst-case running time for our scenario is provided by the algorithm of Gabow and Tarjan, which runs in $O(k^2\sqrt{k \log k})$ time, according to the analysis in [27]. The approach in [103], based on solving a sequence of linear sum assignment problems (to be described in the next section) and bottleneck assignment problems, boils down to an algorithm for the computation of a lexicographic bottleneck assignment running in $O(k^4)$ after the pruning.

We state now two lemmas that are well-known in the matching literature and are used by the aforementioned algorithms. We will require them later.

Lemma 8.7 (Berge's Lemma [21])**.** *A matching σ is a maximum matching in a graph G if and only if there is no augmenting path for σ in G.*

Definition 8.8. Let $G = (A, B; E)$ be a bipartite graph and $w : E \to \mathbb{R}^{\geq 0}$ be a function giving weights to the edges. Given $r \in [|E|]$, let

$$E(r) = \{e \in E : w(e) \text{ is among the } r \text{ smallest values of } w(E)\}.$$

We denote by $G(r)$ the graph $G = (A, B; E(r))$.

We state the following easy lemma without proof for future reference.

Lemma 8.9. *Let $G = (A, B; E)$ be a bipartite graph whose maximum matchings have cardinality $|B|$ and $w : E \to \mathbb{R}^{\geq 0}$ be a function giving weights to the edges.*

(i) If $G(r)$ has a matching of size $|B|$, then $G(j)$ has a matching of size $|B|$ for all $j \in [s], j > r$.

(ii) A matching in $G(r^*)$ of size $|B|$ is a bottleneck matching if and only if $G(r^* - 1)$ has no matching of size $|B|$.

8.4 The linear sum assignment problem

Let G be a complete bipartite graph with vertex set partitioned into components A and B with $k = |B| \leq |A| = n$, and edge set $E \subset A \times B$. Given a weight function for the edges $w : E \to \mathbb{R}^+$, the *linear sum assignment problem* seeks a maximum matching having minimum cost, where the cost is the sum of the weights of the edges in the matching. Details concerning this problem and many related ones can be found in the book [27].

The linear sum assignment problem can be formulated as the following min-cost network flow problem:

$$
\begin{aligned}
\min \quad & \sum_{ab \in E} w(ab) x_{ab} \\
\text{subject to:} \quad & \sum_{b \in B} x_{ab} = 1 \qquad \forall a \in A \\
& \sum_{a \in A} x_{ab} \leq 1 \qquad \forall b \in B \\
& x_{ab} \in \{0,1\} \qquad \forall ab \in E \\
& x_{ab} = 0 \qquad \forall ab \notin E,
\end{aligned} \tag{8.1}
$$

where $x_{ab} = 1$ means that a and b are matched together, for $ab \in E$. Since the coefficient matrix is totally unimodular and the costs are integers, an optimal integral solution is guaranteed to exist. Therefore, we can consider the relaxed linear program, in which we do not (explicitly) require the primal variables x to be integral. That is, we can substitute the conditions in (8.1) by $x_{ab} \geq 0$, for all $ab \in E$.

A seminal paper by Kuhn [81] introduced an approach to this problem that would later originate the so-called *primal-dual algorithms*. This algorithm (or family of algorithms) is known as the *Hungarian method*. It starts with a feasible dual solution $(u, v) \in \mathbb{R}^k \times \mathbb{R}^n$ and a partial primal solution $x \in \mathbb{R}^{nk}$ (in which not all the vertices in B are matched) satisfying the complementary slackness conditions. At each iteration, dual feasibility is preserved and the primal solution approaches primal feasibility by trying to increase the cardinality of the corresponding matching using only edges with zero reduced cost. If the augmentation is not possible, the dual variables are updated ensuring that more edges attain reduced cost zero.

The original implementation of the algorithm works for balanced graphs (that is, with $k = n$) and runs in $O(n^4)$ time. Edmonds and Karp [51] and Tomizawa [104] observed that one can apply Dijkstra's algorithm to find the shortest paths that will be used to augment the matching, leading to an $O(n^3)$ algorithm. In a recent technical report by Ramshaw and Tarjan [90], an adaptation of the Hungarian Method to unbalanced bipartite graphs is proposed and analyzed. They use a modification of Dijkstra's algorithm, as described in [58], to find the augmenting paths. The analysis of the careful implementation of the Hungarian method they propose yields a running time of $O(|E|k + k^2 \log k)$, under the hypothesis in our setting. Henceforth, we will refer to the previous specific algorithm as the Hungarian method.

8.5 Outline

In this part, we focus on distances under translations and based on injections from the smaller set into the larger one. We consider, in addition to the optimization problem, a tessellation of the space of transformations that is useful for other types of questions. This tessellation, which we call the *partial-matching Voronoi diagram*, was defined by Rote [93] for the least-squares distance. Roughly speaking, it is the tessellation of the space of translations that groups together the points whose associated positions of the point sets have the same optimal least-squares matchings.

We extend the results by Rote in Chapter 11, deriving a non-trivial bound on the complexity of the least-squares partial-matching Voronoi diagram in any dimension. We study also structural properties of this complex that allow us to design algorithms for its exploration and construction. In Chapter 10, we analyze the analogous tessellation for the bottleneck and lexicographic-bottleneck distances as well, for which polynomial bounds and construction algorithms are derived. Moreover, the provided bounds and running times are sensitive to the size of the smaller set, being some of them optimal if this is constant. We generalize the framework to other transformation spaces and cost functions in Chapter 12, leaving to Chapter 13 the study of several related problems that can be solved taking advantage of the results developed in the preceding chapters.

In order to derive one of the main results of this part, we examine in Chapter 9 a combinatorial problem concerning efficient matchings. We believe that the properties of this purely combinatorial problem are of independent interest.

9

Pareto-efficient matchings for the House Allocation Problem

In this chapter, we present a collection of combinatorial results that will be used later on to derive bounds on the complexity of several partial-matching Voronoi diagrams.

The combinatorial problem we are considering has the flavor of the *Stable Marriage Problem*, which was introduced by Gale and Shapley [60]. It is usually stated as trying to marry the same number of men and women knowing the preferences of every person in a way that no non-married pair would bilaterally want to have an affair. In the classical paper [60] it is shown that such a stable marriage exists for any given set of rankings. In fact, there are in general many stable marriages for a single instance of the problem. Knuth [80] and also Irving and Leather [72] gave some bounds on this number.

Numerous variants of the problem have been studied afterwards (see [74] for a survey). Remarkable examples are stable marriages with incomplete rankings or with ties, the *stable roommates problem* or the *hospital/residents* problem [66]. We are interested in a variant that was first introduced by Shapley and Scarf [100] called the *House Allocation Problem.* This problem receives as input a set of n agents and a set of n houses, together with a ranking of the houses according to each agent. The output should be a one-to-one assignment of houses to agents such that no subset of agents would like to permute their houses. Such an assignment is called a *stable allocation* or a *Pareto-efficient allocation.* The literature around this problem focuses on algorithmic questions, whereas we are interested in bounding the number of stable allocations. Moreover, we need to extend the formulation of the problem in order to deal with instances with fewer agents than houses. We formalize the problem next.

Although the problem can be formulated as a graph matching problem, we will use here the language of permutations and linear orders on the set $[n]$, since it simplifies significantly the notation. Let $\mathcal{P} = (d_1, \ldots, d_k)$ be a list of permutations of $[n]$. This object will be called a (k, n)*-preference* (or *preference*, for short). It may be regarded as k linear orderings on $[n]$, saying for every $i \in [k]$ that $j \leq_i l$ if and only if $d_i^{-1}(j) \leq d_i^{-1}(l)$. The set $[k]$ can be thought of as a set of agents, each of them having a ranking on a set of houses $[n]$. The permutation d_i lists the n houses according to the preference of agent $i \in [k]$, where a house j appears before a house l (or, equivalently, $j \leq_i l$) if the agent i prefers j over l. The target is to assign a house to each person in a "good" way. Note, though, that there is no "best" assignment in general. Throughout this section $\mathcal{P}$ will be considered fixed and we will represent an assignment as an injection $\mu : [k] \hookrightarrow [n]$, which will be called a *matching* as well. Its image $\mu([k])$ will be called the *matched set* of μ.

Definition 9.1. A matching μ is said to be *better* than another matching ν (with respect to a preference $\mathcal{P}$) if $\mu \neq \nu$ and $\mu(i) \leq_i \nu(i)$ for all $i \in [k]$. A matching μ is *Pareto efficient* (or *efficient*, for short) if there is no better matching. We say that an element of $[n]$ is *reachable* if it is contained in the matched set of some efficient matching.

We summarize now some of the properties of efficient matchings and reachable elements.

Proposition 9.2. *Let $\mu \neq \nu$ be two efficient matchings for a preference $\mathcal{P} = (d_1, \dots, d_k)$.*

(i) There exist $i, j \in [k]$ such that $\mu(i) <_i \nu(i)$ and $\nu(j) <_j \mu(j)$.

(ii) If $j \in [n]$ is such that $j \leq_i \mu(i)$ for some $i \in [k]$, then $j \in \mu([k])$.

(iii) If $j \in [n]$ is not among the k smallest values of $\leq_i$ for some $i \in [k]$, then j is not reachable.

(iv) There is no sequence $i_1, \dots, i_m = i_0$ satisfying $\mu(i_{l-1}) <_{i_l} \mu(i_l)$ for all $l \in [m]$.

Proof.

(i) Assume that for all $l \in [k]$ it is $\mu(l) \leq_l \nu(l)$. Then, μ is better than ν and, therefore, ν is not efficient. To avoid the contradiction, it has to exist $i \in [k]$ such that $\nu(i) <_i \mu(i)$. A symmetric argument shows that there is a $j \in [k]$ such that $\mu(j) <_j \nu(j)$.

(ii) Let $j \in [n]$ be such that $j \leq_i \mu(i)$ for some $i \in [k]$ and assume $j \notin \mu([k])$. The matching obtained by modifying μ such that $\mu(i) = j$ is better than μ, which is a contradiction.

(iii) Assume $\mu(i) = j$ and that there are different elements $j_1, \dots, j_k <_i j$. According to (ii), it has to be $\mu([k]) \supset \{j_1, \dots, j_k, j\}$. Hence, $|\mu([k])| > k$, contradicting the fact that μ is a matching.

(iv) Assume that such a sequence exists. Consider the modification μ' of μ obtained by assigning $\mu'(i_l) = \mu(i_{l-1})$ for all $l \in [m]$, and $\mu'(j) = \mu(j)$ for all $j \in [k] \setminus [m]$. Since μ is a matching, we have that $\mu(i_l) \neq \mu(i_{l'})$ for all $l, l' \in [m]$. Consequently, μ' is matching as well. In addition, μ' is better than μ, contradicting the hypothesis that μ is efficient. □

A sequence like the one in (iv) is also known as a *top trading cycle*.

Definition 9.3. Let ρ be a permutation of $[k]$. The *ρ-greedy matching* μ for $\mathcal{P}$ is the unique injection that satisfies $\mu(i) \leq_{\rho(i)} l$ for all $l \in [n] \setminus \{\mu(j) : j \in [k], j < i\}$, for all $i \in [k]$. A matching is called *greedy* if it is ρ-greedy for some permutation ρ.

Intuitively, a greedy matching is obtained by choosing the best free house for the current agent, according to some ordering of the agents. In the game theory literature (see, for instance, [1]), this assignment mechanism is called a *serial dictatorship*. Pareto-efficient matchings are called *Pareto-optima* or *matchings in the core* of the House Allocation Problem as well. The following characterization of efficient matchings appears in [1] only for the case $k = n$. We give a more focused and shorter proof of this result that, in addition, covers the general case $k \leq n$.

Theorem 9.4. *For a given preference $\mathcal{P}$, a matching is efficient if and only if it is greedy.*

Proof. For the *if* direction, let μ be a ρ-greedy matching and let us assume for contradiction that there is a better matching ν. Since ν is better than μ, there exists $i \in [k]$ such that $\nu(\rho(j)) = \mu(\rho(j))$ for all $j < i$ and $\nu(\rho(i)) <_{\rho(i)} \mu(\rho(i))$. This already contradicts the fact that μ is ρ-greedy. Hence, μ is efficient.

For the *only if* direction, assume that μ is an efficient matching. Then, there must be some index $i \in [k]$ such that $\mu(i) = d_i(1)$. Assuming the contrary means, by Proposition 9.2-(iii), that for all $i \in [k]$ there exists an $i' \in [k] \setminus \{i\}$ such that $\mu(i') = d_i(1)$. In particular, there is a cycle $i_1, \dots, i_l = i_0$ such that $\mu(i_{k-1}) = d_{i_k}(1)$, for all $k \in [l]$. This contradicts the statement in Proposition 9.2-(iv) and, thus, the efficiency of μ. We now construct the permutation ρ as follows. Let $\rho(1) = i_1$, where $\mu(i_1) = d_{i_1}(1)$. Consider then the matching $\mu_1 : [k] \setminus \{i_1\} \hookrightarrow [n] \setminus \{\mu(i_1)\}$

$$
\mathcal{P}_{j+1}: \begin{array}{rcl}
d_1: & 1 & \\
d_2: & 2 & \\
 & \vdots & \mathcal{P}_j^1 \\
d_{2^j-1}: & 2^j-1 & \\
d_{2^j}: & 2^j & \\
\hline
d_{2^j+1}: & 1 & \\
d_{2^j+2}: & 2 & \\
 & \vdots & \mathcal{P}_j^2 \\
d_{2^{j+1}-1}: & 2^j-1 & \\
d_{2^{j+1}}: & 2^j &
\end{array}
$$

Figure 9.1: Recursive construction with $\Omega(k \log k)$ reachable elements.

induced by μ. It is an efficient matching, as a better matching would yield a better matching for μ as well. Hence, we can find an element $i_2 \in [k] \setminus \{i_1\}$ such that it is matched to the smallest element in $[n] \setminus \{\mu(i_1)\}$ according to d_{i_2}. We set then $\rho(2) = i_2$. The repetition of this process determines a permutation ρ such that μ is ρ-greedy. □

Corollary 9.5. *The number of efficient matchings with respect to a given preference $\mathcal{P}$ is at most $k!$ and, in general, this bound cannot be improved.*

Proof. By Theorem 9.4, efficient matchings are greedy matchings. There exists only one greedy matching for every permutation of $[k]$. For the lower bound construction, observe that if all the permutations d_i are equal for all $i \in [k]$, then every ρ induces a different greedy matching. □

We investigate now on the number of different matched sets of efficient matchings with respect to a fixed preference. To this end, we introduce the following notation.

Definition 9.6. Let $\mathcal{P}$ be a preference. The set of reachable elements for $\mathcal{P}$ is denoted by

$$R(\mathcal{P}) = \bigcup_{\substack{\mu:[k]\hookrightarrow[n] \\ \mu \text{ efficient for } \mathcal{P}}} \mu([k]).$$

The set of images of efficient matchings for $\mathcal{P}$ is denoted by

$$I(\mathcal{P}) = \{\mu([k]) \mid \mu : [k] \hookrightarrow [n] \text{ efficient for } \mathcal{P}\}.$$

We examine first the set of reachable elements $R(\mathcal{P})$. It follows from Proposition 9.2-(iii) that $|R(\mathcal{P})| \leq k^2$ for any preference $\mathcal{P}$. The following construction gives a lower bound.

Proposition 9.7. *For every $k = 2^l$ with $l \in \mathbb{N}$, there exists a (k, n)-preference $\mathcal{P}$ for some $n \leq k^2$ such that*

$$|R(\mathcal{P})| \geq 2^{l-1}(l+2) = \Omega(k \log k).$$

Proof. We will construct in every step $j = 0, \dots, l$ a preference $\mathcal{P}_j$ consisting of 2^j permutations and having $2^{j-1}(j+2)$ reachable elements. The recursive construction is illustrated in Figure 9.1. For $j = 0$, we consider the trivial preference $\mathcal{P}_0$ having $|R(\mathcal{P}_0)| = 1$. Assuming that we already constructed $\mathcal{P}_j$ and that it reaches $2^{j-1}(j+2)$ elements, we construct $\mathcal{P}_{j+1}$ as follows. We put $d_i(1) = d_{2^j+i}(1) = i$ for all $i \in [2^j]$. The remaining positions of $d_1, \dots, d_{2^j}$ are chosen according to the construction for $\mathcal{P}_j$ but using elements not in $[2^j]$. The remaining positions of $d_{2^j+1}, \dots, d_{2^{j+1}}$ are set similarly using yet another set of elements disjoint from $[2^j]$ and from the ones used in the first copy of $\mathcal{P}_j$. In this way, any element reached by a permutation ρ of $[2^j]$ in the second copy of $\mathcal{P}_j$ (named $\mathcal{P}_j^2$ in the figure) can be reached in $\mathcal{P}_{j+1}$ by a permutation starting with the elements of $[2^j]$ and then continuing as ρ would do it. Any element reached in the first copy of $\mathcal{P}_j$ (named $\mathcal{P}_j^1$ in the figure) can be reached in a similar way. Therefore, the number of elements that can be reached by efficient matchings with respect to $\mathcal{P}_{j+1}$ is at least $2^j + 2|R(\mathcal{P}_j)| = 2^j + 2 \cdot 2^{j-1}(j+2) = 2^j(j+3)$. □

After presenting the results of this chapter in [70], the above combinatorial problem was studied more thoroughly in [10], where it was shown that the previous lower bound construction is asymptotically tight.

Lemma 9.8 (Asinowski, Keszegh and Miltzow [10])**.** *For any (k, n)-preference $\mathcal{P}$,*

$$|R(\mathcal{P})| \leq k(\ln(k) + 1).$$

We now investigate the asymptotic behavior of $I(\mathcal{P})$.

Proposition 9.9. *For every (k, n)-preference $\mathcal{P}$, $|I(\mathcal{P})| = O((e \ln k + e)^k/\sqrt{k})$. There exists a (k, n)-preference $\mathcal{P}$ such that $|I(\mathcal{P})| = \Omega(2^k/\sqrt{k})$.*

Proof. We have that

$$|I(\mathcal{P})| \leq \binom{|R(\mathcal{P})|}{k} \leq \binom{k(\ln(k)+1)}{k} \leq \frac{k^k(\ln(k)+1)^k}{k!} \sim \frac{(k(\ln k + 1))^k}{\sqrt{k}k^k e^{-k}} = \frac{(e(\ln k + 1))^k}{\sqrt{k}}, \tag{9.1}$$

where we used Lemma 9.8 in the second step and Stirling's approximation formula in the fourth one. For the lower bound construction, we will consider only even values of k to avoid floor and ceiling functions. Let $\mathcal{P}$ be such that for every $\leq_i$ the $k/2$ smallest elements are the same (that is, $d_i(m) = d_j(m)$) for all $i, j \in [k]$ and $m \in [k/2]$, and the next elements are pairwise different (that is, $d_i(k/2 + 1) \neq d_j(k/2 + 1)$) for all $i \neq j$. See Figure 9.2 for an illustration. Now, when we greedily choose the smallest elements according to the order of some permutation ρ, in the first $k/2$ steps we choose one of the smallest $k/2$ elements and after exactly $k/2$ steps all of these have been chosen. Then, in the last $k/2$ steps we choose by construction the $(k/2 + 1)$-st smallest element with respect to the corresponding $\leq_i$. Thus, every subset of size $k/2$ is chosen in the second half of some ordering ρ, and hence we have $\binom{k}{k/2}$ different sets in $I(\mathcal{P})$. It is known that this (central) binomial coefficient is asymptotically $\Theta(2^k/\sqrt{k})$. □

Note that, using Stirling's approximation in the three factorials of the binomial coefficient in Equation (9.1), we would could reduce the base of the exponential part. For clarity, we prefer to use the simpler and slightly weaker bound we have derived.

Finally, we prove the following technical lemma, which will be used later.

Lemma 9.10. *Let $\mathcal{P}$ be a (k, n)-preference and $j \in [k] \cup \{0\}$ satisfying the following conditions.*

(1) The first j permutations of $\mathcal{P}$ start with $(1, 2, \ldots, j)$.

(2) The $(j + 1)$-st permutation of $\mathcal{P}$ starts with $j + 1$.

(3) The $k - j - 1$ last permutations of $\mathcal{P}$ start with $(j + 2, \ldots, k)$.

Every efficient matching for $\mathcal{P}$ has matched set $\{1, \ldots, k\}$.

$$\begin{array}{lccccc} d_1: & 1 & 2 & \dots & k/2 & k/2+1 \\ d_2: & 1 & 2 & \dots & k/2 & k/2+2 \\ & & \vdots & & \vdots & \\ d_k: & 1 & 2 & \dots & k/2 & k/2+k \end{array}$$

Figure 9.2: Construction with $2^{k/2}$ reachable sets.

Proof. Observe first that if we have m permutations that coincide in the first m elements, then all these elements must be included in the matched set of any efficient matching. Indeed, since any efficient matching is a greedy matching, it is clear that the first element of each permutation will be included in the matched set. Since the m permutations coincide in the first element, at least $m-1$ of the corresponding elements will not be matched to it. As a consequence of efficiency, it follows that if $m > 2$, the second element of the permutations will be included as well. And again, at least $m-2$ of the corresponding elements will not be matched to it and they will have to go further. Repeating the last arguments, the claim follows. Applying the previous claim to the groups of permutations indicated in the statement, we have that $\{1, 2, \dots, j, j+1, j+2, \dots, k\}$ must be included in the matched set of any efficient matching. Since this is a set of k elements, this is the unique possible matched set. □

10

Bottleneck partial-matching Voronoi diagrams

We study in this chapter the bottleneck partial-matching Voronoi diagram and its lexicographic version. We assume throughout the chapter that we are given two point sets $A, B \subset \mathbb{R}^d$, and that B is allowed to be translated. We will use the term *edge* as a shorthand for a pair of points $(a, b) \in A \times B$ and its *length* will mean the Euclidean distance between them. Our aim is to subdivide the space of translations in a way that we can retrieve, for every translation, one matching whose longest edge is as short as possible, for the corresponding position of B. Using this structure, we will obtain algorithms to find the position of B that best resembles a subset of A in the bottleneck sense, and for other related problems. Recall that an edge $(a, b) \in A \times B$ is denoted by ab for short. We will refer to the *weight* or the *length* of an edge as a (sometimes implicit) function of $t \in \mathbb{R}^d$, the parameter that represents the position $B + t$ of the point set. We also identify a matching with the associated injection of B into A. In the figures, we use white disks to represent the points of B and black ones to represent the points of A.

10.1 Bottleneck matchings under translations

The problem of finding a bottleneck matching for two point sets $A, B \subset \mathbb{R}^d$ can be thought of as a bottleneck assignment problem in a bipartite graph on A and B, where the weight of an edge from $a \in A$ to $b \in B$ is the Euclidean distance between the corresponding points. When we consider the variant under translation, the cost of a matching varies according to a parameter representing the position of one of the point sets. The following definition introduces the main objects of study of this chapter.

Definition 10.1. A *bottleneck matching* for point sets $A, B \subset \mathbb{R}^d$ with $|B| \leq |A|$ is a matching that minimizes

$$f(\sigma) = \max_{b \in B} \|b - \sigma(b)\|, \text{ among all the matchings } \sigma : B \hookrightarrow A.$$

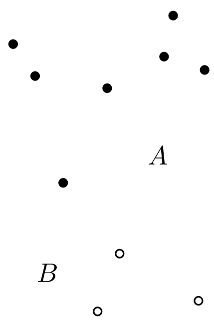
The *bottleneck cost* of a matching $\sigma : B \hookrightarrow A$ is the function

$$f_\sigma(t) = \max_{b \in B} \|b + t - \sigma(b)\|^2, \text{ where } t \in \mathbb{R}^d.$$

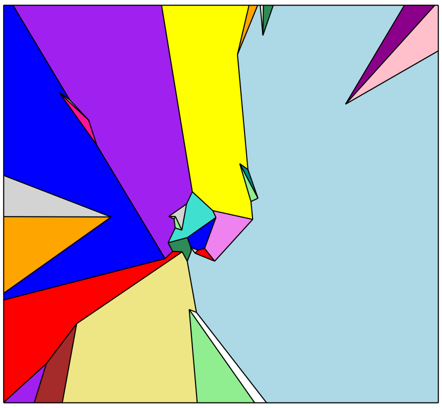
The *bottleneck value* is the function

$$\mathcal{E}(t) = \min_{\sigma : B \hookrightarrow A} f_\sigma(t), \text{ where } t \in \mathbb{R}^d.$$

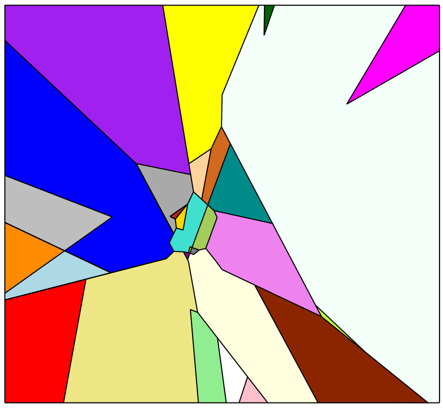
Note that being bottleneck is defined in terms of the Euclidean distance while the functions used depend on the square of this value. This squaring is harmless and will come in handy later.



(a) Two point sets.



(b) The minimization diagram of $\mathcal{E}$.



(c) A coarsening of a bottleneck diagram of A and B.

Figure 10.1: The minimization diagram of $\mathcal{E}$ and a coarsening of a bottleneck diagram for a pair of point sets.

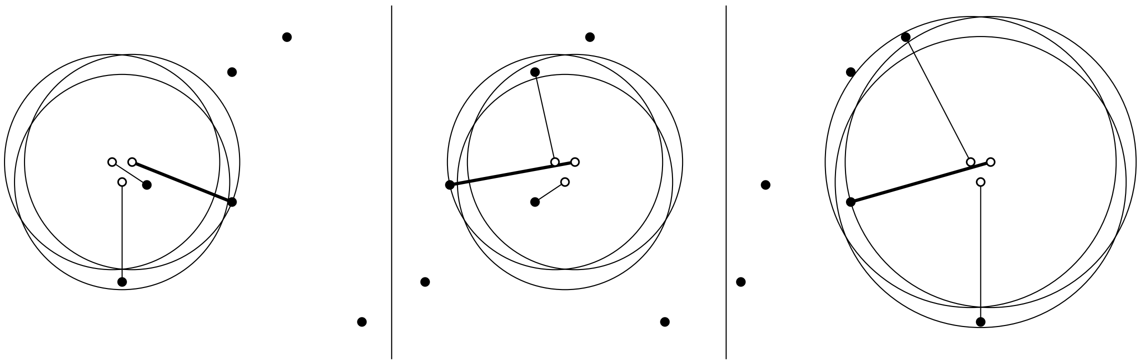


Figure 10.2: Three positions of a pair of point sets and a bottleneck matching for each of them showing that the minimization diagram of the corresponding $\mathcal{E}$ function has a non-convex region.

Figures 10.1(a) and 10.1(b) show a pair of point sets and the partition of the plane induced by the quadratic pieces of the function $\mathcal{E}$. In every polygon the function is quadratic and the regions are colored according to which (unique, in this example) edge attains the bottleneck value. Note that some regions are not convex and some are even disconnected. More precisely, the red and the blue regions consist of two connected components. Figure 10.2 displays a concise proof for the existence of non-convex regions. The pictured disks certify that the drawn edges are the ones attaining the bottleneck value for the three represented positions of the small point set. The three positions correspond to aligned parameters.

Figure 10.2 serves as an example where different positions with the same longest edge (that is, corresponding to faces of the minimization diagram of $\mathcal{E}$ in which the quadratic function is the same) have disjoint sets of bottleneck matchings. This is because the leftmost and the rightmost matchings are the only bottleneck matchings for the corresponding positions, which can be deduced from the position of the points with respect to the drawn circles. Conversely, it can be the case that the same matching is the unique bottleneck matching in two open sets contained in different regions of the minimization diagram of $\mathcal{E}$. An instance of this last situation is illustrated in Figure 10.3. Although the pictured matching is the unique bottleneck matching for the three pictured positions of B, the left and the right translations (and neighborhoods around them) belong to two different cells of the minimization diagram of the associated function $\mathcal{E}$.

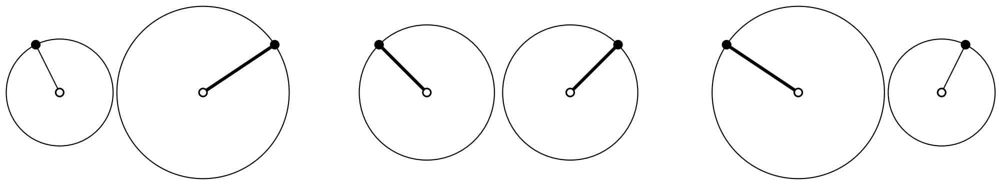


Figure 10.3: A matching that is a bottleneck matching for three positions of B.

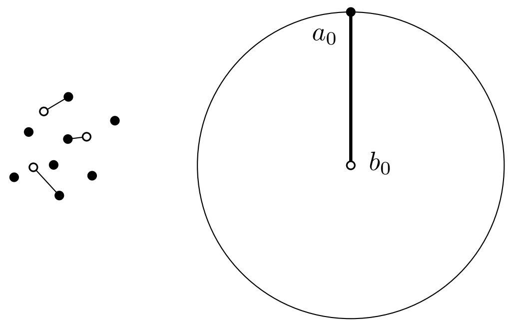


Figure 10.4: A pair of point sets with many bottleneck matchings.

For some applications, it is enough to find a minimum of the function $\mathcal{E}$. For a fixed parameter $t_0 \in \mathbb{R}^d$, a bottleneck matching for $B+t_0$ and A and, hence, the value of $\mathcal{E}(t_0)$, can be computed applying the algorithms described in Section 8.3. A global minimum of $\mathcal{E}$ can be found for point sets of size n in $O(n^5 \log n)$ time applying the algorithm in [52]. However, we are here interested in finding, for every fixed position of the point set B, a matching minimizing the bottleneck cost. To this end, we would like to define a Voronoi-type diagram that allows us to retrieve such a matching for each of its regions. Although the minimization diagram of $\mathcal{E}$ would be a natural candidate to approach this problem, the previous observations imply that it is in general neither a coarsening nor a refinement of such a diagram.

We observe that, even for a fixed position of the point sets, there are in general many bottleneck matchings. Figure 10.4 shows an example that can be easily generalized to show that a set of k points and a set of $n \geq k$ points can have $(n-1)!/(n-k)!$ different bottleneck matchings. Indeed, the point b_0 must be matched to the point a_0 since any other choice would result into a larger bottleneck cost. On the other hand, the remaining points of B can be matched arbitrarily to the remaining points of A.

The bottleneck matchings for the point sets in Figure 10.4 have all the same longest edge. In addition, they remain bottleneck matchings if the point set B is translated anywhere in a neighborhood. However, this is not a necessary condition. That is, there may be different edges that are the longest edge of bottleneck matchings anywhere in a neighborhood of a position, as illustrated in Figure 10.5. The matching to the left of the figure contains two edges attaining the bottleneck value. The edge-disjoint bottleneck matchings in the center and to the right have different edges attaining the bottleneck value.

One attempt to break ties between bottleneck matchings for a given position and be more sensitive to the geometry of the point sets is to consider a lexicographic version. That is, among the matchings whose longest edge is as short as possible, take the one whose second longest edge is as short as possible, and so on.

Definition 10.2. The *lexicographic order* on $\mathbb{R}^k$ is the total order induced by the relation $(x_1, \ldots, x_k) \prec (y_1, \ldots, y_k)$ if and only if there exists an $m \in [k]$ such that $x_i = y_i$ for all $i < m$, and $x_m < y_m$. We write $x \preceq y$ if $x \prec y$ or $x = y$.

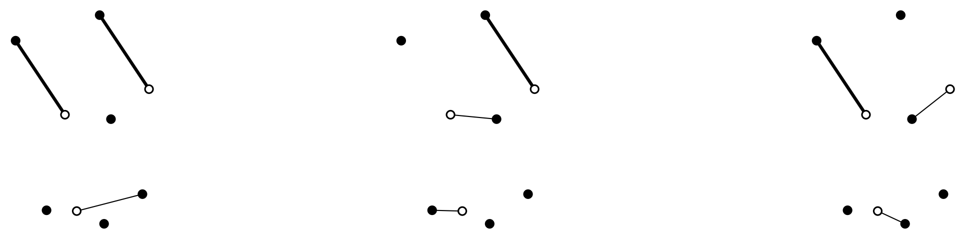


Figure 10.5: Bottleneck matchings with different longest edges.

Definition 10.3. Let $A, B \subset \mathbb{R}^d$ be two finite point sets with $k = |B| \leq |A|$. The *lex-bottleneck cost* of a matching $\sigma : B \hookrightarrow A$ is the function $g_\sigma : \mathbb{R}^d \to \mathbb{R}^k$ where the i-th coordinate of $g_\sigma(t)$ corresponds to the length of the i-th longest edge of σ for $B+t$ and A. A *lex-bottleneck matching* for A and B is an matching π such that $g_\pi(0) \preceq g_\sigma(0)$ for every other matching σ.

Note that a lex-bottleneck matching is a bottleneck matching as well. Although this definition certainly breaks some ties, we will show in the next section that it does not guarantee uniqueness.

10.2 Definitions and basic properties

As seen in the previous section, we face some difficulties to define a Voronoi-type structure for bottleneck matching. One of them is the existence of open sets of translations for which neither the bottleneck matching nor the longest edge are uniquely determined. The problem of the non-uniqueness of the longest edge can be solved by requiring the point set to be in an *ad hoc* general position. The non-uniqueness of the matching may be attacked by considering the lexicographic version of the cost. Nevertheless, it is of interest to study the original version as well in order to solve problems like the ones in Chapter 13 or to explore the minimization diagram of $\mathcal{E}$.

As expected, the Voronoi-type diagrams we are going to study can be required to be made of polyhedral pieces. However, in addition to the non-uniqueness issues, we will also show that the region in which a matching is optimal may be non-convex. Furthermore, for the lexicographic variant, these regions may be neither open nor closed sets. We add then some arbitrariness, for the sake of simplicity, requiring the partition to be given in form of polyhedral complex. This facilitate traversing the partition or optimizing in a region, operations that are often required in related problems. In addition, this demand makes our upper bounds on the complexity of the partitions stronger, whereas it does not weaken the proposed lower bounds and embraces our pathological examples.

Definition 10.4. Let $A, B \subset \mathbb{R}^d$ be two finite point sets with $k = |B| \leq |A| = n$. A *bottleneck partial-matching Voronoi diagram* (or *bottleneck diagram*, for short) for A and B is a polyhedral complex $\mathcal{T}$ covering $\mathbb{R}^d$ and such that for every cell C of $\mathcal{T}$ there is at least one matching $\pi_C : B \hookrightarrow A$ such that $f_{\pi_C}(t) \leq f_\sigma(t)$ for all $t \in C$ and all matchings $\sigma : B \hookrightarrow A$. A *bottleneck labeling* of this diagram is a function mapping each cell to one such matching.

A coarsening of a bottleneck diagram of the point set in Figure 10.1(a) is displayed in Figure 10.1(c), where cells with the same label have the same color. Note that for $B = \{b\}$ the Voronoi diagram of $A - b$ is a bottleneck diagram. Observe also that any polyhedral coarsening refining a bottleneck diagram is a bottleneck diagram, but we cannot choose a tessellation canonically as the coarsest of the bottleneck diagrams because it would be not well-defined.

We define analogously the diagram for the lexicographic version of the bottleneck cost.

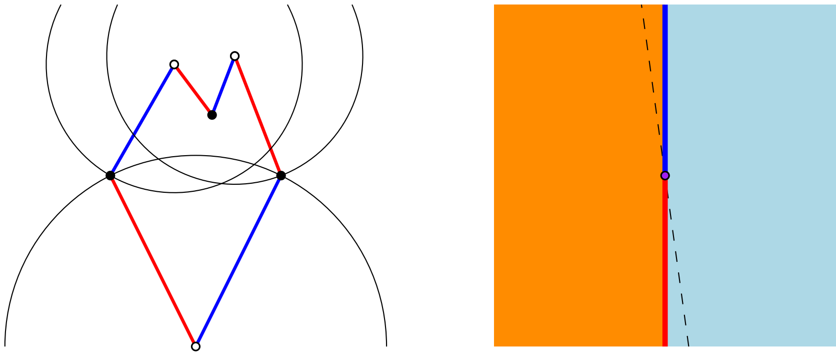


Figure 10.6: A pair of point sets and a lex-bottleneck labeling of the neighborhood of the represented position.

Definition 10.5. Let $A, B \subset \mathbb{R}^d$ be two finite point sets with $k = |B| \leq |A| = n$. A *lex-bottleneck partial-matching Voronoi diagram* (or *lex-bottleneck diagram*, for short) for A and B is a polyhedral complex $\mathcal{T}$ covering $\mathbb{R}^d$ and such that for every face c of $\mathcal{T}$ there is at least one matching $\pi_c : B \hookrightarrow A$ such that $g_{\pi_c}(t) \preceq g_\sigma(t)$ for all t interior to c and all matchings $\sigma : B \hookrightarrow A$. A *lex-bottleneck labeling* of this diagram is a function mapping each face to one such matching.

Note that a bottleneck matching in a lower-dimensional face of a bottleneck diagram is given by the labeling of a cell containing it (that is, the label for a cell is valid everywhere in the cell). This is not the case for a lex-bottleneck diagram, which is the reason why a label for each face is required. Figure 10.6 shows an example. To the left of the figure, the point sets A and B are displayed, for which the blue and the red matchings are both lex-bottleneck matchings. To the right, a small neighborhood in a lex-bottleneck diagram around the point $t \in \mathbb{R}^2$ corresponding to the depicted position of the point sets is represented. If the point set B (the white dots) moves infinitesimally to the right, only the blue matching is a lex-bottleneck matching, whereas if it moves infinitesimally to the left, only the red matching is. This forces the cyan and orange regions to be labeled with the blue and red matchings, respectively. However, if B is vertically translated an infinitesimal amount, the longest blue edge and the longest red edge have the same length. In addition, the second longest red edge is longer than the second longest blue edge if the perturbation is upwards, while it is shorter if the perturbation is downwards, forcing the blue and red regions to be labeled accordingly. For the depicted position (corresponding to the purple point), both matchings are equally good, since the respective shortest edges are equally long. Nonetheless, perturbing infinitesimally the point of A matched by these edges, we can make the lex-bottleneck matching in the purple position to be unique: the red or the blue one.

Since any lex-bottleneck diagram is a bottleneck diagram, we will prove some properties for the first, more restrictive type. However, we will give later algorithms that compute more efficiently a bottleneck labeling than a lex-bottleneck one. For several applications, like the ones included in Chapter 13, the first type of labeling is enough.

Definition 10.6. Given $x, y, z, v \in \mathbb{R}^d$, let

$$h(x, y, z, v) = \left\{ t \in \mathbb{R}^d : 2 \langle t, y - x - v + z \rangle = \|v - z\|^2 - \|y - x\|^2 \right\}.$$

Given two finite point set $A, B \subset \mathbb{R}^d$, let $\mathcal{H}(A, B)$ be the arrangement of the hyperplanes $h(a, b, a', b')$, called *bisectors*, for all pairs $a, a' \in A$ and $b, b' \in B$ such that $b - a \neq a' - b'$.

Proposition 10.7. *For any pair of finite point sets $A, B \subset \mathbb{R}^d$, the arrangement $\mathcal{H}(A, B)$ is a lex-bottleneck diagram.*

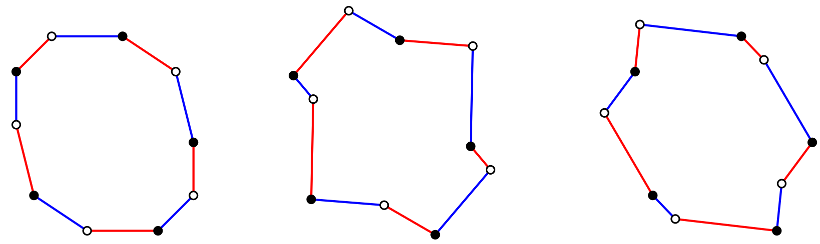


Figure 10.7: Two lex-bottleneck matchings, represented for three positions of B.

Proof. Note that the squared length of an edge matching b to $a = \sigma(b)$ is given by

$$\|b + t - a\|^2 = \|t\|^2 + \|b - a\|^2 + 2\langle t, b - a\rangle .$$

For a pair of edges $ab, a'b' \in A \times B$, the locus of points $t \in \mathbb{R}^d$ for which $\|b + t - a\|^2 = \|b' + t - a'\|^2$ is exactly $h(a, b, a', b')$. If $b - a \neq b' - a'$, this set is a hyperplane. Otherwise, the right hand side vanishes as well, and the set is $\mathbb{R}^d$. Let c be a face in $\mathcal{H}(A, B)$, and let π be a lex-bottleneck matching for a point t_0 in the relative interior of c. Since being a lex-bottleneck matching depends only on the relative length of the edges of the complete bipartite graph, π must be a lex-bottleneck matching as long as no edge becomes strictly shorter than another edge that was strictly longer for t_0. For continuity reasons, this cannot happen in the relative interior of c. Since an arrangement of hyperplanes is a polyhedral complex, $\mathcal{H}(A, B)$ is a lex-bottleneck diagram for A and B. □

Observe that there may be open sets for which two different matchings are lex-bottleneck matchings, as shown in Figure 10.7. This is because every edge from the red matching can be paired with an edge of the blue matching having the same length for any position of the matching. As long as the blue match and the red match of every point in B are its two closest points (as in the three positions represented in the figure), both are lex-bottleneck matchings. Nonetheless, we show below that the matched set of such matchings must be the same.

Proposition 10.8. *Let $A, B \subset \mathbb{R}^d$ be two finite point sets with $|B| \leq |A|$. If two matchings are lex-bottleneck matchings in an open set $U \subset \mathbb{R}^d$, then they have the same matched set and they have the same lex-bottleneck cost for any $t \in \mathbb{R}^d$. In particular, in the interior of a cell of $\mathcal{H}(A, B)$ there is a unique subset $A' \subset A$ that is the matched set of all lex-bottleneck matchings.*

Proof. As argued in the proof of Proposition 10.7, the set of lex-bottleneck matchings is the same for all the translations interior to a cell of $\mathcal{H}(A, B)$. Let C be a cell of $\mathcal{H}(A, B)$, t_0 be a point interior to C, and π and σ be two different lex-bottleneck matchings for $B + t_0$ and A. Consider the symmetric difference of π and σ; that is, the set of edges that belong to π or to σ but not to both. This graph is a collection of (even length) vertex-disjoint paths and cycles alternating edges from π with edges from σ. In addition, since we assume that both matchings are lex-bottleneck matchings, we have that there is a one-to-one correspondence between edges of every path or cycle belonging to π and the edges from the same path or cycle belonging to σ, such that the corresponding edges have the same length. Otherwise, one of the two matchings would not be a lex-bottleneck matching. As it can be derived from the proof of Proposition 10.7, the edges ab and $a'b'$ have the same length over an open set if and only if $b - a = b' - a'$. Thus, there is no path in the symmetric difference, because following the edges in a path we would arrive to the starting vertex, since every edge is "canceled" by the corresponding edge of the other matching. In view of the fact that its symmetric difference is made exclusively of cycles, the sets matched by π and σ must coincide. □

10.3 The complexity of the diagrams

The construction in the proof of Proposition 10.7 leads immediately to a first bound on the complexity of a lex-bottleneck diagram.

Corollary 10.9. *For any pair of point sets $A, B \subset \mathbb{R}^d$ with $|B| = k$ and $|A| = n$, there is a lex-bottleneck diagram of combinatorial complexity $O(n^{2d}k^{2d})$.*

Proof. Proposition 10.7 states that $\mathcal{H}(A, B)$ is a lex-bottleneck diagram for A and B. It is well-known (see, for instance, [47]) that the complexity of an arrangement consisting of m hyperplanes in $\mathbb{R}^d$ is $O(m^d)$. The arrangement $\mathcal{H}(A, B)$ consists of $\binom{n}{2}\binom{k}{2} = O(n^2k^2)$ hyperplanes in $\mathbb{R}^d$. □

We will show that there are some hyperplanes of $\mathcal{H}(A, B)$ that we can safely ignore. Before, we introduce some technical notation resulting of translating the graph-theoretic notions in Definition 8.5 into the geometric setting. Recall that we implicitly refer to a weighted bipartite graph on A and B where the weights of the edges are the Euclidean distances between points of A and the corresponding points of $B + t$, for the position parameter $t \in \mathbb{R}^d$.

Definition 10.10. Let $A, B \subset \mathbb{R}^d$ be two finite point sets with $k = |B| \leq |A|$. Given $t \in \mathbb{R}^d$ and $b \in B$, we define the *candidate function of* b (denoted by $E_b(t)$) as the candidate set of b in graph associated to t. We let $Z(t)$ denote the union of $E_b(t)$ for all $b \in B$.

The following lemma will allow us to improve our complexity bound in low dimensions.

Lemma 10.11 (Sharir [102])**.** *Let $P \subset \mathbb{R}^2$ be a set of m points, and let $s \in [m]$. There are $O(ms)$ bisectors that support all order-j Voronoi edges of P for all $j \leq s$.*

Proof. Consider the set of bisectors defined by pairs of points in P. For each such bisector, we define its conflict set as a set $Q \subset P$ of minimum cardinality such that there is an edge in the Voronoi diagram of $P \setminus Q$ supported by the bisector. The weight of a bisector is the size of its conflict set. Let $N_{<s}(m)$ denote the maximum number of bisectors of weight smaller than s defined by m points in the plane and by $N_0(m)$ the maximum number of bisectors with empty conflict set (that is, the bisectors supporting Voronoi edges of P). The technique of Clarkson and Shor (see [38, 101]) gives $N_{<s}(m) = O(s^2 N_0(m/s))$. Note that a bisector supports an order-j edge for $j \leq s$ if and only if its weight is smaller than s. Hence, the number we want to bound is indeed $N_{<s}(m)$. Since the Voronoi diagram of a point set $R \subset \mathbb{R}^2$ of size m/s has $O(m/s)$ edges, at most that many bisectors can support an edge of weight zero of R. Thus, the desired bound $N_{<s}(m) = O(sm)$ follows. □

We establish first a simple property of the arrangement introduced in Proposition 2.11.

Lemma 10.12. *Let $S \subset \mathbb{R}^2$ be a finite point set and consider the set $\mathcal{Z}$ of planes*

$$z_p = \{(x, y, \|p\|^2 - 2\langle (x, y), p\rangle) : (x, y) \in \mathbb{R}^2\} \subset \mathbb{R}^3, \text{ for all } p \in S.$$

Every three planes in $\mathcal{Z}$ intersect in at most one point. Equivalently, the locus of points equidistant from three different points of S is either a point or empty.

Proof. For every point $q \in \mathbb{R}^2$, the order of the planes z_p over q is the same as the distances from q to the corresponding $p \in S$. The locus of points equidistant from three points in the plane is the center of the circle through them (or the empty set if they are aligned). Hence, three planes can coincide over at most one point. □

Theorem 10.13. *For any pair of point sets $A, B \subset \mathbb{R}^2$ with $|B| = k$ and $|A| = n$, there is a lex-bottleneck diagram of complexity $O(n^2k^6)$. If $A, B \subset \mathbb{R}$, there is lex-bottleneck diagram of complexity $O(nk^3)$.*

Proof. Observe that $E_b(t_1) = E_b(t_2)$ for every pair of points t_1, t_2 interior to a cell C of $\mathcal{H}(A, B)$, for all $b \in B$. Given a cell C of $\mathcal{H}(A, B)$, let $E_b(C)$ and $Z(C)$ denote respectively $E_b(t)$ and $Z(t)$ for any (and every) point t interior to C. By Lemma 8.6-(ii), any lex-bottleneck matching for a point interior to C is contained in $Z(C)$. Let now t_0 be a point in a lower-dimensional face L of $\mathcal{H}(A, B)$, and let $C \supset L$ be a cell. For continuity reasons, the set $E_b(C) \subset E_b(t_0)$ is a b-minimal set at t_0 as well. Therefore, by Lemma 8.6-(i), there is a lex-bottleneck matching for t_0 contained in $Z(C)$.

Consider then a labeling Λ of $\mathcal{H}(A, B)$ that uses in every face of a cell C a matching contained in $Z(C)$. We say that a facet W of $\mathcal{H}(A, B)$ between two cells C_l and C_r *uses* a bisector $h = h(a, b, a', b')$ if $h \supset W$ and $ab, a'b' \in Z(C_l) \cup Z(C_r)$. If no facet uses a bisector h, then h can be omitted from $\mathcal{H}(A, B)$. The resulting hyperplane arrangement is again a polyhedral complex and each new face resulting of merging a set of old faces can be labeled according to whichever of them, since h does not intersect any face whose label in Λ uses the edges ab or $a'b'$. We show now that many bisectors are not used by any facet, distinguishing the two following cases.

Let $h(a, b, a', b)$ be a bisector used by a facet $W = C_l \cap C_r$, and consider the point set $S(b) = A - b$. The facet W must be contained in an order-j Voronoi edge of $S(b)$ for some $j \leq k$. Indeed, in view of Lemma 10.12, infinitesimally to the right or to the left of W both $a - b$ and $a' - b$ are among the $k + 1$ closest points. In addition, for a point t_0 in the relative interior of W, the points $a - b$ and $a' - b$ are the only two points of $S(b)$ that lie at distance $\|b + t_0 - a\| = \|b + t_0 - a'\|$. Hence, there is a circle centered at t_0 and through $a - b$ and $a' - b$ that contains $j - 1 \leq k - 1$ points, which is a characterization for points in the relative interior of edges of the order-j Voronoi diagram.

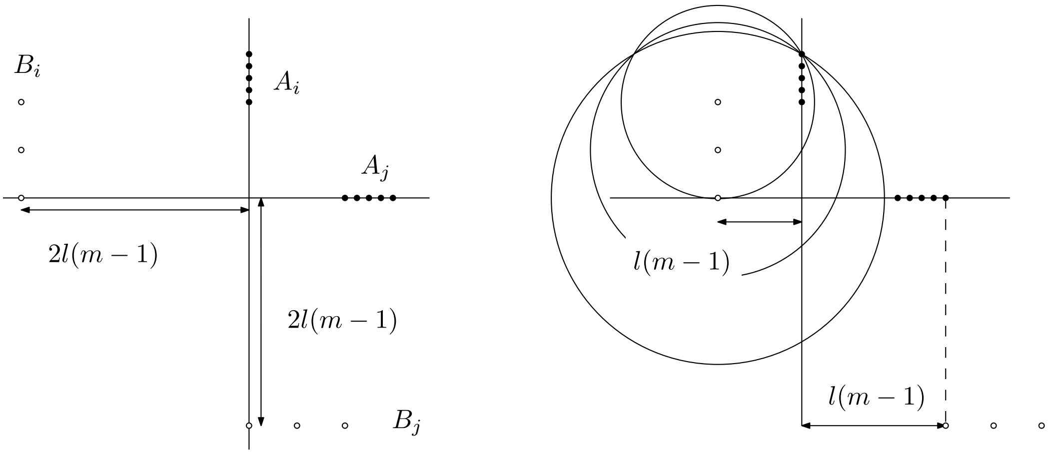
Let $h(a, b, a', b')$ with $b \neq b'$ be a bisector used by a facet $W = C_l \cap C_r$, and consider the point set $S(b, b') = (A - b) \cup (A - b')$. Note that the number of points in $S(b, b')$ is not necessarily $2n$: a point $a_1 - b = a_2 - b'$ can belong to both $A - b$ and $A - b'$. However, this is the case if and only if the edges a_1b and a_2b' are equally long everywhere. In particular, the points $a - b$ and $a' - b'$ are distinct since, otherwise, they would not induce any bisector. Furthermore, simple algebraic manipulations show that $t \in \mathbb{R}^d$ is closer to $a_1 - b$ than to $a_2 - b'$ if and only if $b + t$ is closer to a_1 than $b' + t$ is to a_2, for any choice of $a_1, a_2 \in A$. Since the bisector is used by W, the point $a - b$ is among the k closest points of $A - b$ infinitesimally to at least one of the sides of W. Lemma 10.12 ensures that along the interior of W only the points $a - b$ and $a' - b'$ are at distance $\|b + t_0 - a\| = \|b' + t_0 - a'\|$ among the points in $S(b, b')$, which implies that in fact $a - b$ is among the k closest points of $A - b$ infinitesimally to both sides of W. Similarly, the point $a' - b'$ belongs to the k closest points of $A - b'$ for points infinitesimally apart from W. Hence, for any point t_0 in the relative interior of W, there is a disk centered at t_0 and passing through $a - b$ and $a' - b'$ that contains $j - 1 \leq 2k - 2$ points of $S(b, b')$. Equivalently, the bisector $h(a, b, a', b')$ supports an order-j Voronoi edge of $S(b, b')$ for some $j \leq 2k - 1$.

Applying Lemma 10.11 to $S(b)$ for all $b \in B$ and to $S(b, b')$ for every pair $b, b' \in B$, it follows that the number of bisectors that are used by some edge is $O(k^2 \cdot nk)$. The complexity of the diagram resulting from removing all unused bisectors from $\mathcal{H}(A, B)$ is thus $O(n^2k^6)$.

The case of the line is proven analogously, since the statement and the proof of Lemma 10.11 and the previous arguments carry over easily to the one-dimensional case. $\square$

We provide now a lower bound on the complexity of any lex-bottleneck diagram.

(a) Point sets on the line.



(b) Point sets in the plane.

Figure 10.8: Construction of the lower bound example.

Proposition 10.14. *For any $k, n \in \mathbb{N}$ with $n \geq k \geq d$, there exist point sets $A, B \subset \mathbb{R}^d$ with $|B| = k$ and $|A| = n$ such that any lex-bottleneck diagram for A and B has complexity $\Omega(k^d(n-k)^d)$.*

Proof. We describe first a construction similar to the one used for the lower bound in [93]. Let A be the set of points on the line with coordinates

$$(l-1)(m-1), (l-1)(m-1)+1, \ldots, l(m-1)$$

and B be the set of points with coordinates

$$0, (m-1), 2(m-1), \ldots, (l-1)(m-1),$$

as depicted in Figure 10.8(a). Note that, for every point in B, the l closest points of A are the l leftmost points. When we start translating B to the right, the rightmost point enters the Voronoi region of the $(l+1)$-st point of A. Continuing with the motion, this point traverses the Voronoi region of all the remaining points of A and after that, it has always the same l closest points. A bit later, the second rightmost point of B performs a similar chain of changes, and so do the remaining points in a sequential order. When a point of B is traversing the point set A, the remaining ones have fixed l closest neighbors. In addition, the preferences (in the sense of Chapter 9) induced at every sufficiently general position (that is, such that $\|b+t-a\|$ are distinct for all $b \in B$ and $a \in A$) are of the type studied in Lemma 9.10. Hence, since any lex-bottleneck matching for a sufficiently general translation is efficient, the number of image sets of any lex-bottleneck diagram for A and B must be $l(m-l)$, which proves the statement for $d = 1$.

For $d \geq 2$, we assume that d divides k and n (otherwise, we remove at most $2d$ points to ensure it, without altering the asymptotic bound) and carefully place one copy of the previous example in the direction of every coordinate axis. Specifically, we set

$$A_i = \{(x_1, \ldots, x_d) \in \mathbb{R}^d : x_j = 0, \text{ for all } j \in [d] \setminus \{i\}, x_i = lm - l - m + t, t \in [m]\},$$

$$B_i = \{(x_1, \ldots, x_d) \in \mathbb{R}^d : x_j = -2l(m-1), \text{ for all } j \in [d] \setminus \{i\}, x_i = (t-1)(m-1), t \in [l]\},$$

for each $i \in [d]$. Let $A = A_1 \cup \ldots \cup A_d$ and $B = B_1 \cup \ldots \cup B_d$. We will show that the points in B_i will be matched to A_i in $l(m-l)$ ways as in the one dimensional case independently for every $i \in [d]$. To this end, we must show that when B_i is shifted by $l(m-1)$ in order to generate all the matchings of the one-dimensional case, the other B_j with $j \neq i$ are still matched to A_j in all lex-bottleneck matchings. For this, it is enough to show that for every translation $t \in [0, l(m-1)]^d$ all the points in A_i are closer to B_i than to any point in $A \setminus A_i$ for all $i \in [d]$. First note that to compare such distances, it is enough to look at the projection to the $x_i x_j$-plane, since all the other coordinates coincide for points in B_i and B_j. Then, the situation can be analyzed in the plane, as sketched in Figure 10.8(b). The initial position of B_i has been chosen such that, when translated in the direction of the x_j-axis for $j \neq i$ a distance $l(m-1)$, the distance to the farthest point in A_i is smaller than to the closest point of A_j. This is because $l(m-1)+(l-1)(m-1) > \sqrt{l^2(m-1)^2 + l^2(m-1)^2}$, since $2l-1 > \sqrt{2}l$ if $l > 1$. This shows that there are at least $l^d(m-l)^d = \frac{k^d(n-k)^d}{d^{2d}}$ different lex-bottleneck matchings between A and $B+t$ for $t \in [0, l(m-1)]^d$. The example can be easily perturbed such that $A \cup B$ is in general position. □

10.4 Construction of bottleneck diagrams

In this section, we discuss some algorithmic techniques in order to construct a labeled bottleneck or a lex-bottleneck diagram for a pair of point sets in the plane. To this end, we introduce the following notation.

Definition 10.15. Given two finite point set $A, B \subset \mathbb{R}^2$ (or $A, B \subset \mathbb{R}$), we denote henceforth by $\mathcal{L}(A, B)$ the arrangement constructed as in the proof of Theorem 10.13. Given a cell C of $\mathcal{L}(A, B)$ we denote by $Z(C)$ the set $Z(t)$ for any (and every) point t interior to C.

Lemma 10.16. *The lex-bottleneck diagram $\mathcal{L}(A, B)$ for a pair of point sets $A, B \subset \mathbb{R}^2$ with $k = |B| \leq |A| = n$ can be constructed in $O(n^2k^6)$ time.*

Proof. An arrangement of m lines in the plane can be computed in $O(m^2)$ time using an optimal algorithm, such as the incremental algorithm [36] or a topological sweep [49]. However, the proof of Theorem 10.13 is not constructive and, hence, it is not obvious how to select the bisectors that are used by some edge (in the sense of Theorem 10.13) among the $O(n^2k^2)$ candidates. Fortunately, there exists an algorithm by Chan [34] that constructs the facial structure of the ($\leq s$)-level of an arrangement of m planes in $O(m \log m + ms^2)$ expected time. In addition, this algorithm can be derandomized, leading to a deterministic version running in $O(ms^2(\log m / \log s)^{O(1)})$ time. We can then construct the $O(k^2)$ necessary structures in $O(k^4 n (\log n / \log k)^{O(1)})$ and traverse each of them discovering the "used" bisectors to finally construct their arrangement. □

A similar procedure works for the one-dimensional case.

Lemma 10.17. *The lex-bottleneck diagram $\mathcal{L}(A, B)$ for a pair of point sets $A, B \subset \mathbb{R}$ with $k = |B| \leq |A| = n$ can be constructed in $O(k^2 n(\log n + k \log k))$ time.*

Proof. Similarly to the proof of Lemma 10.16, we will select the vertices that belong to shallow levels in the Voronoi arrangements associated to $S(b)$ and $S(b, b')$, as defined in the proof of Theorem 10.13.

The work in [54] provides us with an algorithm to construct the ($\leq s$)-level of an arrangement of m lines in time $O(m \log m + ms)$. We compute the appropriate levels for each point set and construct an abscissa-sorted list of the projections of its vertices in $O(nk \log k)$ time by traversing the edges of every level and using a priority queue to merge their vertices. Afterwards, we can merge the $O(k^2)$ lists of $O(nk)$ points in $O(nk^3 \log k)$ time using again a priority queue. $\square$

We now show how to find a bottleneck labeling of $\mathcal{L}(A, B)$. Before detailing the algorithm, we need a technical lemma that examines how small changes in a graph affect its bottleneck matchings. We say henceforth that a matching in a bipartite graph is a *complete matching* if it matches all the points in the smaller set of vertices.

Lemma 10.18. *Let $G = (A, B; E; w)$ be a bipartite graph with $w : E \to [|E|]$ giving weights to its edges. Let μ be a bottleneck matching for G, and let $b \in E$ be the longest edge of μ in G. For a fixed $j \in [|E| - 1]$, let $G' = (A, B; E; w')$ where w' coincides with w except that $w'(e) = j$ if $w(e) = j + 1$ and $w'(e') = j + 1$ if $w(e') = j$, for all $e \in E$.*

(i) If $w(b) \notin \{j, j + 1\}$, then μ is a bottleneck matching for G'.

(ii) If $w(b) \in \{j, j + 1\}$ and $G'(j)$ does not have a complete matching, then μ is a bottleneck matching for G'.

(iii) If $w(b) \in \{j, j + 1\}$ and $G'(j)$ has a complete matching ν, then ν is a bottleneck matching for G'.

Proof. Note first that $G(i) = G'(i)$ for all $i \neq j$ and recall the characterization of bottleneck assignments in Lemma 8.9-(ii).

(i) Let $w(b) = l \notin \{j, j + 1\}$. Since μ is a bottleneck matching for G, the graph $G(l)$ has a complete matching and $G(m)$ has not, for any $m < l$. Hence, the graph $G'(l) = G(l)$ has a complete matching and $G'(l-1) = G(l-1)$ has not. In addition, $\mu \subset G'(l)$, which ensures that μ is indeed a bottleneck matching for G'.

(ii) Since we assumed that $G'(j)$ has no complete matching, any complete matching in $G'(j+1)$ is a bottleneck matching for G'. The matching μ is contained in $G'(j+1) = G(j+1)$ because we assumed $w(b) \in \{j, j + 1\}$.

(iii) It is clear that $G'(i) = G(i)$ for all $i < j$ and, hence, it does not have a complete matching. Since $\nu \subset G'(j)$, it is a bottleneck matching for G'. $\square$

As seen in Section 10.1, several edges can have the same length wherever the point set B is translated. However, this happens if and only if all such edges are between points $b \in B$ and $a \in A$ with the same vector $b - a$. To handle also point sets in this special position, we introduce the following equivalence relation.

Definition 10.19. Given finite point sets A and B, two edges $ab, a'b' \in A \times B$ are in the same *class* if $b - a = b' - a'$.

Note that every edge in a class must match a distinct element of B and, hence, the size of every class is at most k. Besides providing some insight into the dynamic behavior of the bottleneck matchings, the previous lemma helps to prove the main result of this section.

Theorem 10.20. *Let $A, B \subset \mathbb{R}^2$ be sets of k and n points. A labeled bottleneck diagram of A and B can be computed in $O(n^2k^8)$ time, and a lex-bottleneck diagram in $O(n^2k^{10})$ time.*

Proof. We will construct the diagrams by labeling the cells (and the faces) of $\mathcal{L}(A, B)$ with a bottleneck (or lex-bottleneck) matching. A naive algorithm to do this would compute such a matching from scratch in every cell. However, we can easily maintain a bottleneck matching during the traversal, improving the time complexity of the algorithm. Unfortunately, this is not the case for the lex-bottleneck diagram, for which the best algorithm we known consists in a traversal of $\mathcal{L}(A, B)$ recomputing (most of) the matching in a number of cells that we are not able to bound apart from the total.

We detail first the algorithm to construct a bottleneck labeling. We first group the edges into classes (that is, groups of edges that have the same length for any fixed translation) as defined above. To do that, we can sort lexicographically in the coordinates the two-dimensional vectors $b - a$ for all $b \in B$ and $a \in A$ in $O(nk \log n)$ time.

Then, we construct $\mathcal{L}(A, B)$ as described in Lemma 10.16, but making sure that we remember for every selected line which are the edges involved. Note however that a line might be selected several times (even from a single one of the arrangements associated $S(b)$ or $S(b, b')$). In such a case, we construct a list containing for each line all pairs of edges that induce it. As a consequence of Lemma 10.12, every pair of edges that induces a fixed bisector of $S(b)$ is counted by Lemma 10.11. That is, if $h(a_1, b, a_2, b) = h(a_3, b, a_4, b)$, then the first pair is counted as an order-j_1 edge and the second pair as an order-j_2 edge of $S(b)$ with $j_2 \neq j_1$. More precisely, since we could infinitesimally perturb the points in $S(b)$ such that $h(a_1, b, a_2, b) \neq h(a_3, b, a_4, b)$ for any choice of different points $a_1, a_2, a_3, a_4 \in A$ without altering the level of the edges inducing them and the bound derived in Lemma 10.11 would be valid, the bound counts already all the pairs inducing the same bisector. Every point in $S(b, b')$, for $b, b' \in B$ with $b \neq b'$, can correspond to two edges that are equally long everywhere. Hence, for each "double" point inducing a bisector, we add the corresponding additional pair of edges to the list of the bisector. Therefore, the total number of elements in the lists associated to bisectors is only a constant factor bigger than the bound on the total number of bisectors we derived. Moreover, we can construct the lists without requiring additional time by adapting the algorithm described in Lemma 10.16.

Assume now that we have already constructed $\mathcal{L}(A, B)$ and the list of pairs of edges associated to each of its bisectors. We choose arbitrarily a cell C of $\mathcal{L}(A, B)$, and pick a point t_0 interior to C (for instance, the centroid of its vertices). We sort the values $\|b + t_0 - a\|$ choosing one representative edge $ab \in A \times B$ from every class. We initialize also a graph G with the k^2 edges of $Z(C)$, since we know by Lemma 8.6-(ii) that it contains one (and every) bottleneck label for C. Moreover, we construct the weight function $w : Z(C) \to [k^2]$ for G representing the order of the lengths of the edges of $Z(C)$ in the relative interior of C. We then find a bottleneck matching in G weighted by w in $O(k^2\sqrt{k \log k})$ time using the Gabow-Tarjan algorithm introduced in Section 8.3.

We will traverse $\mathcal{L}(A, B)$ visiting neighboring cells and maintaining the graph G having as edges $Z(D)$ for the current cell D and the weight function $w : Z(D) \to [k^2]$ encoding the relative lengths of these edges. Let h be the bisector we want to cross. We first look for all the pairs of the type $(ab, a'b)$ in its list. If exactly one of the edges, say $a'b$, is not in the graph G, we should include it and remove ab (because it will not be in the candidate set anymore). The weight of the new edge must the same as the old one (as a consequence of Lemma 10.12). In addition, if the removed edge ab had weight j and belongs to the current bottleneck matching μ whose longest edge b has weight $w(b)$, a new matching having longest edge of weight $w(b)$ can be found in G after including $a'b$ with weight j. In other words, a pair involving edges incident to the same $b \in B$ may change the candidate set replacing an edge by another edge with the same weight, but the weight of the longest edge of the bottleneck matchings remains invariant. Moreover, to find the new bottleneck matching, it is enough to augment $\mu \setminus \{ab\}$ in the graph $G(w(b))$. Such an augmenting path is guaranteed to exist by Lemma 8.7 because $G(w(b))$ has a complete matching. If none of the edges are in G, we do nothing. If both edges are in G, Lemma 10.12 ensures that the edges have consecutive weights and they are swapped after crossing the bisector. Consider now all the pairs in the list of h of the type $(ab, a'b')$ with $b \neq b'$. If at least one of the edges is not in G, we do nothing. Otherwise, the weights must be swapped as in the previous case. Hence, the update of w can be implemented as a sequence of operations as the one described in Lemma 10.18 and the bottleneck matching can be updated accordingly. After each swap, we might need to test if $G'(j)$ (in the notation of Lemma 10.18) has a complete matching. If m edges had weight j in G, the matching $\nu = \mu \cap G'(j)$ can have up to m exposed vertices. We search in $O(k^2)$ time for an augmenting path for ν in $G'(j)$. If there is one, we augment the matching with it and search again. By Lemma 8.7, if there is no augmenting path, the matching is maximum. Therefore, we can either decide that $G'(j)$ does not have a complete matching or find one performing (or trying to perform) at most as many augmentations as pairs in the list of h. After handling all the class swaps, the resulting graph contains the candidate set of edges $Z(D)$ for the new cell D and w expresses the relative order of their lengths. Thus, the bottleneck matching we obtained for the last weighted graph is guaranteed to be a bottleneck matching for any point interior to D.

The number of graph and matching updates performed during the traversal is bounded by the number of edges that $\mathcal{L}(A, B)$ would have if we replace each bisector associated to s pairs of edges by s infinitesimally-separated new lines parallel to it. As argued before, the number of new lines would be still $O(nk^3)$ and, thus, the complexity of this virtual arrangement is $O(n^2k^6)$. We then bound the running time required by the traversal combining this bound with the $O(k^2)$ update time.

To construct a lex-bottleneck labeling, we maintain the weighted graph as in the bottleneck case. We apply in every face of $\mathcal{L}(A, B)$, after updating the function w to indicate the ties that are active in the current face, the algorithm described in [103], which finds a lex-bottleneck matching in $O(k^4)$ time. □

Theorem 10.21. *Let $A, B \subset \mathbb{R}$ be point sets with $k = |B| \leq |A| = n$. A labeled bottleneck diagram of A and B can be computed in $O(nk^2(\log n + k^3))$ time. A labeled lex-bottleneck diagram can be computed in $O(nk^2(\log n + k^5))$ time.*

Proof. We construct the polyhedral complex $\mathcal{L}(A, B)$ in $O(k^2n(\log n + k\log k))$ time as described in the proof of Lemma 10.17, while keeping track of the pairs of edges that are swapped in every bisector, as in the proof of Theorem 10.20.

We then traverse the arrangement updating the bottleneck or lex-bottleneck matching as in the planar case. The number of cells is $O(nk^3)$, as shown in Theorem 10.13, which combined with the updating times studied in the proof of Theorem 10.20 gives the claimed bounds. □

Note that the dependency in n of the bounds of Theorem 10.21 is optimal, since the bottleneck diagram for $k = 1$ is the classic Voronoi diagram, which is known to have time complexity $\Omega(n \log n)$ because it is powerful enough to sort a set of numbers.

In higher dimensions, we do not have good bound for the ($\leq s$)-level of an arrangement. However, the previous proof obviously carries over using $\mathcal{H}(A, B)$ instead of $\mathcal{L}(A, B)$, leading to the following result.

Theorem 10.22. *Let $A, B \subset \mathbb{R}^d$ be sets of k and n points. A labeled bottleneck diagram of A and B can be computed in $O(n^{2d}k^{2d+2})$ time. A labeled lex-bottleneck diagram can be computed in $O(n^{2d}k^{2d+4})$ time.*

11

The least-squares partial-matching Voronoi diagram

In this chapter, we address the problem of partial matching under translations as in Chapter 10, taking now as the cost function the sum of the squared Euclidean distances between matched pairs. In the literature, this measure is called the least-squares distance, the root-mean-squares (RMS) distance or the quadratic mean.

11.1 Least-squares matching under translations

We assume that we are given two point sets $A, B \subset \mathbb{R}^d$ with $|B| \leq |A|$ and we will use the same notation and conventions as in the previous chapter. In the figures, we use white disks to represent B and black disks to represent A.

Definition 11.1. A *least-squares matching* for point sets $A, B \subset \mathbb{R}^d$ with $k = |B| \leq |A| = n$ is a matching that minimizes

$$g(\sigma) = \sum_{b \in B} \|b - \sigma(b)\|^2, \text{ among all the matchings } \sigma : B \hookrightarrow A.$$

The *least-squares cost* of a matching $\sigma : B \hookrightarrow A$ is the function

$$g_\sigma(t) = \sum_{b \in B} \|b + t - \sigma(b)\|^2, \text{ where } t \in \mathbb{R}^d.$$

The *linear part of the least-squares cost* of a matching $\sigma : B \hookrightarrow A$ is the function

$$L_\sigma(t) = \sum_{b \in B} \|b - \sigma(b)\|^2 + 2 \left\langle t, \sum_{b \in B} (b - \sigma(b)) \right\rangle, \text{ where } t \in \mathbb{R}^d.$$

The *least-squares value* is the function

$$\mathcal{G}(t) = \min_{\sigma : B \hookrightarrow A} g_\sigma(t), \text{ where } t \in \mathbb{R}^d.$$

For a fixed position $t \in \mathbb{R}^d$ of B, a least-squares matching for A and $B + t$ is a solution of the linear sum assignment problem described in Section 8.4. Specifically, it is a solution of this problem for the complete bipartite graph on A and B taking as weight for the edge $ab \in A \times B$ the value $\|b+t-a\|^2$. Therefore, it can be computed in $O(nk^2)$. The problem of minimizing this distance under translations (that is, of finding a global minimum of $\mathcal{G}(t)$) is however not well-solved. The Iterative Closest Points [22] algorithm, used for the Hausdorff analogous setup, can be adapted to find a local minimum of $\mathcal{G}(t)$. Nevertheless, the running time of this algorithm is

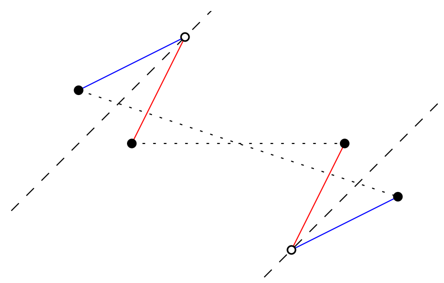


Figure 11.1: Two matchings with the same cost everywhere.

not known to be polynomial. Using some of the structural properties derived in the next section, an efficient algorithm for finding a local minimum of $\mathcal{G}(t)$ is given in [19].

Note that $\mathcal{G}(t)$ is a piecewise quadratic function and it induces a partition of $\mathbb{R}^d$ according to which matchings attain the minimum. However, we face again problems of non-uniqueness, as shown in Figure 11.1. This figure exhibits two matchings that have the same least-squares cost wherever the point set B is translated. This happens because there is a translation for which the points of B lie in bisectors of the points of A and the subsets of A matched by the red and the blue matchings have the same centroid, and hence both costs have the same linear part. The reason why these two conditions are sufficient for the values of the matchings to coincide for every translations becomes clear after reproducing the following observation.

Lemma 11.2 (Rote [93])**.** *Let $A, B \subset \mathbb{R}^d$ be two finite point sets with $|B| \leq |A|$. For any pair of matchings $\sigma, \tau : B \hookrightarrow A$ and any $t \in \mathbb{R}^d$,*

$$g_\sigma(t) < g_\tau(t) \text{ if and only if } L_\sigma(t) < L_\tau(t).$$

As a consequence, the minimization diagram of $\mathcal{G}(t)$ is a regular polyhedral complex covering $\mathbb{R}^d$.

Proof. For a fixed translation $t \in \mathbb{R}^d$ and matching π,

$$g_\pi(t) - L_\pi(t) = |B| \cdot \|t\|^2.$$

Thus,

$$g_\sigma(t) - g_\tau(t) = L_\sigma(t) - L_\tau(t).$$

□

Observe that, for any matching π, $L_\pi(t) = c_\pi + \langle t, v_\pi \rangle$, where $c_\pi = \sum \|b - \pi(b)\|^2$ and $v_\pi = \sum(b - \pi(b))$. The vector v_π depends only on the the centroid of the matched set. Therefore, if two matchings have matched sets with the same centroid, and they have the same cost for some translation $t_0 \in \mathbb{R}^d$, the matchings have the same cost for all $t \in \mathbb{R}^d$. Hence, the two matchings in Figure 11.1 have the same cost everywhere. Note however that these two matching are optimal only along a line segment. The matchings displayed in Figure 10.7 are both, in addition to lex-bottleneck, least-squares matchings over an open set of translations. Observe that these matchings use the same subset of points in A. It will become clear later that this is in fact a necessary condition for two matchings to be optimal in the same open set.

The following lemma states another nice property specific to this cost function, which appeared in [110] and was used in [93]. We provide here a self-contained proof for the sake of completeness.

Lemma 11.3 (Zikan and Silberberg [110])**.** *Let $A, B \subset \mathbb{R}^d$ be two finite point sets with $|A| = |B|$. A matching is a least-squares matching for A and B if and only if it is least-squares for $B + t$ and A for all $t \in \mathbb{R}^d$.*

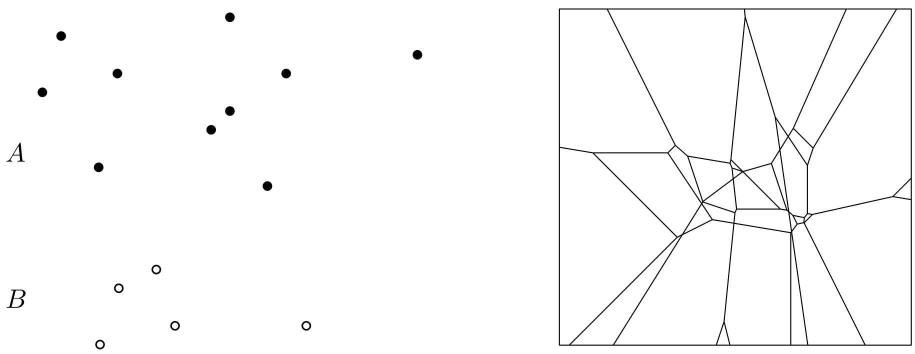


Figure 11.2: Two point sets and its least-squares diagram.

Proof. As already observed, the gradient of L_π depends only on the matched set. Thus, a matching τ will be least-squares if and only if it minimizes $c_\tau = \sum_{b \in B} \|b - \tau(b)\|^2$ and, in such a case, it will be least-squares for any $t \in \mathbb{R}^d$. □

In contrast with the bottleneck diagram, and as a consequence of the previous lemmas, we know that the locus of points where two matchings have the same least-squares cost is either a hyperplane or the whole space. This allows us to define a canonical Voronoi diagram. However, the labeling of the diagram is in general not unique, as argued above.

Definition 11.4. The *least-squares partial-matching Voronoi diagram* (or *least-squares diagram*, for short) of two point sets $A, B \subset \mathbb{R}^d$ is the coarsest polyhedral complex $\mathcal{V}$ covering $\mathbb{R}^d$ such that for every cell $C \in cells(\mathcal{V})$ there is at least one matching that minimizes $g_\sigma(t)$ for all $t \in C$ among all matchings σ. A *labeling* of this diagram is a function mapping each cell to one such matching. Hereafter, we denote this complex by $\mathcal{V}(A, B)$. Equivalently, we can define $\mathcal{V}(A, B)$ as the polyhedral complex induced by

$$V_\pi = \{t \in \mathbb{R}^d : \pi \text{ is a least-squares matching for } B + t \text{ and } A\}.$$

An example of least-squares diagram is illustrated in Figure 11.2.

11.2 The complexity of the diagram

It was claimed by Rote [93] that a global minimum of $\mathcal{G}(t)$ can be found in time proportional, up to a polynomial factor, to the complexity of $\mathcal{V}(A, B)$. The best known upper bound for this complexity, however, was the trivial $O(n^k)$. The great open question is then whether the number of regions of $\mathcal{V}(A, B)$ is polynomial in k and n. This was answered only for dimension one in [93] via the following (more general) result.

Theorem 11.5 (Rote [93])**.** *Let $A, B \subset \mathbb{R}^d$ be two sets with $k = |B| \leq |A| = n$. A line can intersect the interior of at most $k(n-k)+1$ different regions of $\mathcal{V}(A, B)$.*

In the remainder of this section, we focus on establishing a global bound on the complexity of $\mathcal{V}(A, B)$. We begin by deriving the following technical auxiliary results.

Definition 11.6. Let $A, B \subset \mathbb{R}^d$ be two finite point sets. Given $t \in \mathbb{R}^d$, we define $G_t(A, B)$ to be the weighted complete bipartite graph on the vertex set $A \cup B$ and edges $ab \in A \times B$ with weights $w_t(ab) = \|b + t - a\|^2$.

Lemma 11.7. *Let $A, B \subset \mathbb{R}^d$ be two finite point sets. For any $t \in \mathbb{R}^d$, there is a least-squares matching for $B + t$ and A that is an efficient matching for $G_t(A, B)$.*

Proof. Let π be a least-squares matching for $B+t$ and A. If π is not efficient for $G_t(A,B)$, there is a better matching in the sense of Definition 9.1. That is, there exists a matching $\sigma \neq \pi$ such that for all $b \in B$ either $\sigma(b) = \pi(b)$ or $\sigma(b)$ goes before $\pi(b)$ in the permutation associated to b. Then, we have that $\|b+t-\sigma(b)\|^2 \leq \|b+t-\pi(b)\|^2$ for all $b \in B$ and, hence, $g_\sigma(t) \leq g_\pi(t)$. If σ is not efficient, we iterate the argument. Since the relation "being better than" has no cycles, we eventually reach an efficient matching that is least-squares as well. □

We are now ready to present a non-trivial bound on the number of cells of the diagram.

Theorem 11.8. *Let $A, B \subset \mathbb{R}^d$ be two sets with $k = |B| \leq |A| = n$. The number of cells of the least-squares diagram $\mathcal{V}(A,B)$ is $O(n^{2d}k^{d-\frac{1}{2}}(e\ln k + e)^k)$.*

Proof. For a fixed $b \in B$, consider the overlay of the order-j Voronoi diagrams of $A - b$ for all $j \in [k]$. That is, the polyhedral complex $\mathcal{H}_b$ of $\mathbb{R}^d$ such that in every cell C, the k closest neighbors (and their order) is the same for all the points t interior to C. Now consider the polyhedral complex $\mathcal{H}(A,B)$ consisting of the overlay of $\mathcal{H}_b$ for all $b \in B$. The complexity of $\mathcal{H}(A,B)$ must be smaller than the one of the hyperplane arrangement of

$$h(b,a,a') = \{t \in \mathbb{R}^d : \|b+t-a\| = \|b+t-a'\|\},$$

for all $b \in B$ and $a, a' \in A$, which is $O(n^{2d}k^d)$.

Note now that the k shortest edges incident to every $b \in B$ are the same and in the same order, for all $G_t(A,B)$ with t interior to a cell C of $\mathcal{H}(A,B)$. Thus, in view of Proposition 9.2-(iii), all these graphs have the same set of efficient matchings. As a consequence of Lemma 11.7, the $O(k!)$ efficient matchings of $G_t(A,B)$ are then enough to label the portion of $\mathcal{V}(A,B)$ intersected by C. However, the observation in Lemma 11.3 together with Lemma 11.2, imply that it is enough to consider one matching for every possible matched set. Thus, by Lemma 9.8, a cell of $\mathcal{H}(A,B)$ intersects $O((e\ln k + e)^k/\sqrt{k})$ cells of $\mathcal{V}(A,B)$, which yields the claimed bound. □

This theorem will lead, together with a structural property studied in the next section, to a bound on the complexity of the least-squares diagram in any fixed dimension, stated in Corollary 11.13. In the plane, we can obtain a better bound directly.

Theorem 11.9. *Let $A, B \subset \mathbb{R}^2$ be two sets with $k = |B| \leq |A| = n$. The combinatorial complexity of the least-squares diagram $\mathcal{V}(A,B)$ is $O(n^2k^{3.5}(e\ln k + e)^k)$.*

Proof. In the plane, there is a sharper bound on the complexity of $\mathcal{H}_b$ for a fixed $b \in B$, as defined in the previous proof. As a direct consequence of Lemma 10.11, we know that $\mathcal{H}_b$ is a coarsening of the arrangement of $O(nk)$ lines. Thus, $\mathcal{H}(A,B)$ is a coarsening of the arrangement of $O(nk^2)$ lines and, hence, its complexity is $O(n^2k^4)$. Following the observations in the previous proof, we only need to multiply this quantity by $O((e\ln k + e)^k/\sqrt{k})$ to derive a bound on the number of cells of $\mathcal{V}(A,B)$. Since the number of cells of a polyhedral complex in the plane is asymptotically equal to the complexity of the complex, the claimed bound follows. □

A natural question at this point is if this approach of decoupling the combinatorial part and the geometric part can be used to further improve the bounds we obtained. Proposition 9.9 proves that it will not be possible to obtain a polynomial bound even if we manage to count better the number of different images of efficient matchings.

Adapting the proof of Lemma 11.7, one can see that if there are no ties in the values $\|b+t-a\|$ for $b \in B$ and $a \in A$, then any least-squares matching is efficient. Since the ties occur only for a nowhere-dense set of translations, the counterexample in the proof of Proposition 10.14 (and the proof itself) are enough to yield the following lower bound.

Proposition 11.10. *For any $k, n \in \mathbb{N}$ with $n \geq k \geq d$, there exist point sets $A, B \subset \mathbb{R}^d$ with $|B| = k$ and $|A| = n$ such that its least-squares diagram $\mathcal{V}(A,B)$ has $\Omega(k^d(n-k)^d)$ cells.*

11.3 Structural properties of the diagram

We derive here several additional properties of $\mathcal{V}(A,B)$, which show that the diagram has locally polynomial complexity. These properties will help to design algorithms for the construction of $\mathcal{V}(A,B)$ in Section 11.4, and have been used to find a local minimum of $\mathcal{G}(t)$ in [19].

Proposition 11.11. *Let $A, B \subset \mathbb{R}^d$ be two finite point sets. Every facet of the least-squares diagram $\mathcal{V}(A,B)$ has a normal vector of the form $a - a'$ for suitable $a, a' \in A$.*

Proof. Let F be a facet of $\mathcal{V}(A,B)$ common to the regions associated with the matchings π and σ, respectively. By definition, $g_\pi(t) = g_\sigma(t) \leq g_\tau(t)$ for any $t \in F$ and for every matching τ. If $L_\pi(t) = c_\pi + \langle t, v_\pi \rangle$ and $L_\sigma(t) = c_\sigma + \langle t, v_\sigma \rangle$, Lemma 11.2 implies that

$$F \subset h(\pi,\sigma) = \{t \in \mathbb{R}^2 : \langle t, v_\pi - v_\sigma \rangle = c_\sigma - c_\pi\}.$$

As shown in Section 8.2, the set of edges $\pi \oplus \sigma$ is the vertex-disjoint union of cycles and alternating paths, each starting at a matched vertex of π in A that is not matched by σ and ending at a matched vertex of σ in A not matched by π. Let Δ be the set of these cycles and paths. Since the components are vertex-disjoint, they can be "flipped" independently while preserving the validity of the matching; that is, we can choose within any $\gamma \in \Delta$ either all the edges belonging to π or all the edges belonging to σ and the resulting collection of edges still represents an injection from B into A. Observe now that

$$h(\pi,\sigma) = \left\{ t \in \mathbb{R}^2 : \left\langle t, \sum_{\gamma \in \Delta} v_\gamma \right\rangle = -\sum_{\gamma \in \Delta} c_\gamma \right\},$$

where v_γ is the sum of the terms in $v_\pi - v_\sigma$ that involve only the $a \in A$ contained in γ and c_γ is analogously defined for $c_\pi - c_\sigma$. That is,

$$v_\gamma = \sum_{b \in \gamma} (\pi(b) - \sigma(b)) \text{ and } c_\gamma = \sum_{b \in \gamma} \left(\|b - \pi(b)\|^2 - \|b - \sigma(b)\|^2 \right), \text{ for all } \gamma \in \Delta.$$

Note also that $v_\gamma = 0$ for every cycle $\gamma \in \Delta$ and, therefore, there is at least one path $\rho \in \Delta$. In addition, $\langle t, v_\gamma \rangle = -c_\gamma$ for all $\gamma \in \Delta$ and all $t \in h(\pi, \sigma)$. Otherwise, a flip in a path or cycle violating the equation would contradict the optimality of π or σ along $h(\pi,\sigma)$. Therefore, all the vectors v_γ must be linearly dependent. In particular, the direction of $v_\pi - v_\sigma$ is the same as the one of v_ρ. If ρ starts at $a' \in A$ and ends at $a \in A$ then $v_\rho = a' - a$ is normal to F. □

As a consequence, the following statement holds.

Corollary 11.12. *Let $A, B \subset \mathbb{R}^d$ be two sets with $k = |B| \leq |A| = n$. Every cell of the least-squares diagram $\mathcal{V}(A,B)$ has at most $k(n-k)$ facets.*

Proof. As seen in the proof of Proposition 11.11, every facet must have a normal vector of the form $a' - a$ where a belongs to the matching corresponding to the cell and a' does not. We have k options for the first point and $n - k$ for the second one. □

The previous corollary easily implies the following complexity bound.

Corollary 11.13. *Let $A, B \subset \mathbb{R}^d$ be two sets with $k = |B| \leq |A| = n$. The combinatorial complexity of the least-squares diagram $\mathcal{V}(A,B)$ is $O(n^{2d+\lfloor d/2 \rfloor} k^{d+\lfloor d/2 \rfloor - 1/2} (e \ln k + e)^k)$.*

Proof. Theorem 11.8 provides a bound on the number of cells of the diagram. Using the Upper-bound Theorem [83] in combination with Proposition 11.11, each of these cells has combinatorial complexity $O((nk)^{\lfloor d/2 \rfloor})$. □

When we restrict the study to the plane, we get easily more bounds.

Lemma 11.14. *Let $A, B \subset \mathbb{R}^2$ be point sets with $k = |B| \leq |A| = n$.*

(i) The least-squares diagram $\mathcal{V}(A, B)$ contains at most $4k(n-k)$ unbounded regions.

(ii) Every vertex in $\mathcal{V}(A, B)$ has degree at most $2k(n-k)$.

(iii) Any convex path can intersect at most $k(n-k) + n(n-1)$ regions of $\mathcal{V}(A, B)$.

Proof.

(i) Let us take a bounding box that encloses all the vertices of the diagram. By Theorem 11.5, every edge of the bounding box crosses at most $k(n-k)+1$ regions of $\mathcal{V}(A, B)$. The edges of the box traverse only unbounded regions, and cross every unbounded region exactly once, except for the coincidences of the last region traversed by an edge and the first region traversed by the next edge.

(ii) Let v be a vertex of $\mathcal{V}(A, B)$. Draw two generic parallel lines close enough to each other to enclose v and no other vertex of $\mathcal{V}(A, B)$. Each edge adjacent to v is crossed by one of the two lines, and by Theorem 11.5 each of these lines crosses at most $k(n-k)$ edges.

(iii) We use the following property that was observed in Rote's proof of Theorem 11.5. Suppose that we translate B along a line with direction v. Rank the points of A according to v; that is, put $a \prec a'$ if and only if $\langle a, v \rangle < \langle a', v \rangle$ (for simplicity, assume that v is generic so there are no ties). Let $\Phi(\pi)$ denote the sum of the ranks according to $\preccurlyeq$ of the k points of A that participate in the least-squares matching π. As Rote showed, whenever the line enters a new cell of $\mathcal{V}(A, B)$ the value of Φ increases. Let γ be a convex path which, without loss of generality, can be assumed to be polygonal. As we traverse an edge of γ, the value of Φ increases every time we enter a new cell of $\mathcal{V}(A, B)$. When we turn counterclockwise at a vertex of γ, the ranking of $\preccurlyeq$ on A may change, but this change consists of a sequence of swaps of consecutive elements in the present ranking. For each such swap, the value of Φ for the matching at the vertex can decrease by at most one. Since γ is convex, each pair of points of A can be swapped at most twice and, thus, the total decrease in Φ is at most $2\binom{n}{2} = n(n-1)$. Hence, the accumulated increase in Φ, and thus also the total number of regions of $\mathcal{V}(A, B)$ crossed by γ, is at most

$$\Big(n + (n-1) + \ldots + (n-k+1)\Big) - \Big(1 + 2 + \ldots + k\Big) + n(n-1) = k(n-k) + n(n-1). \quad \square$$

11.4 Construction of the least-squares diagram

We first study the intersection of a least-squares diagram with a line in the space of translations. The proof of the following theorem is analogous to the one of Lemma 10.17. We recall first a piece of notation used in the proof of Theorem 11.8.

Definition 11.15. Given a pair of finite point sets $A, B \subset \mathbb{R}^d$, we define $\mathcal{H}(A, B)$ as the overlay of the order-j Voronoi diagrams of $A - b$ for all $j \in [k]$ and all $b \in B$.

Lemma 11.16. *Given a pair of point sets $A, B \subset \mathbb{R}^d$ with $k = |B| \le |A| = n$ and a line $\ell \subset \mathbb{R}^d$, the restriction of $\mathcal{H}(A, B)$ to ℓ can be constructed in $O(nk(\log n + k \log k))$ time.*

Proof. We identify ℓ with the parametrization $\ell : \mathbb{R} \to \mathbb{R}^d$, $\ell(\lambda) = t_0 + \lambda v$, for suitable $t_0, v \in \mathbb{R}^d$. Consider the functions defined on ℓ as

$$z_{ab}(\lambda) = \|b+t_0+\lambda v-a\|^2 - \|t_0+\lambda v\|^2 = \|b-a\|^2 + 2\langle t_0, b-a\rangle + 2\lambda \langle v, b-a\rangle\,, \text{ for all } ab \in A\times B.$$

We regard the graphs of these linear functions as lines in the plane. Then, by simple algebraic manipulations it becomes clear that, for $h(b, a, a')$ as defined in the proof of Theorem 11.8,

$$h(b, a, a') \cap \ell = \{\ell(\lambda) : z_{ab}(\lambda) = z_{a'b}(\lambda)\}.$$

For a fixed point $b \in B$, consider the vertices up to level k of the arrangement of lines induced by z_{ab} for all $a \in A$. Projecting such vertices onto ℓ for every choice of $b \in B$ we obtain the desired arrangement. The work in [54] provides us with an algorithm to construct the ($\le s$)-level of an arrangements of m lines in time $O(m \log m + ms)$. The resulting k sets of $O(nk)$ points in ℓ can be merged then in $O(nk^2 \log k)$ time using a priority queue. □

Using ideas by Rote [94], we get the following lemma and the subsequent theorem.

Lemma 11.17. *Given a pair of point sets $A, B \subset \mathbb{R}^d$ with $k = |B| \le |A| = n$ and points $t_0, v \in \mathbb{R}^d$, a least-squares matching for $B + t_0 + \lambda v$ and A for $\lambda \in \mathbb{R}^+$ arbitrarily large (and for $B + t_0 + \varepsilon v$ and A for $\varepsilon \in \mathbb{R}^+$ arbitrarily small) can be computed in $O(nk^2)$ time.*

Proof. Note that, as explained in Section 8.4, the Hungarian method can be implemented as an iteration of Dijkstra searches for shortest paths. In such searches, the only operations that depend on the weights of the edges are additions and comparisons. The lengths of the edges are quadratic polynomials in λ (respectively, ε). When two expressions depending on λ (or ε) are to be added, we do it as polynomials, without evaluating the variables. The comparisons are evaluated in the limit $\lambda \to +\infty$ (respectively, $\varepsilon \to 0$). To do it for λ, we compare the quadratic part and, in case of a tie, we compare the linear coefficients in λ. If a tie occurs again, we break it with the independent coefficient. In the ε case, we compare first the part independent of ε and, in case of a tie, we compare the linear coefficients in ε. Only in case of a tie, we compare then the quadratic terms. The only ties remaining are between quantities that are equal all along the line and, hence, they lead to matchings with the same cost. □

Theorem 11.18. *Given a pair of point sets $A, B \subset \mathbb{R}^d$ with $k = |B| \le |A| = n$ and a line $\ell \subset \mathbb{R}^d$, the restriction of the least-squares diagram $\mathcal{V}(A, B)$ to ℓ and a labeling for the cells that ℓ intersects can be constructed in $O(nk(\log n + k^3))$ time.*

Proof. We first construct the arrangement $\mathcal{U}$ equivalent to the restriction of $\mathcal{H}(A, B)$ to ℓ as described in Lemma 11.16, and store the list of the k nearest neighbors for each $b \in B$ in every cell. Recall that $\mathcal{V}(A, B)$ is the minimization diagram of the linear functions L_π for all matchings $\pi : B \hookrightarrow A$. Hence, the intersection of $\mathcal{V}(A, B)$ with ℓ is equivalent to the projection of the lower envelope of an arrangement of lines $\mathcal{I}$. We identify ℓ with the parametrization $\ell : \mathbb{R} \to \mathbb{R}^d$, $\ell(\lambda) = t_0 + \lambda v$, for suitable $t_0, v \in \mathbb{R}^d$. We compute a least-squares matching for $\lambda \to +\infty$ and for $\lambda \to -\infty$, as indicated in Lemma 11.17. Let π^+ and π^- be the obtained matchings, which we identify with the corresponding lines ℓ^+ and ℓ^- in $\mathcal{I}$. If $\pi^+ = \pi^-$, the arrangement is trivial. Otherwise, consider the point $t^* \in \ell$ resulting from projecting $\ell^+ \cap \ell^-$. We can locate t^* in $\mathcal{U}$ in $O(\log n)$ time and recover the k^2 relevant edges for this translation. Then, applying the Hungarian method again in $O(k^3)$ time (as detailed in Section 8.4), we either find a new line that belongs to the lower envelope of $\mathcal{I}$ (the one corresponding to the least-squares matching at t^*)

or we conclude that the arrangement has only the two discovered cells. In the second case, we stop. In the first case, we update the lower envelope consisting of ℓ^+ and ℓ^- with the new line we discovered and detect two new tentative vertices of the arrangement. We iterate this process evaluating vertices until no new line is discovered. It is clear that the resulting arrangement is $\mathcal{V}(A,B) \cap \ell$. Every time we execute the Hungarian method is to certify that a vertex belongs to the arrangement or to discover a new cell. Hence, not more than $O(nk)$ executions will be performed. Moreover, the maintenance of the lower envelope is equivalent to the incremental construction of the convex hull of a set of $O(nk)$ points, that can be done in a total running time of $O(nk \log n)$ using the approach in [88]. The total running time of the algorithm is thus $O(nk(\log n + k \log k) + nk(\log n + k^3))$. □

As a trivial implication we obtain the following bound.

Corollary 11.19. *Given a pair of point sets $A, B \subset \mathbb{R}$ with $k = |B| \leq |A| = n$, the least-squares diagram $\mathcal{V}(A,B)$ with a labeling can be constructed in $O(nk(\log n + k^3))$ time.*

Exploiting the structural properties derived in the previous section, we can design an algorithm able to construct the least-squares diagram in any dimension.

Theorem 11.20. *Given a pair of point sets $A, B \subset \mathbb{R}^d$ with $k = |B| \leq |A| = n$, the least-squares diagram $\mathcal{V}(A,B)$ with a labeling can be constructed in $O(nk^4 + nk \log n + (nk)^{\lfloor d/2 \rfloor})$ time for each cell of $\mathcal{V}(A,B)$.*

Proof. We construct the complex by exploring the space from an arbitrary starting cell. We start by choosing a point $t_0 \in \mathbb{R}^d$, which we first assume to be interior to a cell of $\mathcal{V}(A,B)$, and learn the boundary of the cell that contains it. To do that, we compute the least-squares matching π_0 for t_0 in $O(nk^2)$ time as usual. As a consequence of Proposition 11.11, we know that every possible facet of the cell containing t_0 is supported by the hyperplane where an alternating path γ starting at some matched point $a \in A$ and ending at an unmatched point $a' \in A$ has zero cost. That is, the sum of the weights of the edges in $\pi \cap \gamma$ and the sum of the weights of the edges in $(\gamma \setminus \pi)$ are the same. Every possible such path γ generates a hyperplane h_γ, which is the projection of $L_\pi \cap L_{\pi \oplus \gamma}$, where we identify the linear function L_σ for a matching σ with its graph. All the graphs $L_{\pi \oplus \gamma}$ with γ starting at a and ending at a' for a fixed pair $a, a' \in A$, which can be identified with hyperplanes in $\mathbb{R}^{d+1}$, are parallel. Thus, from the hyperplanes h_γ, the one closest to t_0 is the locus of points $\ell_0(a,a')$ for which π_0 and the least-squares matching $\pi(a,a')$ for B and $(\pi_0(B) \cup \{a'\}) \setminus \{a\}$ have the same cost. Hence, π_0 is optimal in the cell of t_0 in the arrangement of all hyperplanes $\ell_0(a,a')$ for all $a \in \pi_0(B), a' \in A \setminus \pi_0(B)$. This cell is then the intersection of $O(nk)$ halfspaces (or the convex hull of $O(nk)$ points, in the dual), which can be computed in $O(nk \log n + (nk)^{\lfloor d/2 \rfloor})$ using the algorithm of Chazelle [35]. To compute the least-squares matchings in order to learn every $\ell_0(a,a')$ we use the Hungarian method, which requires $O(k^3)$ time in this (balanced) case.

We then learn the least-squares matching corresponding to the cells adjacent to the cell of t_0. If the hyperplane $\ell_0(a,a')$ supports a facet and it is not repeated (that is, if $\ell_0(a,a') \neq \ell_0(a'',a''')$ for any other pair $a'', a''' \in A$), the other cell of $\mathcal{V}(A,B)$ sharing the facet supported by $\ell_0(a,a')$ corresponds to the already computed matching $\pi(a,a')$. However, if more than one $\ell_0(a,a')$ coincide, there might be an exponential number of candidates to least-squares matchings on the other side. In such a case, we apply a perturbation technique. More precisely, if v is a vector perpendicular to $\ell_0(a,a')$ and $\bar{t}$ is a point interior to the computed facet (which can be taken to be the centroid of its vertices), we execute the Hungarian method for $t = \bar{t} + \varepsilon v$, with ε an arbitrarily small positive real number, as indicated in Lemma 11.17. Once we know the least-squares matchings in the neighboring cells, we can continue constructing the arrangement repeating the previous procedure.

If the point t_0 we picked as initial point was not interior to a cell of $\mathcal{V}(A, B)$, we might have found a least-squares matching that is least-squares only in a lower-dimensional face F of $\mathcal{V}(A, B)$. If this is the case, we notice it after computing the polytope where the matching is still optimal. Then, we take a point in the relative interior of F and move an ε amount in a direction not contained in the affine hull of the face. We compute a least-squares matching for this symbolic point and repeat the process of learning the polytope for which this matching is optimal. The obtained polytope should be of strictly higher dimension than the previous one. Repeating this process at most d times, we end up learning a cell and the algorithm as described above can be then applied. □

Combining the previous bound with Theorem 11.8, we get the following statement.

Corollary 11.21. *Given a pair of point sets $A, B \subset \mathbb{R}^d$ with $k = |B| \leq |A| = n$, the least-squares diagram $\mathcal{V}(A, B)$ with a labeling can be constructed in time*

$$O(n^{2d+1}k^{d+1/2}(e \ln k + e)^k(k^3 + \log n + (nk)^{\lfloor d/2 \rfloor - 1})).$$

For $d = 2$, it can be constructed in $O(n^3 k^{7.5}(e \ln k + e)^k)$. For $d = 3$, in $O(n^7 k^{6.5}(e \ln k + e)^k)$.

Proof. The bound for general dimension is obtained by multiplying the bound on the complexity of $\mathcal{V}(A, B)$ derived in the previous section with the time required per cell studied in Theorem 11.20. In dimensions two and three, there exist output-sensitive algorithms for computing the convex hull of m points in $O(m \log h)$ time, where h is the number of vertices of the computed polytope (see [33, 77, 37]). If we use one of these algorithms, denoting by $h(C)$ the number of vertices of a cell C, the time required to perform all the convex-hull computations is

$$\sum_{C \in cells(\mathcal{V}(A,B))} k(n-k) \log h(C) \leq k(n-k) \log \left(\sum_{C \in cells(\mathcal{V}(A,B))} h(C) \right),$$

where we use that the logarithm is a concave function. Since the number of vertex-cell incidences is at most the square of the complexity of the complex, the total time to compute the required convex hulls is $O(kn \log(T(\mathcal{V}(A, B)))$, where $T(\mathcal{V}(A, B))$ is the complexity of $\mathcal{V}(A, B)$. Thus, we can express the total time required to construct the diagram as

$$O\big(T(\mathcal{V}(A, B))(nk^4 + (nk)^{\lfloor d/2 \rfloor}) + kn \log(T(\mathcal{V}(A, B)))\big).$$

Using the general complexity bound for the three-dimensional case and the improved complexity bound of Theorem 11.9 for the plane, we derive the stated bounds. □

12

Generalizations

In this chapter, we explore how far can the results from this part be extended. Recalling how the bound on the number of cells of the least-squares diagram was derived, two independent steps can be distinguished. First, we decomposed the space of translations into equivalent cells and obtained a bound for its number. Then, we observed that any least-squares matching must be efficient and used the bound for the number of efficient matchings of a (k, n)-preference. That is, we decoupled a geometric part of the analysis from a combinatorial part. For lex-bottleneck or bottleneck diagrams the situation was similar albeit simpler. We develop now a more general framework that will allow us to extend the bounds to a wider set of cost functions and transformation spaces. Nonetheless, it is beyond the aim of this chapter to give an exhaustive list of general results. Instead, we give only some illustrative examples. In particular, we do not combine the different directions of generalization we explore, for the ease of presentation. The following definition sets the notion of preference for the geometric setup, analogous to the one in Chapter 9.

Definition 12.1. Let $A, B \subset \mathbb{R}^d$ be two finite point sets. The *A-preference* of B is the set of permutations of the points in A according to the distances $\|b - a\|$, for each $b \in B$. The permutation of the pairs $ab \in A \times B$ according to the values $\|b - a\|$ will be called the *total preference* of A and B.

12.1 Other transformation spaces

A first direction in which the previous results can be extended is allowing B to be transformed by a more general set of functions, instead of only by translations. The main target of this section is to give tools for the study of how the preference and the total preference change depending on the transformation applied to B. Hereafter, Γ will denote a real manifold whose points are the allowed transformations. For a transformation $\gamma \in \Gamma$, we denote by $\gamma(B)$ the result of applying the transformation γ to each of the points in B. We now extend some objects examined in previous chapters to this setting.

Definition 12.2. Let $A, B \subset \mathbb{R}^d$ be two finite point sets, and let Γ be a real manifold whose points are transformations $\gamma : \mathbb{R}^d \to \mathbb{R}^d$. A *bottleneck diagram of A and B under* Γ is a partition $\mathcal{V}$ of Γ such that in every set $S \in \mathcal{V}$ there is at least one matching that is a bottleneck matching for $\gamma(B)$ and A for all $\gamma \in S$. A bottleneck labeling of $\mathcal{V}$ is a function mapping every set of $\mathcal{V}$ to one such matching. *Lex-bottleneck diagrams under* Γ and *least-squares diagrams under* Γ are defined analogously for the corresponding cost functions.

The following definition is motivated by the observations introducing this chapter.

Definition 12.3. Let $A, B \subset \mathbb{R}^d$ be two finite point sets. The *bisector of $a, a' \in A$* is the set

$$\mathcal{B}(a, a') = \{x \in \mathbb{R}^d : \|x - a\|^2 = \|x - a'\|^2\}.$$

Given $b, b' \in B$, the (b, b')-*pullback* of the bisector $\mathcal{B}(a, a')$ to the transformation space Γ is

$$\mathcal{B}(b, b'; a, a') = \{\gamma \in \Gamma : \|\gamma(b) - a\|^2 = \|\gamma(b') - a'\|^2\}.$$

We denote by $\mathcal{P}(\Gamma, A, B)$ the arrangement in Γ induced by the pullbacks $\mathcal{B}(b, b; a, a')$, for $a, a' \in A$ and $b \in B$. We denote by $\mathcal{T}(\Gamma, A, B)$ the arrangement in Γ induced by $\mathcal{B}(b, b'; a, a')$, for $a, a' \in A$ and $b, b' \in B$. We abbreviate $\mathcal{P}(\Gamma, A, B)$ by $\mathcal{P}$ and $\mathcal{T}(\Gamma, A, B)$ by $\mathcal{T}$ whenever the transformation space Γ and the involved point sets are clear from the context. By construction, two transformations $\gamma_1, \gamma_2 \in \Gamma$ lying in the same cell of $\mathcal{P}$ map the point set B in a way that the A-preference of $\gamma_1(B)$ and $\gamma_2(B)$ is the same. Similarly, points lying in the same region of $\mathcal{T}$ correspond to transformations inducing the same total preference.

Next, we study several specific transformation spaces and the associated $\mathcal{P}$-type arrangements. Let $\text{Aff}(d)$ be the linear space of affinities from $\mathbb{R}^d$ to $\mathbb{R}^d$. Let $\text{U-Scal}(d)$ be the set of uniform scalings, $\text{Scal}(d)$ be the set of scalings (with scale factor independent for each coordinate axis), and $\text{Hom}(d)$ be the set of homotheties (compositions of a scaling with a translation); all of them linear subspaces of $\text{Aff}(d)$. Consider also the following parametrization of each space where the parameters are real vectors indicated in brackets and we represent the functions writing its image for a generic point $x \in \mathbb{R}^d$.

$$\begin{aligned}
\text{U-Scal}(d) &= \{f[\lambda](x) = \lambda x : \lambda \in \mathbb{R}\} \\
\text{Hom}(d) &= \{f[\lambda, t](x) = \lambda x + t : \lambda \in \mathbb{R}, t \in \mathbb{R}^d\} \\
\text{Scal}(d) &= \{f[\lambda_1, \ldots, \lambda_d](x) = (\lambda_1 x_1, \ldots, \lambda_d x_d) : \lambda_1, \ldots, \lambda_d \in \mathbb{R}\} \\
\text{Aff}(d) &= \{f[M, t](x) = Mx + t : t \in \mathbb{R}^d, M \in \mathbb{R}^{d \times d}\}
\end{aligned}$$

We next study the arrangements $\mathcal{P}$ for the transformation spaces defined above. Note that we identify $\text{Aff}(d)$ with the space of its parameters $\mathbb{R}^{d \times d} \times \mathbb{R}^d \cong \mathbb{R}^{d(d+1)}$, and its subspaces are identified accordingly.

Proposition 12.4. *Let $A, B \subset \mathbb{R}^d$ be two point sets with $k = |B| \leq |A| = n$.*

(i) The arrangement $\mathcal{P}(\text{Aff}(d), A, B)$ is a polyhedral complex of complexity $O((n^2k)^{d(d+1)})$.

(ii) The arrangements $\mathcal{P}(\text{U-Scal}(d), A, B)$, $\mathcal{P}(\text{Scal}(d), A, B)$ and $\mathcal{P}(\text{Hom}(d), A, B)$ are polyhedral complexes of complexities $O(n^2k)$, $O(n^{2d}k^d)$ and $O((n^2k)^{d+1})$, respectively.

Proof.

(i) The (b, b)-pullback of the bisector $\mathcal{B}(a, a')$ to Γ is

$$\mathcal{B}(b, b; a, a') = \{(M, t) \in \mathbb{R}^{d(d+1)} : 2\left\langle Mb + t, a' - a\right\rangle = \|a\|^2 - \|a'\|^2\}.$$

Since $\langle Mb + t, a' - a\rangle$ is linear in the entries of M and t, the pullback is identified with a hyperplane in $\mathbb{R}^{d(d+1)}$. It is well-known that the complexity of an s-dimensional arrangement of h hyperplanes is $O(h^s)$ (see, for instance, [47]). Since the number of pullbacks to consider is $k\binom{n}{2} = O(n^2k)$, the bound follows.

(ii) The transformation spaces $\text{U-Scal}(d)$, $\text{Scal}(d)$ and $\text{Hom}(d)$ are linear subspaces of $\text{Aff}(d)$ of dimension 1, d and $d+1$, respectively. Thus, the arguments from (i) lead to the claimed complexities. □

The following lemma shows that, as for translations, there is a canonical diagram for least-squares matchings under affinities.

Lemma 12.5. *For any pair of finite point sets $A, B \subset \mathbb{R}^d$, there is a (regular) polyhedral complex that is the coarsest least-squares diagram of A and B under* $\mathrm{Aff}(d)$.

Proof. For a given matching π, consider the set

$$V_\pi = \{(M,t) \in \mathbb{R}^{d(d+1)} : \pi \text{ is a least-squares matching for } f[M,t](B) \text{ and } A\}.$$

We will show that V_π is a polyhedron. If there is no $(M_0, t_0) \in \mathbb{R}^{d(d+1)}$ for which π is least-squares for $f[M_0, t_0](B)$ and A or it is for all $(M,t) \in \mathbb{R}^{d(d+1)}$, the set V_π is empty or the whole space. Otherwise, let (M_0, t_0) be such a pair and let $(M_1, t_1) \in \mathbb{R}^{d(d+1)}$ be such that $\pi_1 \neq \pi$ is least-squares for $f[M_1, t_1](B)$ and A. The least-squares cost of a matching σ for a generic pair $(M,t) \in \mathbb{R}^{d(d+1)}$ is

$$\begin{aligned} \tilde{g}_\sigma(M,t) : \mathbb{R}^{d(d+1)} &\to \mathbb{R} \\ (M,t) &\mapsto \sum_{b \in B} \|Mb + t - \sigma(b)\|^2. \end{aligned}$$

Developing the squared Euclidean norms, we have that

$$\tilde{g}_\sigma(M,t) = \sum_{b \in B} \|Mb + t\|^2 - 2\sum_{b \in B} \langle Mb + t, \sigma(b)\rangle + \sum_{b \in B} \|\sigma(b)\|^2.$$

Hence, $\tilde{g}_\pi(M,t) - \tilde{g}_{\pi_1}(M,t)$ is a linear function in the entries of (M,t). As in the case of translations, the minimization diagram of the functions $\tilde{g}_\sigma$ is the same as the one of their linear part. Such minimization diagram is a polyhedral complex and, in addition, is the obviously the coarsest tessellation that is a least-squares diagram for A and B. □

The following theorem shows that the previous property is inherited by the linear subspaces of $\mathrm{Aff}(d)$.

Theorem 12.6. *Let $A, B \subset \mathbb{R}^d$ be two point sets with $k = |B| \leq |A| = n$. There are polyhedral complexes that are least-squares diagram for A and B under* $\mathrm{Aff}(d)$, $\mathrm{U\text{-}Scal}(d)$, $\mathrm{Scal}(d)$ *and* $\mathrm{Hom}(d)$ *with $O((n^2k)^{d(d+1)}k!)$, $O(n^2k(e\ln k + e)^k/\sqrt{k})$, $O(n^{2d}k^dk!)$ and $O((n^2k)^{d+1}k!)$ cells, respectively.*

Proof. Consider the polyhedral complex $\mathcal{V}$ induced by the cells V_π as defined in the proof of Lemma 12.5. A cell of $\mathcal{P}(\mathrm{Aff}(d), A, B)$ can intersect the interior of at most $k!$ cells of $\mathcal{V}$. As a matter of fact, all the transformations in a cell of $\mathcal{P}(\mathrm{Aff}(d), A, B)$ induce the same A-preference and, thus, the efficient matchings for this preference are enough to label the portion of the diagram intersected by the cell. For the remaining transformation spaces, consider the result of intersecting the linear subspace of $\mathbb{R}^{d(d+1)}$ they are parametrized by with $\mathcal{V}$. These polyhedral complexes are least-squares diagrams for the corresponding transformation spaces. In addition, the restrictions of $\mathcal{P}(\mathrm{Aff}(d), A, B)$ to these subspaces are arrangements of hyperplanes in them. Every cell of these arrangements can intersect the interior of at most $k!$ cells of $\mathcal{V}$. For the case of uniform scaling, a better bound can be easily derived if we observe that it is enough to use one label for every matched set, as in the case of translations. This is because, for equally sized sets, the property of being least-squares matchings is invariant under (uniform) scalings [13, 110]. Combining Proposition 12.4 with the previous observations, the claimed bounds follow. □

The situation is more complicated for the non-linear submanifolds of $\mathrm{Aff}(d)$, such as isometries and rigid motions, denoted by $\mathrm{Iso}(d)$ and $\mathrm{Rig}(d)$ and parametrized as indicated below.

$$\mathrm{Iso}(d) = \{f[M,t](x) = Mx + t : t \in \mathbb{R}^d\ , M \in \mathbb{R}^{d\times d}, M^{-1} = M^{\intercal}\}$$
$$\mathrm{Rig}(d) = \{f[M,t](x) = Mx + t : t \in \mathbb{R}^d\ , M \in \mathbb{R}^{d\times d}, M^{-1} = M^{\intercal}, \det(M) = 1\}$$

Although the least-squares diagrams under these transformations are not necessarily polyhedral, a bound on the number of their "faces" can be derived using machinery from algebraic geometry. To this end, we adapt some definitions appearing, for instance, in [17].

Definition 12.7. Let $\mathcal{Q} = \{q_1(x), \ldots, q_m(x)\}$ be a set of polynomials. A *sign condition* is a vector $\Sigma \in \{+, -, 0\}^m$. For a variety $\mathcal{Z}$ and a sign condition Σ, the set

$$\Sigma_{\mathcal{Q}}^{\mathcal{Z}} = \{x \in \mathcal{Z} : (\mathrm{sign}(q_1(x)), \ldots, \mathrm{sign}(q_m(x)) = \Sigma\}$$

is called the *realization space* of Σ over $\mathcal{Z}$ and its non-empty (in $\mathcal{Z}$) semialgebraic connected components are called the *faces* of Σ. The union of faces of of all sign conditions (with non-empty realization space over $\mathcal{Z}$) is called the *algebraic arrangement* of $\mathcal{Q}$ over $\mathcal{Z}$. The number of faces in the arrangement is called its *combinatorial complexity.*

Theorem 12.8 (Basu, Pollak and Roy [17])**.** *Let $\mathcal{Z} \subseteq \mathbb{R}^V$ be an s-dimensional real variety that is the zero-set of a polynomial of degree D in V variables. Consider m subvarieties of $\mathcal{Z}$ each defined by a polynomial of degree at most D (in the same variables). The combinatorial complexity of the algebraic arrangement of these m subvarieties over $\mathcal{Z}$ is $\binom{m}{s}O(D)^V$.*

We are now ready to prove the following statement.

Proposition 12.9. *Let $A, B \subset \mathbb{R}^d$ be two point sets with $k = |B| \leq |A| = n$. The algebraic arrangements $\mathcal{P}(\mathrm{Iso}(d), A, B)$ and $\mathcal{P}(\mathrm{Rig}(d), A, B)$ have combinatorial complexity $O((n\sqrt{k})^{d(d+1)})$, for any fixed dimension d.*

Proof. It is well-known that the set of orthogonal $d\times d$ matrices is a $(d(d-1)/2)$-dimensional real algebraic subvariety (defined by quadratic polynomials) of the space of matrices. This variety has two connected components corresponding to the matrices M satisfying $\det(M) = 1$ and the ones satisfying $\det(M) = -1$. Taking into account the translation part, it follows that $\mathrm{Iso}(d)$ is a real algebraic variety of dimension $d(d+1)/2$. In addition, as observed in [17, Remark 1], this variety may be defined by a single polynomial of degree 4 in $d(d+1)$ variables. As we have already seen in the proof of Proposition 12.4, the bisectors $\mathcal{B}(b, b; a, a')$ are hyperplanes in $\mathbb{R}^{d(d+1)}$ and, hence, Theorem 12.8 applies with $\mathcal{Z} = \mathrm{Iso}(d)$, $s = d(d+1)/2$, $D = 4$, $V = d(d+1)$ and $m = k\binom{n}{2}$. Thus, the complexity of the arrangements $\mathcal{P}(\mathrm{Iso}(d), A, B)$ and $\mathcal{P}(\mathrm{Rig}(d), A, B)$ is

$$\binom{\binom{n}{2}k}{d(d+1)/2} O(1)^{d(d+1)} = O\left((n\sqrt{k})^{d(d+1)}\right). \qquad \square$$

As a consequence, we have the following bounds.

Theorem 12.10. *Let $A, B \subset \mathbb{R}^d$ be two point sets with $k = |B| \leq |A| = n$. There are least-squares diagrams for A and B under* $\mathrm{Iso}(d)$ *and* $\mathrm{Rig}(d)$ *that can be labeled by $O((n\sqrt{k})^{d(d+1)}k!)$ matchings, for any fixed dimension d.*

With help of Theorem 12.8, we can study the complexity of the $\mathcal{T}$-type arrangements for the transformation spaces defined above and thereby derive bounds for the complexity of the corresponding lex-bottleneck diagrams.

Theorem 12.11. *Let $A, B \subset \mathbb{R}^d$ be two point sets with $k = |B| \leq |A| = n$. There are lex-bottleneck diagrams for A and B under* Aff(d), U-Scal(d), Scal(d), Hom(d), Iso(d) *and* Rig(d) *that are algebraic arrangements of combinatorial complexity $O((n^2k^2)^{d(d+1)})$, $O(n^2k^2)$, $O((n^2k^2)^d)$ and $O((n^2k^2)^{d+1}))$, $O((nk)^{d(d+1)})$ and $O((nk)^{d(d+1)})$, respectively.*

Proof. The pullbacks $\mathcal{B}(b, b'; a, a')$ to the space Aff(d) are not necessarily hyperplanes unless $b = b'$. However, they are polynomials of degree two in the entries of M and t. Indeed:

$$\mathcal{B}(b, b'; a, a') = \{(M, t) \in \mathbb{R}^{d(d+1)} : \|Mb'\|^2 - \|Mb\|^2 + 2\left\langle Mb + t, a' - a\right\rangle + \|a'\|^2 - \|a\|^2 = 0\}.$$

Hence, Theorem 12.8 applies with $m = \binom{k}{2}\binom{n}{2}$ and $D = 2$ for linear subspaces and with $D = 4$ for isometries and rigid motions. The claimed bounds follow. □

12.2 Other cost functions

A second direction in which some of the results from the previous chapters can be generalized is by changing the cost of the matchings. As far as the cost function is the sum of monotone functions in the Euclidean distances, we can restrict the search for optimal matchings to efficient ones. For instance, if one considers the minimum Euclidean bipartite matching (that is, the matching minimizing the sum of the Euclidean distances between matched points), the bound on the number of necessary labels of Theorem 11.8 still holds true, although the diagram is not necessarily polyhedral. We make this claim precise in the following statement, after giving a more general definition of Voronoi diagram.

Definition 12.12. Let $A, B \subset \mathbb{R}^d$ be a pair of finite point sets with $|B| \leq |A|$ and let $\mathcal{C}$ be a *cost function* assigning to every matching $\sigma : B \hookrightarrow A$ the cost $\mathcal{C}(\sigma) = g_\sigma : \mathbb{R}^d \to \mathbb{R}$. A matching π is $\mathcal{C}$*-optimal* for $t \in \mathbb{R}^d$ if it minimizes $\mathcal{C}(\sigma)(t)$ among all matchings σ. A $\mathcal{C}$*-Voronoi diagram* for A and B is a partition $\mathcal{V}$ of $\mathbb{R}^d$ such that for every set $S \in \mathcal{V}$ there is a matching that minimizes $\mathcal{C}(\sigma)(t)$ for all $t \in S$ and among all matchings σ. A function assigning to each set of $\mathcal{V}$ such a matching (called a *label* for the set) is called a *labeling*. The *size* of a labeling is the cardinality of its image set.

Examples of Euclidean, logarithmic and exponential matching Voronoi diagrams or, more precisely, $\mathcal{C}$-Voronoi diagrams with

$$\mathcal{C}(\sigma) = \sum_{b \in B} \|b + t - \sigma(b)\|, \ \mathcal{C}(\sigma) = \sum_{b \in B} \ln(\|b + t - \sigma(b)\| + 1) \text{ and } \mathcal{C}(\sigma) = \sum_{b \in B} \exp(\|b + t - \sigma(b)\|/2),$$

are shown in Figure 12.1(a), Figure 12.1(b) and Figure 12.1(c), respectively. Each color represents a label (that is, a matching that is $\mathcal{C}$-optimal in all the regions colored with the corresponding color).

Proposition 12.13. *Let $A, B \subset \mathbb{R}^d$ be a pair of point sets with $k = |B| \leq |A| = n$, and let $\mathcal{C}$ be a cost function of the form*

$$\mathcal{C}(\sigma) = g_\sigma : t \mapsto \sum_{b \in B} f_b(\|b + t - \sigma(b)\|), \ \text{with } f_b : \mathbb{R} \to \mathbb{R} \text{ monotone for all } b \in B.$$

There is a $\mathcal{C}$-Voronoi diagram for A and B with a labeling of size $O\left(n^{2d}k^d k!\right)$.

Note that we allow the functions f_b to be monotonically decreasing in the formulation of the previous proposition, embracing then scenarios in which we want some points in B to be far away from its matched point in A.

Proof. Let $\mathcal{P}(\Gamma, A, B)$ be the arrangement of pullbacks to the space of translations $\Gamma \cong \mathbb{R}^d$, which has complexity $O(n^{2d}k^d)$ as argued in the previous chapter. Observe that for a cell C of $\mathcal{P}(\Gamma, A, B)$, the A-preference is fixed. Let $\mathcal{V}$ be a $\mathcal{C}$-Voronoi diagram for A and B. Consider the restriction of $\mathcal{V}$ to a fixed cell of $C \in \mathit{cells}(\mathcal{P}(\Gamma, A, B))$ and let π be a label for a cell in this restricted diagram. Let $\mathcal{P}_C$ be the (k, n)-preference, as defined in Chapter 9, resulting of reversing in the A-preference of C the permutations of every $b \in B$ such that f_b is decreasing. If π is not an efficient matching for the $\mathcal{P}_C$, then a better matching can be used as a label. Such a matching will have cost not larger than π in the whole cell, because f_b is monotone for all $b \in B$. Thus, iterating this relabeling process, we can find a set of labels that are efficient for the A-preference in C, which are bounded in number by $k!$, as a consequence of Theorem 9.4. □

A natural option to construct a cost function $\mathcal{C}$ is to give a weight to every point in A, which will be added to the squared Euclidean lengths of the edges using it. As far as these weights are constant (in fact, linear is enough), the diagram remains polyhedral and all the presented results hold for least-squares, bottleneck and lex-bottleneck matchings. Even the invariance with respect to translation for equally sized sets stated in Lemma 11.3 generalizes to this case.

Proposition 12.14. *Let $A, B \subset \mathbb{R}^d$ be a pair of point sets with $k = |B| \leq |A| = n$, and let $\mathcal{C}$ be a cost function of the form*

$$\mathcal{C}(\sigma) = g_\sigma : t \mapsto \sum_{b \in B} \left(\|b + t - \sigma(b)\|^2 + w(\sigma(b)) \right), \text{ with } w(a) \in \mathbb{R} \text{ for all } a \in A.$$

There is a $\mathcal{C}$-Voronoi diagram that is a polyhedral complex of combinatorial complexity

$$O(n^{2d + \lfloor d/2 \rfloor} k^{d + \lfloor d/2 \rfloor - 1/2} (e \ln k + e)^k).$$

Proof. Consider, as in Lemma 12.5, the set

$$V_\pi = \{(M, t) \in \mathbb{R}^{d(d+1)} : \pi \text{ is } \mathcal{C}\text{-optimal for } f[M, t](B) \text{ and } A\}.$$

Note that $g_\pi(t) \leq g_\sigma(t) =$ if and only if

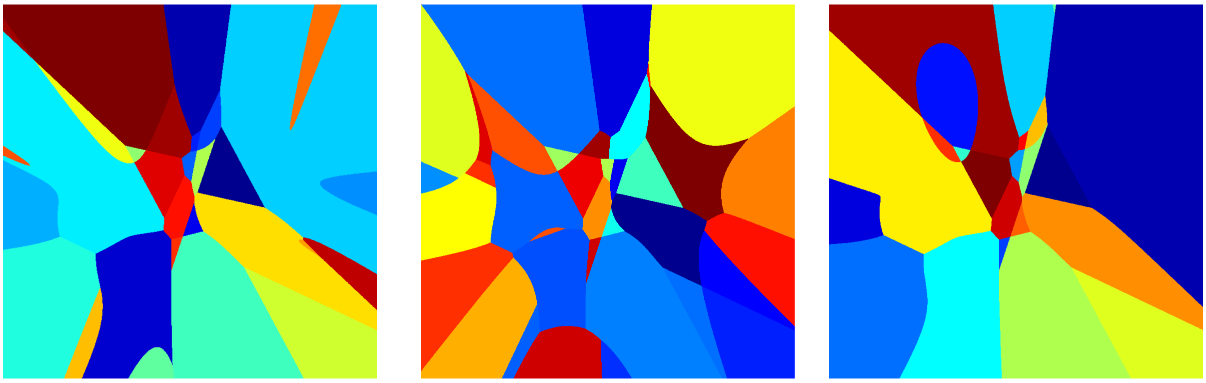
$$\sum_{b \in B} \left(\|b + t - \pi(b)\|^2 + w(\pi(b)) \right) \leq \sum_{b \in B} \left(\|b + t - \sigma(b)\|^2 + w(\sigma(b)) \right),$$

which holds in the halfspace given by the inequality

$$2 \left\langle t, \sum_{b \in B} (b - \pi(b)) \right\rangle + \sum_{b \in B} \|b - \pi(b)\|^2 + \sum_{b \in B} w(\pi(b)) \leq$$
$$\leq 2 \left\langle t, \sum_{b \in B} (b - \sigma(b)) \right\rangle + \sum_{b \in B} \|b - \sigma(b)\|^2 + \sum_{b \in B} w(\sigma(b)).$$

Thus, V_π is a polyhedron. In addition, in view of the previous expression, matchings having the same image set have costs differing by a constant and, hence, there is one of them which is better (or not worse) than the others for all the values of $t \in \mathbb{R}^d$. The proof of Proposition 11.11 carries over as well. Therefore, the arguments from Corollary 11.13 apply in this situation and yield the stated bounds. □

Observe that the analogous to Theorem 11.5 holds in this scenario as well. One could also change the underlying metric taking, for instance, the L_1 or L_∞ metrics to measure the length of the edges. Then, the number of optimal matchings in a cell of the arrangement of bisectors



(a) Euclidean diagram (b) Logarithmic diagram (c) Exponential diagram

Figure 12.1: $\mathcal{C}$-Voronoi diagrams for the point sets in Figure 10.1(a)

would be still bounded by the number of efficient matchings of a (k, n)-preference. However, the bound for the number of such regions should be adapted and additional "degeneracies" (as bisectors having non-empty interior) should be handled.

The bottleneck and lex-bottleneck diagrams admit similar generalizations. We state next an example whose proof can be deduced *mutatis mutandis* from the proof of Proposition 12.14.

Proposition 12.15. *Let $A, B \subset \mathbb{R}^d$ be a pair of point sets with $k = |B| \le |A| = n$, and let $\mathcal{C}$ be a cost function of the form*

$$\mathcal{C}(\sigma) = g_\sigma : t \mapsto \max_{b \in B} \left\{ \|b + t - \sigma(b)\|^2 + w(\sigma(b)) \right\},$$

with $w(a) \in \mathbb{R}$ for all $a \in A$. There is a $\mathcal{C}$-Voronoi diagram that is a polyhedral complex of combinatorial complexity $O(n^{2d}k^{2d})$.

12.3 Matchings on multisets

We explore in this section yet another direction of generalization where we allow repetitions in the input point sets. That is, considering the point sets as multisets of points. This could be interesting if one wants to approximate certain clusters of points by a single one. In addition, it can be interpreted as a special case of the Earth Mover's distance where the weights of the points are positive integers.

Proposition 12.16. *Let $A, B \subset \mathbb{R}^d$ be multisets consisting respectively of l and m (distinct) points. Let $k_1, \ldots, k_m$ be the multiplicities of the points in B and let $k = \sum k_j$. There is a least-squares diagram of A and B that is a polyhedral complex of combinatorial complexity*

$$O\left(l^{2d+\lfloor d/2\rfloor} m^{d+\lfloor d/2\rfloor} \min\left\{(e \ln k + e)^k/\sqrt{k}, k!/(k_1! \cdots k_m!)\right\}\right).$$

Proof. Note first that the complexity of the diagram of A-preferences depends now only on the number of distinct points. For a fixed A-preference, we count the number of efficient matchings, regarding the repeated points of A and B as distinct points (breaking the ties between points of A in an arbitrary but consistent way). This can only lead to overcounting the number of efficient matchings that we need to label the regions having this preference. On one hand, we have that for a fixed image set, there is only one efficient matching, since the cost of a matching is still a linear function whose gradient depends only on the matched set. On the other hand, we know that the number of efficient matchings is bounded by the number of greedy matchings, as seen in Theorem 9.4. Every greedy matching though, can be obtained at least in $k_1! \cdots k_m!$ ways, by permuting the positions of the instances of every distinct point. The Upper-bound Theorem [83] finishes the proof as in Corollary 11.13, since the number of facets of each cell is bounded, using the arguments in Proposition 11.11, by lm. □

This generalization allows us to point out a nice limit property of the least-squares diagram. Consider the extreme case in which the points of A are all distinct and B consists of k instances of the same point. The order-k Voronoi diagram of A is then a least-squares diagram for the multisets A and B. The previous theorem gives a bound of $O(n^{2d+\lfloor d/2 \rfloor})$ for the least-squares diagram, which is loose because the complexity of the order-k Voronoi diagram of n points is

$$O\left(k^{\lceil (d+1)/2 \rceil} n^{\lfloor (d+1)/2 \rfloor}\right),$$

as follows from [38]. Thus, although least-squares diagrams for multisets comprise all higher-order Voronoi diagrams, they are in general more complex, as Proposition 11.10 implies.

13
Related problems

In this chapter, we explore some of the applications of the Voronoi diagrams studied in this part. For simplicity of presentation, we restrict the study to the plane and mainly to bottleneck diagrams. Similar results can be derived for the least-squares or lex-bottleneck cost functions, adapting the running times. An obvious application is to report a set of matchings containing at least one optimal matching for every translation (that is, a set that can be used as a labeling). A partial-matching Voronoi diagram can also serve as data structure for interactively showing the best matching for each position of a cursor or, similarly, as a preprocessing for a later sequence of position queries of a pattern.

13.1 Optimal matching under translations

Let $A, B \subset \mathbb{R}^2$ be two point sets with $k = |B| \leq |A| = n$. We are interested here in finding a matching σ such that

$$f_\sigma(t^*) = \min_{t \in \mathbb{R}^2} \mathcal{E}(t), \text{ where } t^* \in \mathbb{R}^2,$$

using the notation from Definition 10.1. Such a matching will be called an *optimal matching under translations* and such a translation is indeed a minimum of the cost function $\mathcal{E}$. It is often interesting to know a translation for which this minimum is attained. The algorithm we present can return this information as well. Note that if we have the polyhedral complex $\mathcal{L}(A, B)$ and one bottleneck labeling for it, we only need to traverse the diagram and find the optimal value of the function indicated by the labeling in every convex polygon separately.

Theorem 13.1. *Let $A, B \subset \mathbb{R}^2$ be sets of k and n points. An optimal bottleneck matching for A and B under translations can be found in $O(n^2k^8)$ time.*

Proof. We first construct $\mathcal{L}(A, B)$ and a labeling for it as described in the proof of Theorem 10.20. We traverse then the arrangement optimizing for t the value $f(t) = \|b + t - a\|^2$ in every (convex) cell, and keeping the minimum throughout the diagram, where ab is the longest edge of the bottleneck matching given by the label of the cell. Let C be one cell in $\mathcal{L}(A, B)$ and let $t_0 = a - b$ be the global minimum of the function $f(t)$. If $t_0 \in C$, obviously $f(t_0) = 0$ is the minimum of f in C (in fact, a global minimum as well). Otherwise, the minimum is attained at the point of C closest to t_0. Such a point must be either a vertex of C or the orthogonal projection of t_0 in an edge of C (that is, the projection on the line supporting e if the projection lies in e and the empty set otherwise). In addition, if t_0 is the minimum of $f(t)$ and t_1 is the minimum of the corresponding function for a neighboring cell D sharing the edge e with C, then the projection of t_0 in e coincides with the projection of t_1 in e. We can thus calculate the minimum examining once every vertex of the diagram and at most one candidate point for every edge. Thus, the total time needed to perform the mentioned optimization in every cell is proportional to the complexity of the diagram. □

The problem addressed in this section was introduced in [9] for the balanced case, where an algorithm running in $O(n^6 \log n)$ is given. We can however adapt their algorithm and analyze its complexity in the unbalanced case as well. Their approach consists of two steps. First, they define $\varepsilon(b_1, a_1, b_2, a_2, b_3, a_3)$ for every pair of triples $a_1, a_1, a_3 \in A$, $b_1, b_2, b_3 \in B$ to be the minimum $\varepsilon \in \mathbb{R}$ such that there is a translation placing b_i into an ε-neighborhood of a_i for all $i \in [3]$. They claim that the bottleneck distance under translations must be one of such $O(n^6)$ values, and they compute every value in constant time. In the unbalance case, the number of values and the time for its computation is then $O(n^3k^3)$. They sort these values into an array $\mathfrak{E}$ and perform then a binary search, testing for every $\varepsilon \in \mathfrak{E}$ whether there is a bottleneck matching under translations having cost ε. In order to do that, they assume that $\|b+t-a\|^2 = \varepsilon^2$ for a fixed pair $ab \in A \times B$, which restricts the set of candidate translations to a circle. They parametrize the circle by polar coordinates and compute the set of angles $\alpha(a', b')$ for which b' lies in an ε-neighborhood of a', for all $a'b' \in A \times B$. The computation of such intervals is not trivial and requires some careful observations (which carry over to the unbalanced case). They compute the arrangement in this circle induce by the circular intervals α (sorting their endpoints). In every interval, they construct the bipartite graph whose edges are shorter than ε and look for a maximum matching in it. This is done computing the graph for an arbitrary initial point and traversing the arrangement computed on the circle, adding or deleting at each interval the corresponding edge (or edges). If edges are only added, nothing is left to be done. If some edges are delete leaving some points of B unmatched, one (or several) augmenting-path computations must be performed in order to decide whether there is a complete matching in the next interval. The construction of each one of the arrangements in the $O(nk)$ circles for a fixed value $\varepsilon \in \mathfrak{E}$ can be done in $O(nk \log n)$. The traversal requires $O(nk)$ to construct the initial graph, $O(nk + k^{2.5})$ to prune non-relevant edges and obtain a maximum matching for the initial graph using the Hopcroft-Karp algorithm, and $O(nk)$ time per cell to compute the augmenting path to update the maximum matching. The updates of the graph require constant time for each edge. Thus, the total time required for each $\varepsilon \in \mathfrak{E}$ is $O(nk \cdot (nk \log n + nk + k^{2.5} + nk \cdot nk)) = O(n^3k^3)$. Since sorting the values in $\mathfrak{E}$ is done in $O(n^3k^3 \log n)$ time, the whole algorithm runs in time $O(n^3k^3 \log n)$. Hence, it outperforms our algorithm only if $k = \Omega((n \log n)^{1/5})$.

The algorithm for the balanced case described above was later improved in [52] to $O(n^5 \log^2 n)$ time. This improvement requires the use of a non-trivial data structure and parametric search combined with sorting networks. The data structure is used to create an oracle that, given $\varepsilon \in \mathfrak{E}$, answers whether the bottleneck distance of A and B under translations is at most ε. The equivalent oracle in [9] runs in $O(n^6)$ and, according to the analysis above, can be adapted to run in $O(n^3k^3)$ in the unbalanced case. In [52], the oracle runs in $O(n^5 \log n)$ thanks to the gain in the computation of augmenting paths, which is done in $O(n \log n)$ time. In the unbalanced case, the augmenting path can be found in $O(k \log n)$ time, as can be derived adapting their analysis. Therefore, the oracle runs in $O(n^2k^3 \log n)$ time. However, the time to sort the $O(n^3k^3)$ values into $\mathfrak{E}$ dominates the running time preventing the new algorithm to improve the simplest algorithm in [9]. This problem is solved by using Cole's parametric search technique to avoid the construction of $\mathfrak{E}$, and reducing the number of oracle calls to $O(\log n)$. The final running time is hence $O(n^2k^3 \log^2 n)$, which beats the algorithm proposed in Theorem 13.1 if $k = \Omega(\log^{2/5} n)$.

It is worth mentioning that a traversal like the one in Theorem 13.1 of the least-squares diagram results into the best known algorithm to find the optimal least-squares matching under translations, which is cubic in n for constant k.

Theorem 13.2. *Let $A, B \subset \mathbb{R}^2$ be sets of k and n points. An optimal least-squares matching for A and B under translations can be found in $O(n^3k^{7.5}(e \ln k + e)^k)$ time.*

13.2 Computing the safest path

Here we approach the problem of finding a motion for B from an initial position to a final position such that the maximum bottleneck value (as defined in Definition 10.1) attained during the motion is minimized. We define the problem more formally below.

Definition 13.3. Let $A, B \subset \mathbb{R}^2$ be two finite point sets and $\mathcal{E}$ be the associated bottleneck value function. The *bottleneck value of a closed curve* $\gamma : [0,1] \to \mathbb{R}^2$ with respect to A and B is

$$F(\gamma) = \max_{s \in [0,1]} \mathcal{E}(\gamma(s)).$$

A curve γ is a *safest path* from $t_0 = \gamma(0)$ to $t_1 = \gamma(1)$ with respect to A and B if it minimizes $F(\varphi)$ among all curves $\varphi : [0,1] \to \mathbb{R}^2$ with $\varphi(0) = t_0$ and $\varphi(1) = t_1$.

Such a path can be useful in motion planning if the points of A represent antennas and the points of B represent receivers of a moving robot. A related problem is to decide whether a robot, whose arms can reach any point at distance at most r from its basis, can climb a wall with given anchor points from an initial to a target given position. The notion of safest path could be of interest also for recovering the trajectory of an object. For instance, if we have got a sensor network and the points of A correspond to the sensors that detected one of the emitters of a moving object represented by B.

We next construct a graph that will contain all the necessary information to compute a safest path from any initial position to any final position.

Definition 13.4. The *bottleneck graph* of two finite point sets $A, B \subset \mathbb{R}^2$ is the graph $\mathcal{L}(A,B)^*$ dual to $\mathcal{L}(A,B)$ where an edge e^* of the graph dual to an edge e of $\mathcal{L}(A,B)$ has weight $\min\limits_{t \in e} \mathcal{E}(t)$.

The following technical lemma will be crucial in the proof of the main result of this section.

Lemma 13.5. *Let $A, B \subset \mathbb{R}^2$ be two finite point sets. Let $t_0, t_1 \in \mathbb{R}^2$ and $\delta \in \mathbb{R}$ be given and let C_0 and C_1 be cells of $\mathcal{L}(A,B)$ such that $t_0 \in C_0$ and $t_1 \in C_1$. There is a curve with bottleneck value at most δ from t_0 to t_1 with respect to A and B if and only if $\mathcal{E}(t_0), \mathcal{E}(t_1) \leq \delta$ and there is a path from C_0^* to C_1^* in $\mathcal{L}(A,B)^*$ whose longest edge has weight not larger than δ.*

Proof. In every cell of $\mathcal{L}(A,B)$ there is a bottleneck matching whose cost coincides with $\mathcal{E}$ in the cell. By definition, $\mathcal{E}$ is convex in every cell of $\mathcal{L}(A,B)$. Hence, if $C_0 = C_1$ the straight line segment joining t_0 and t_1 has bottleneck value $\max\{\mathcal{E}(t_0), \mathcal{E}(t_1)\}$ and no path can attain a smaller value. If $C_0 \neq C_1$ and γ is any path from t_0 to t_1, we can replace each of the arcs of γ entering a cell C of $\mathcal{L}(A,B)$ in a point t_{in} and leaving it in a point t_{out} by the straight line segment joining t_{in} with t_{out} without increasing the bottleneck value of the path. Again, we can only decrease the bottleneck value of the path if we substitute this line segment by the one joining the points $\hat{t}_{\text{in}}$ and $\hat{t}_{\text{out}}$ where the first is the point having minimum bottleneck value from the edge of t_{in} and the second is the one attaining the minimum value in the edge of t_{out}. The path in C_0 can be perturbed to be the straight line segment from t_0 to the point attaining the minimum of $\mathcal{E}$ on whichever edge the path crosses first. Similarly, the path in C_1 can be transformed into a line segment. Putting the previous observations together, it follows that at least one safest path must be among the polygonal paths having vertices on the minimum value attained at edges. The bottleneck value of such a path is the maximum between the maximum weight of the edges forming the corresponding path in the bottleneck graph, the value of the matching at t_0 and the value at t_1. □

Theorem 13.6. *Let $A, B \subset \mathbb{R}^2$ be two finite point sets. Given $t_0, t_1 \in \mathbb{R}^2$, a safest path from t_0 to t_1 with respect to A and B can be computed in $O(n^2k^6(k^2 + \log n))$ time.*

Proof. We first compute $\mathcal{L}(A, B)$ in the time indicated in Theorem 10.20. If σ is a bottleneck matching along an edge e, the weight of the edge e^* of $\mathcal{L}(A, B)^*$ is the minimum among the value of σ at the endpoints of e and at the projection of the minimum of f_σ in e (as defined in the proof of Theorem 13.1). Since, in view of Theorem 10.13, the number of edges and vertices of $\mathcal{L}(A, B)^*$ is $O(n^2k^6)$, this graph with the weights of its edges can be computed in $O(n^2k^6)$ time.

Since the weights of the edges are non-negative, the path with minimum bottleneck length in the graph can be found in $O(n^2k^6 \log n)$ using the implementation of Dijkstra's algorithm using heaps. By the previous lemma, we only need to take the associated polygonal path and it will be guaranteed to be a safest path. Moreover, the value of the path can be easily computed. □

The analogous question may be asked in terms of the least-squares cost function and can be solved similarly, as in every cell of the least-squares diagram the objective function is quadratic as well. The running time should be of course adapted accordingly.

13.3 Finding the polygon cover radius

In this section, we investigate a covering problem. Given a pair of finite point sets $A, B \subset \mathbb{R}^2$, and a convex polygon $Q \subset \mathbb{R}^2$, we want to determine the minimum radius $\delta \in \mathbb{R}$ such that for any position of the point set B in Q there is a matching whose bottleneck value is at most δ. We can think of the points in A as antennas and the disks of radius δ as the region on which they provide signal. The point set B can be thought of as a robot that moves in Q and needs to connect each of its points to a different antenna (for instance, to learn its position). The target is to minimize the power consumed by the antennas while ensuring that the robot can move in Q having a one-to-one connection for its receivers. Another scenario would be that the robot has arms and the points in A represent anchor points (as in the previous application) and we want the robot to be able to reach any position in Q.

Definition 13.7. The *cover radius* of a polygon with respect to two finite point sets $A, B \subset \mathbb{R}^2$ is the maximum bottleneck value among all $t \in \mathbb{R}^2$ such that $B + t \subset Q$.

Theorem 13.8. *The cover radius of a polygon Q on m vertices with respect to $A, B \subset \mathbb{R}^2$ with $k = |B| \leq |A| = n$ can be computed in $O(n^2k^8 + (n^2k^6 + m)\log(n + m))$ time.*

Proof. We start by computing $\mathcal{L}(A, B)$ and a bottleneck labeling as indicated in Theorem 10.20. Note that the set of $t \in \mathbb{R}^2$ for which $B + t \subset Q$ is a convex polygon $\widehat{Q}$, which is the intersection of k translated copies of Q. This polygon is indeed given by the m linear inequalities obtained by imposing that the extreme point of B in the direction orthogonal to an edge of Q is in the right side of the corresponding edge. This polytope can be computed easily in $O(k \log k + m \log m)$ time and has at most m edges. Then, we compute the overlay of the boundary of $\widehat{Q}$ and $\mathcal{L}(A, B)$. Note that every edge of $\mathcal{L}(A, B)$ can intersect the boundary of $\widehat{Q}$ in at most two points. Thus, the number of vertices of the overlay is $O(n^2k^6 + m)$ and, hence, it can be computed in $O((n^2k^6 + m)\log(n + m))$ time using the techniques appearing in [44]. The next step involves traversing the overlay and taking the maximum attained by $\mathcal{E}$ in every cell. Since the function is convex, the maximum can be calculated as the maximum of the values attained at the vertices of overlay. □

14 Summary and conclusion

We studied the combinatorial complexity of a collection of tessellations called partial-matching Voronoi diagrams. We focused on the least-squares and the bottleneck versions of them, together with the lexicographic bottleneck variant. We obtained algorithms for its construction that are sensitive to the size of the smaller set (the pattern). In particular, these algorithms can be used to find the best occurrence of a pattern in a cloud of points. In addition, we derived structural properties and conditions under which they are canonical. We also provided a more general framework to study different transformation spaces and cost functions. Finally, we have explored several related problems that can be solved with the help of our main results.

Specifically, we provided an algorithm to construct the bottleneck partial-matching Voronoi diagram of two point sets of sizes k and n in the d-dimensional Euclidean space in $O(n^{2d}k^{2d+2})$ time, and the lexicographic variant in $O(n^{2d}k^{2d+4})$. In the plane, the diagrams can be constructed in $O(n^2k^8)$ and $O(n^2k^{10})$ time, respectively. As a consequence, we can compute the bottleneck distance under translations in the plane in $O(n^2k^8)$ time. We prove that the algorithm constructing the lex-bottleneck diagram is optimal for constant size patterns by deriving a lower bound of $\Omega(n^2k^2)$ on its complexity.

We proved a first non-trivial bound of $O(n^{2d+\lfloor d/2 \rfloor}k^{d+\lfloor d/2 \rfloor -1/2}\rho(k)^k)$ on the complexity of the least-squares partial-matching Voronoi diagram, where $\rho(k) = e \ln k + e$. An improved bound of $O(n^2k^{3.5}\rho(k)^k)$ in the plane is shown to be optimal for constant k. We designed an algorithm to find the least-squares optimal matching under translations in the plane spending $O(nk^4)$ time per cell. When the pattern is translated only along a line we can compute the optimal matching in $O(nk(\log n + k^3))$ constructing the intersection of the least-squares diagram with the line. Still, it remains open to know if the complexity of the partial-matching Voronoi diagram (or of the problem of finding the optimal value under translations) is polynomial in the pattern size.

Generalizing our methods, we proved that the least-squares diagram under affinities is polyhedral as well, and we bounded its complexity. We derived also bounds for non-polyhedral diagrams, such as the Euclidean diagram under translations or the bottleneck diagram under rigid motions.

A

A subdivision whose regularity tree has two levels

We prove here that the subdivision $\mathcal{S}$ pictured in Figure 4.1 (left) is not regular and that its finest regular coarsening is the subdivision $\mathcal{S}_0$ defined by the second level of the tree in Figure 4.1 (right). To this end, we provide a solution to the dual of its regularity system in the sense of Lemma 4.5.

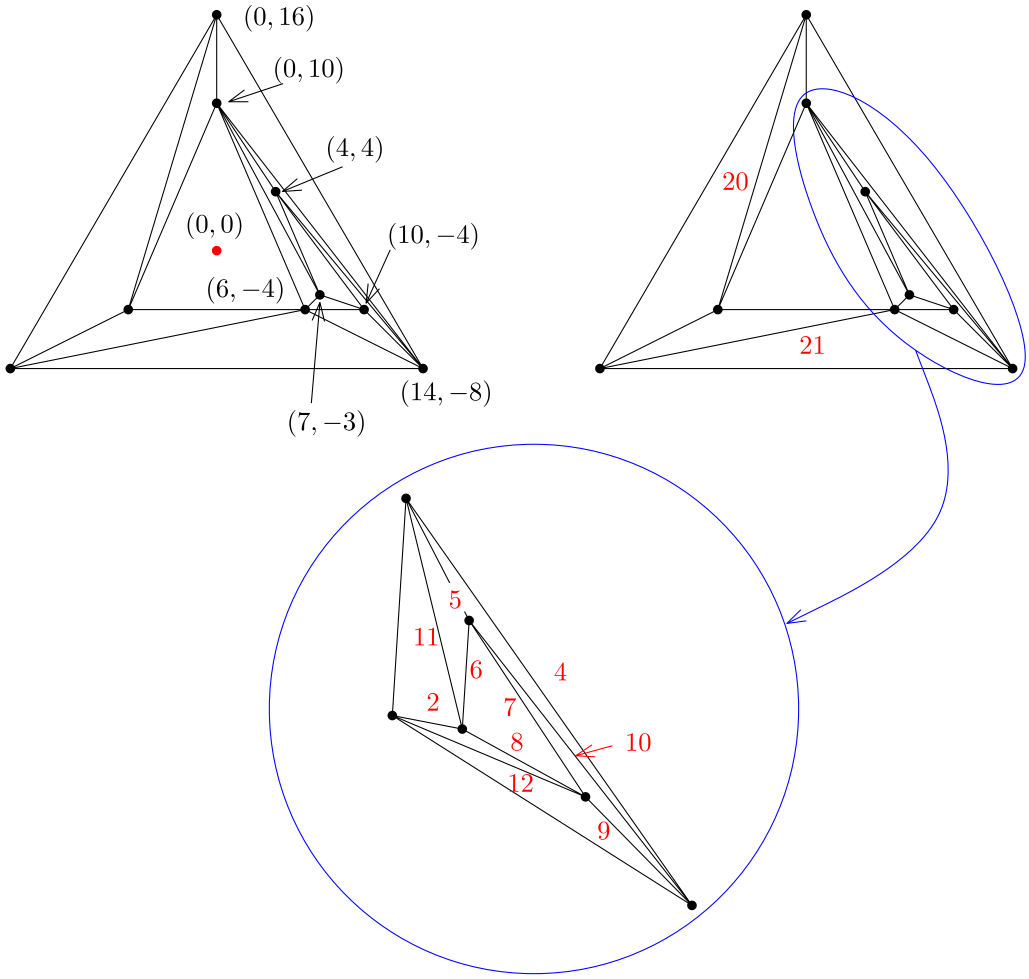


Figure A.1: A non-regular, recursively regular subdivision.

The rows of the matrix of the regularity system of $\mathcal{S}$ associated to the edges labeled with numbers in Figure A.1 are

$$\begin{aligned}
s_2 &= (-56, 20, 0, 4, 32, 0, 0, 0, 0)\\
s_4 &= (0, 0, 84, -72, 0, -24, 0, 0, 12)\\
s_5 &= (12, 0, -56, 34, 0, 10, 0, 0, 0)\\
s_6 &= (4, 10, -32, 18, 0, 0, 0, 0, 0)\\
s_7 &= (8, -34, 8, 0, 0, 18, 0, 0, 0)\\
s_8 &= (-32, 10, 4, 0, 18, 0, 0, 0, 0)\\
s_9 &= (0, 56, 16, 0, 8, 32, 0, 0, 0)\\
s_{10} &= (0, 12, -16, 8, 0, -4, 0, 0, 0)\\
s_{11} &= (-20, 0, 20, -10, 10, 0, 0, 0, 0)\\
s_{12} &= (16, -12, 0, -8, 4, 0, 0, 0, 0)\\
s_{20} &= (0, 0, 0, 136, 0, 0, 36, -84, -88)\\
s_{21} &= (0, 0, 0, 0, -112, 48, -48, 112, 0).
\end{aligned}$$

The following positive coefficients for the dual variables (and zero for the omitted variables) are a solution to the dual system

$$y_4 = y_5 = y_6 = y_{10} = y_{12} = 1, y_7 = \frac{50}{99}, y_8 = \frac{71}{99}, y_{11} = \frac{36}{55}, y_{20} = \frac{3}{22}, y_{21} = \frac{9}{88},$$

since it is easy to check that

$$s_4 + s_5 + s_6 + s_{10} + s_{12} + \frac{50}{99}s_7 + \frac{71}{99}s_8 + \frac{36}{55}s_{11} + \frac{3}{22}s_{20} + \frac{9}{88}y_{21} = 0.$$

As discussed in Theorem 4.8 and its algorithmic translation described in Proposition 4.18, to show that the finest regular coarsening of $\mathcal{S}$ is $\mathcal{S}_0$ it is enough to show that the variables y_2 and y_9 can attain positive values in the dual (if we allow the variables $y_4, y_5, y_6, y_7, y_8, y_{10}, y_{11}, y_{12}, y_{20}, y_{21}$ to take any real value). We certify that it is indeed the case by means of the solution having as non-zero coefficients:

$$y_2 = y_4 = y_9 = y_{10} = 1, y_{12} = -2, y_5 = -\frac{1}{11}, y_{11} = -\frac{49}{11}, y_{20} = \frac{3}{22}, y_{21} = \frac{9}{88}.$$

The sub-subdivision of $\mathcal{S}$ pictured (after an affine transformation) in the lower part of Figure A.1 is a variant of a typical non-regular subdivision called “the mother of all examples” in [45]. However, since the lines supporting the edges 2, 5 and 9 are concurrent, the sub-subdivision is recursively regular (is the projection of a truncated pyramid). Similarly, it is clear that $\mathcal{S}_0$ is regular as well.

B

A non-regular recursively-regular subdivision

We prove here that the subdivision $\mathcal{S}$ pictured in Figure 4.2(a) is not regular. To this end, we provide a solution to the dual of its regularity system in the sense of Lemma 4.5. The vertices of the subdivision lie symmetrically with respect to the coordinate axes.

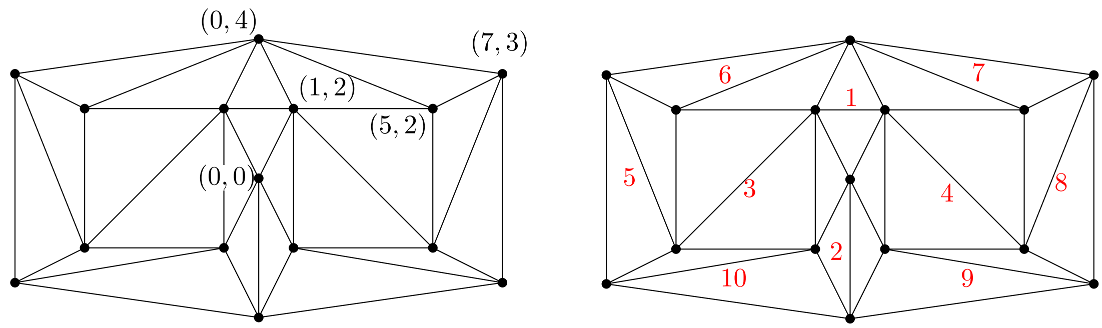


Figure B.1: A non-regular, recursively regular subdivision.

The rows of the matrix of the regularity system of $\mathcal{S}$ associated to the edges labeled in Figure B.1 are

$$
\begin{aligned}
s_1 &= (0, 0, -4, 0, 4, -4, 0, 0, 0, 4, 0, 0, 0, 0, 0)\\
s_2 &= (0, 0, 0, 4, -4, 0, 4, 0, 0, 0, -4, 0, 0, 0, 0)\\
s_3 &= (-16, 16, -16, 16, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)\\
s_4 &= (0, 0, 0, 0, 0, -16, 16, -16, 16, 0, 0, 0, 0, 0, 0)\\
s_5 &= (-12, 12, 0, 0, 0, 0, 0, 0, 0, 0, 0, 8, 0, 0, -8)\\
s_6 &= (0, -13, 9, 0, 0, 0, 0, 0, 0, -4, 0, 0, 0, 0, 8)\\
s_7 &= (0, 0, 0, 0, 0, 9, 0, 0, -13, -4, 0, 0, 0, 8, 0)\\
s_8 &= (0, 0, 0, 0, 0, 0, 0, -12, 12, 0, 0, 0, 8, -8, 0)\\
s_9 &= (0, 0, 0, 0, 0, 0, 0, -9, 13, 0, 0, 4, 0, -8, 0)\\
s_{10} &= (13, 0, 0, -9, 0, 0, 0, 0, 0, 0, 4, -8, 0, 0, 0).
\end{aligned}
$$

The following positive coefficients for the dual variables (and zero for the omitted variables) are a solution to the dual system

$$y_1 = y_2 = 1, y_3 = y_4 = \frac{1}{32}, y_5 = y_6 = y_7 = y_8 = y_9 = y_{10} = \frac{1}{2},$$

since it is easy to check that

$$s_1 + s_2 + \frac{1}{32}(s_3 + s_4) + \frac{1}{2}(s_5 + s_6 + s_7 + s_8 + s_9 + s_{10}) = 0.$$

The coarsening $\mathcal{S}'$ of $\mathcal{S}$ in Figure B.2 is regular, because the depicted heights represent a convex lifting. The subdivisions $\mathcal{S}$ is then recursively-regular because it can be obtained refining with regular subdivisions the cells of $\mathcal{S}'$.

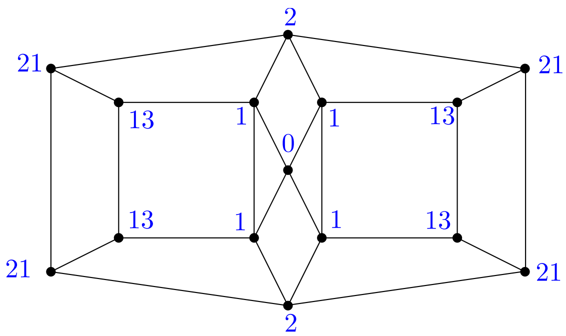


Figure B.2: A regular coarsening of $\mathcal{S}$.

C

A non-recursively-regular acyclic triangulation

We prove here that the triangulation $\mathcal{S}$ pictured in Figure 4.2(a) is not recursively-regular. To this end, we prove that its finest regular coarsening has only one cell providing a solution to the dual of its regularity system in the sense of Lemma 4.5.

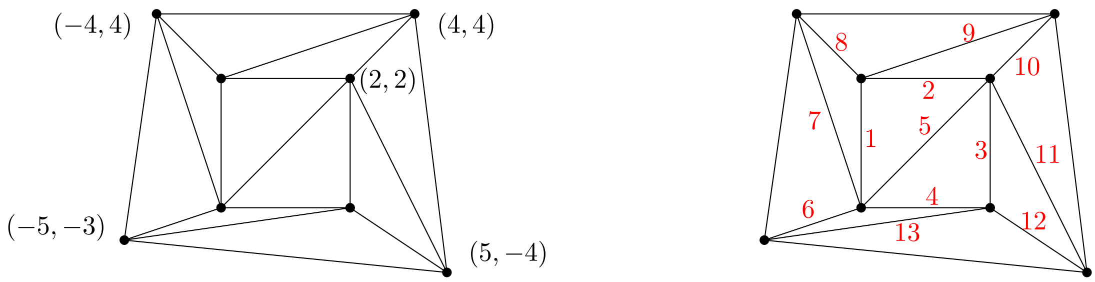


Figure C.1: A non-recursively-regular triangulation.

The rows of the matrix of the regularity system of $\mathcal{S}$ associated to the edges labeled in Figure C.1 are

$$
\begin{aligned}
s_1 &= (8, -32, 8, 0, 16, 0, 0, 0)\\
s_2 &= (8, 0, -24, 0, 0, 16, 0, 0)\\
s_3 &= (12, 0, 8, -36, 0, 0, 16, 0)\\
s_4 &= (-28, 0, 4, 8, 0, 0, 0, 16)\\
s_5 &= (-16, 16, -16, 16, 0, 0, 0, 0)\\
s_6 &= (-48, 0, 0, 20, 4, 0, 0, 24)\\
s_7 &= (-16, 20, 0, 0, -12, 0, 0, 8)\\
s_8 &= (16, -48, 0, 0, 24, 8, 0, 0)\\
s_9 &= (0, -16, 16, 0, 8, -8, 0, 0)\\
s_{10} &= (0, 18, -50, 0, 0, 24, 8, 0)\\
s_{11} &= (0, 0, -22, 18, 0, 12, -8, 0)\\
s_{12} &= (0, 0, 17, -57, 0, 0, 28, 12)\\
s_{13} &= (17, 0, 0, -13, 0, 0, 4, -8).
\end{aligned}
$$

The following positive coefficients for the dual variables (and zero for the omitted variables) are a solution to the dual system

$$y_1 = y_5 = \frac{1}{4}, y_3 = \frac{1}{12}, y_7 = y_9 = y_{13} = 1, y_{11} = \frac{2}{3},$$

since it is easy to check that

$$\frac{1}{4}(s_1+s_5)+\frac{1}{12}s_3+s_7+s_9+s_{13}+\frac{2}{3}s_{11}=0.$$

Following the algorithm in the proof of Corollary 4.19, we allow the variables appearing in the previous equation to take negative values and observe that

$$-\frac{51}{8}s_1+s_2-\frac{25}{6}s_3+s_4-3s_5+s_6-\frac{11}{2}s_7+s_8+s_9+s_{10}-\frac{10}{3}s_{11}+s_{12}+s_{13}=0.$$

Hence, the finest regular coarsening of $\mathcal{S}$ is the trivial one, having one single cell and thus $\mathcal{S}$ is not recursively regular.

We prove now that $\mathcal{S}$ has no cycle in visibility. We show first that the coarsening $\mathcal{S}'$ of $\mathcal{S}$ depicted in Figure C.2 is acyclic. Assume $x \in \mathbb{R}^2$ to be a point interior to a cell C_x of $\mathcal{S}'$ from which $\mathcal{S}'$ might be cyclic. Consider the fan $\mathcal{F} \subset \mathbb{R}^3$ obtained by taking the cone with the origin for all the cells of $\mathcal{S}'$, when $\mathcal{S}'$ is embedded in the horizontal plane of intercept $-1/8$ with the vertical axis. The construction is similar to the fan studied in Appendix D. In particular, both subdivisions have the same dual graph (illustrated in Figure D.2).

We will prove that the in-front relation from x in the plane for $\mathcal{S}'$ is equivalent to the in-front relation in the direction $(-x, 1/8) \in \mathbb{R}^3$ for $\mathcal{F}$, which is shown to be acyclic in Appendix D. Therefore, it will be proven that $\mathcal{S}'$ is acyclic as well. Take a line ℓ in the plane through x and consider points $y, z \in \ell$ such that y is interior to C_y and z is interior to C_z with $C_y, C_z \in \mathit{cells}(\mathcal{S}')$ sharing the wall $W = C_y \cap C_z$ of $\mathcal{S}'$. Assume further that y is before z in the visibility relation from x. Let $\Pi \subset \mathbb{R}^3$ be a plane through the origin and containing W in the embedded copy of $\mathcal{S}'$ (and, hence, supporting the wall of $\mathcal{F}$ associated to W). Then, the plane Π separates λy from μz for any $\lambda, \mu \in \mathbb{R}^+$. In particular, it separates $\lambda(\mu)y$ from μz if for every $\mu \in \mathbb{R}^+$ we take $\lambda(\mu)$ such that x, $\lambda(\mu)y$ and μz are coplanar. If v is a vector normal to Π and pointing "towards" x, we have then that $\langle v, \lambda(\mu)y - \mu z\rangle > 0$ for all $\mu \in \mathbb{R}^+$. Since the scalar product is continuous, we have that

$$\lim_{\mu\to 0}\langle v, \lambda(\mu)y - \mu z\rangle = \langle v, \alpha x\rangle \geq 0,$$

where $\alpha \in \mathbb{R}^+$ is a constant. Since $\langle v, x\rangle \neq 0$ because x is interior to C_x, we have that $\langle v, x\rangle > 0$.

It only remains to observe that all the sub-subdivisions refining each of the cells of $\mathcal{S}'$ into a pair of cells of $\mathcal{S}$ are obviously acyclic. By a reasoning similar to the one in the proof of Proposition 4.16, the combination of the previous observation with the fact that $\mathcal{S}'$ is acyclic already implies that $\mathcal{S}$ is acyclic.

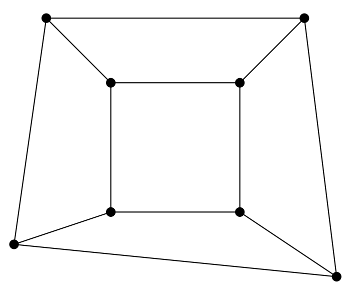


Figure C.2: An acyclic coarsening of $\mathcal{S}$.

D
A non-universal acyclic polyhedral fan

We present here the calculations for the proof of Proposition 5.15. Recall that the point set from the counterexample is

$$\begin{aligned}
p_1 =& (29, 95, 89) \\
p_2 =& (55, 19, 92) \\
p_3 =& (54, 10, 82) \\
p_4 =& (78, 2, 68) \\
p_5 =& (15, 40, 92).
\end{aligned}$$

The fan involved is drawn in Figure D.1, where the facets and cells of the fan have been labeled and it has been truncated. From the coordinates of the points in this figure and assuming that we embed it into the plane $\{z = -1/8\}$ and we cone the cells with the origin, normal vectors to the facets of the fan can be computed:

$$\begin{aligned}
v_{12} &= (4, 0, -32) \\
v_{13} &= (2, 2, 0) \\
v_{15} &= (1, -3, 16) \\
v_{23} &= (0, 4, 32) \\
v_{24} &= (4, 0, 32) \\
v_{25} &= (0, -4, 32) \\
v_{34} &= (2, -2, 0) \\
v_{45} &= (-2, -3, 8).
\end{aligned}$$

The first column of the table visits the 5! permutations representing all possible assignments from cells to points. The notation used for the permutations is simply the concatenation of the labels of the points assigned to C_1, C_2, C_3, C_4 and C_5 in this order. The second row indicates one facet for which the corresponding assignment does not satisfy the overlapping condition. The two following columns just extract the points involved in the violation. The last column computes the "gap" between the two translated floodlights. The positivity of this last value certifies that the overlapping condition is not fulfilled.

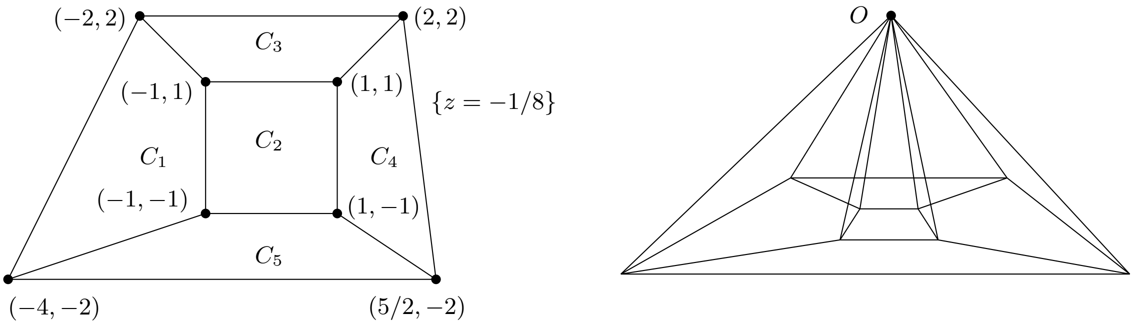


Figure D.1: The fan of the counterexample and a section of it.

σ	f_{ij}	$\sigma(C_i)$	$\sigma(C_j)$	$(\sigma(C_i) - \sigma(C_j)) \cdot v_{ij}$
54321	f_{12}	p_5	p_4	$(p_4 - p_5) \cdot v_{12} = (63, -38, -24) \cdot (4, 0, -32) = 1020$
54312	f_{12}	p_5	p_4	$(p_4 - p_5) \cdot v_{12} = (63, -38, -24) \cdot (4, 0, -32) = 1020$
54231	f_{12}	p_5	p_4	$(p_4 - p_5) \cdot v_{12} = (63, -38, -24) \cdot (4, 0, -32) = 1020$
54213	f_{12}	p_5	p_4	$(p_4 - p_5) \cdot v_{12} = (63, -38, -24) \cdot (4, 0, -32) = 1020$
54123	f_{12}	p_5	p_4	$(p_4 - p_5) \cdot v_{12} = (63, -38, -24) \cdot (4, 0, -32) = 1020$
54132	f_{12}	p_5	p_4	$(p_4 - p_5) \cdot v_{12} = (63, -38, -24) \cdot (4, 0, -32) = 1020$
53421	f_{12}	p_5	p_3	$(p_3 - p_5) \cdot v_{12} = (39, -30, -10) \cdot (4, 0, -32) = 476$
53412	f_{12}	p_5	p_3	$(p_3 - p_5) \cdot v_{12} = (39, -30, -10) \cdot (4, 0, -32) = 476$
53241	f_{12}	p_5	p_3	$(p_3 - p_5) \cdot v_{12} = (39, -30, -10) \cdot (4, 0, -32) = 476$
53214	f_{12}	p_5	p_3	$(p_3 - p_5) \cdot v_{12} = (39, -30, -10) \cdot (4, 0, -32) = 476$
53124	f_{12}	p_5	p_3	$(p_3 - p_5) \cdot v_{12} = (39, -30, -10) \cdot (4, 0, -32) = 476$
53142	f_{12}	p_5	p_3	$(p_3 - p_5) \cdot v_{12} = (39, -30, -10) \cdot (4, 0, -32) = 476$
52341	f_{12}	p_5	p_2	$(p_2 - p_5) \cdot v_{12} = (40, -21, 0) \cdot (4, 0, -32) = 160$
52314	f_{12}	p_5	p_2	$(p_2 - p_5) \cdot v_{12} = (40, -21, 0) \cdot (4, 0, -32) = 160$
52431	f_{12}	p_5	p_2	$(p_2 - p_5) \cdot v_{12} = (40, -21, 0) \cdot (4, 0, -32) = 160$
52413	f_{12}	p_5	p_2	$(p_2 - p_5) \cdot v_{12} = (40, -21, 0) \cdot (4, 0, -32) = 160$
52143	f_{12}	p_5	p_2	$(p_2 - p_5) \cdot v_{12} = (40, -21, 0) \cdot (4, 0, -32) = 160$
52134	f_{12}	p_5	p_2	$(p_2 - p_5) \cdot v_{12} = (40, -21, 0) \cdot (4, 0, -32) = 160$
51324	f_{12}	p_5	p_1	$(p_1 - p_5) \cdot v_{12} = (14, 55, -3) \cdot (4, 0, -32) = 152$
51342	f_{12}	p_5	p_1	$(p_1 - p_5) \cdot v_{12} = (14, 55, -3) \cdot (4, 0, -32) = 152$
51234	f_{12}	p_5	p_1	$(p_1 - p_5) \cdot v_{12} = (14, 55, -3) \cdot (4, 0, -32) = 152$
51243	f_{12}	p_5	p_1	$(p_1 - p_5) \cdot v_{12} = (14, 55, -3) \cdot (4, 0, -32) = 152$
51423	f_{12}	p_5	p_1	$(p_1 - p_5) \cdot v_{12} = (14, 55, -3) \cdot (4, 0, -32) = 152$
51432	f_{12}	p_5	p_1	$(p_1 - p_5) \cdot v_{12} = (14, 55, -3) \cdot (4, 0, -32) = 152$
45321	f_{24}	p_5	p_2	$(p_2 - p_5) \cdot v_{24} = (40, -21, 0) \cdot (4, 0, 32) = 160$
45312	f_{25}	p_5	p_2	$(p_2 - p_5) \cdot v_{25} = (40, -21, 0) \cdot (0, -4, 32) = 84$
45231	f_{15}	p_4	p_1	$(p_1 - p_4) \cdot v_{15} = (-49, 93, 21) \cdot (1, -3, 16) = 8$
45213	f_{15}	p_4	p_3	$(p_3 - p_4) \cdot v_{15} = (-24, 8, 14) \cdot (1, -3, 16) = 176$
45123	f_{23}	p_5	p_1	$(p_1 - p_5) \cdot v_{23} = (14, 55, -3) \cdot (0, 4, 32) = 124$
45132	f_{23}	p_5	p_1	$(p_1 - p_5) \cdot v_{23} = (14, 55, -3) \cdot (0, 4, 32) = 124$
43521	f_{23}	p_3	p_5	$(p_5 - p_3) \cdot v_{23} = (-39, 30, 10) \cdot (0, 4, 32) = 440$
43512	f_{23}	p_3	p_5	$(p_5 - p_3) \cdot v_{23} = (-39, 30, 10) \cdot (0, 4, 32) = 440$
43251	f_{23}	p_3	p_2	$(p_2 - p_3) \cdot v_{23} = (1, 9, 10) \cdot (0, 4, 32) = 356$
43215	f_{23}	p_3	p_2	$(p_2 - p_3) \cdot v_{23} = (1, 9, 10) \cdot (0, 4, 32) = 356$

Continued on next page

σ	f_{ij}	$\sigma(C_i)$	$\sigma(C_j)$	$(\sigma(C_i)-\sigma(C_j))\cdot v_{ij}$
43125	f_{23}	p_3	p_1	$(p_1-p_3)\cdot v_{23}=(-25,85,7)\cdot(0,4,32)=564$
43152	f_{23}	p_3	p_1	$(p_1-p_3)\cdot v_{23}=(-25,85,7)\cdot(0,4,32)=564$
42351	f_{15}	p_4	p_1	$(p_1-p_4)\cdot v_{15}=(-49,93,21)\cdot(1,-3,16)=8$
42315	f_{15}	p_4	p_5	$(p_5-p_4)\cdot v_{15}=(-63,38,24)\cdot(1,-3,16)=207$
42531	f_{23}	p_2	p_5	$(p_5-p_2)\cdot v_{23}=(-40,21,0)\cdot(0,4,32)=84$
42513	f_{23}	p_2	p_5	$(p_5-p_2)\cdot v_{23}=(-40,21,0)\cdot(0,4,32)=84$
42153	f_{23}	p_2	p_1	$(p_1-p_2)\cdot v_{23}=(-26,76,-3)\cdot(0,4,32)=208$
42135	f_{23}	p_2	p_1	$(p_1-p_2)\cdot v_{23}=(-26,76,-3)\cdot(0,4,32)=208$
41325	f_{24}	p_1	p_2	$(p_2-p_1)\cdot v_{24}=(26,-76,3)\cdot(4,0,32)=200$
41352	f_{24}	p_1	p_5	$(p_5-p_1)\cdot v_{24}=(-14,-55,3)\cdot(4,0,32)=40$
41235	f_{25}	p_1	p_5	$(p_5-p_1)\cdot v_{25}=(-14,-55,3)\cdot(0,-4,32)=316$
41253	f_{24}	p_1	p_5	$(p_5-p_1)\cdot v_{24}=(-14,-55,3)\cdot(4,0,32)=40$
41523	f_{24}	p_1	p_2	$(p_2-p_1)\cdot v_{24}=(26,-76,3)\cdot(4,0,32)=200$
41532	f_{25}	p_1	p_2	$(p_2-p_1)\cdot v_{25}=(26,-76,3)\cdot(0,-4,32)=400$
34521	f_{12}	p_3	p_4	$(p_4-p_3)\cdot v_{12}=(24,-8,-14)\cdot(4,0,-32)=544$
34512	f_{12}	p_3	p_4	$(p_4-p_3)\cdot v_{12}=(24,-8,-14)\cdot(4,0,-32)=544$
34251	f_{12}	p_3	p_4	$(p_4-p_3)\cdot v_{12}=(24,-8,-14)\cdot(4,0,-32)=544$
34215	f_{12}	p_3	p_4	$(p_4-p_3)\cdot v_{12}=(24,-8,-14)\cdot(4,0,-32)=544$
34125	f_{12}	p_3	p_4	$(p_4-p_3)\cdot v_{12}=(24,-8,-14)\cdot(4,0,-32)=544$
34152	f_{12}	p_3	p_4	$(p_4-p_3)\cdot v_{12}=(24,-8,-14)\cdot(4,0,-32)=544$
35421	f_{24}	p_5	p_2	$(p_2-p_5)\cdot v_{24}=(40,-21,0)\cdot(4,0,32)=160$
35412	f_{25}	p_5	p_2	$(p_2-p_5)\cdot v_{25}=(40,-21,0)\cdot(0,-4,32)=84$
35241	f_{13}	p_3	p_2	$(p_2-p_3)\cdot v_{13}=(1,9,10)\cdot(2,2,0)=20$
35214	f_{13}	p_3	p_2	$(p_2-p_3)\cdot v_{13}=(1,9,10)\cdot(2,2,0)=20$
35124	f_{23}	p_5	p_1	$(p_1-p_5)\cdot v_{23}=(14,55,-3)\cdot(0,4,32)=124$
35142	f_{23}	p_5	p_1	$(p_1-p_5)\cdot v_{23}=(14,55,-3)\cdot(0,4,32)=124$
32541	f_{23}	p_2	p_5	$(p_5-p_2)\cdot v_{23}=(-40,21,0)\cdot(0,4,32)=84$
32514	f_{23}	p_2	p_5	$(p_5-p_2)\cdot v_{23}=(-40,21,0)\cdot(0,4,32)=84$
32451	f_{13}	p_3	p_4	$(p_4-p_3)\cdot v_{13}=(24,-8,-14)\cdot(2,2,0)=32$
32415	f_{15}	p_3	p_5	$(p_5-p_3)\cdot v_{15}=(-39,30,10)\cdot(1,-3,16)=31$
32145	f_{23}	p_2	p_1	$(p_1-p_2)\cdot v_{23}=(-26,76,-3)\cdot(0,4,32)=208$
32154	f_{23}	p_2	p_1	$(p_1-p_2)\cdot v_{23}=(-26,76,-3)\cdot(0,4,32)=208$
31524	f_{24}	p_1	p_2	$(p_2-p_1)\cdot v_{24}=(26,-76,3)\cdot(4,0,32)=200$
31542	f_{25}	p_1	p_2	$(p_2-p_1)\cdot v_{25}=(26,-76,3)\cdot(0,-4,32)=400$
31254	f_{24}	p_1	p_5	$(p_5-p_1)\cdot v_{24}=(-14,-55,3)\cdot(4,0,32)=40$
31245	f_{25}	p_1	p_5	$(p_5-p_1)\cdot v_{25}=(-14,-55,3)\cdot(0,-4,32)=316$
31425	f_{24}	p_1	p_2	$(p_2-p_1)\cdot v_{24}=(26,-76,3)\cdot(4,0,32)=200$
31452	f_{24}	p_1	p_5	$(p_5-p_1)\cdot v_{24}=(-14,-55,3)\cdot(4,0,32)=40$
24351	f_{12}	p_2	p_4	$(p_4-p_2)\cdot v_{12}=(23,-17,-24)\cdot(4,0,-32)=860$
24315	f_{12}	p_2	p_4	$(p_4-p_2)\cdot v_{12}=(23,-17,-24)\cdot(4,0,-32)=860$
24531	f_{12}	p_2	p_4	$(p_4-p_2)\cdot v_{12}=(23,-17,-24)\cdot(4,0,-32)=860$
24513	f_{12}	p_2	p_4	$(p_4-p_2)\cdot v_{12}=(23,-17,-24)\cdot(4,0,-32)=860$
24153	f_{12}	p_2	p_4	$(p_4-p_2)\cdot v_{12}=(23,-17,-24)\cdot(4,0,-32)=860$
24135	f_{12}	p_2	p_4	$(p_4-p_2)\cdot v_{12}=(23,-17,-24)\cdot(4,0,-32)=860$
23451	f_{12}	p_2	p_3	$(p_3-p_2)\cdot v_{12}=(-1,-9,-10)\cdot(4,0,-32)=316$
23415	f_{12}	p_2	p_3	$(p_3-p_2)\cdot v_{12}=(-1,-9,-10)\cdot(4,0,-32)=316$
23541	f_{12}	p_2	p_3	$(p_3-p_2)\cdot v_{12}=(-1,-9,-10)\cdot(4,0,-32)=316$

Continued on next page

σ	f_{ij}	$\sigma(C_i)$	$\sigma(C_j)$	$(\sigma(C_i) - \sigma(C_j)) \cdot v_{ij}$
23514	f_{12}	p_2	p_3	$(p_3 - p_2) \cdot v_{12} = (-1, -9, -10) \cdot (4, 0, -32) = 316$
23154	f_{12}	p_2	p_3	$(p_3 - p_2) \cdot v_{12} = (-1, -9, -10) \cdot (4, 0, -32) = 316$
23145	f_{12}	p_2	p_3	$(p_3 - p_2) \cdot v_{12} = (-1, -9, -10) \cdot (4, 0, -32) = 316$
25341	f_{34}	p_3	p_4	$(p_4 - p_3) \cdot v_{34} = (24, -8, -14) \cdot (2, -2, 0) = 64$
25314	f_{45}	p_1	p_4	$(p_4 - p_1) \cdot v_{45} = (49, -93, -21) \cdot (-2, -3, 8) = 13$
25431	f_{13}	p_2	p_4	$(p_4 - p_2) \cdot v_{13} = (23, -17, -24) \cdot (2, 2, 0) = 12$
25413	f_{13}	p_2	p_4	$(p_4 - p_2) \cdot v_{13} = (23, -17, -24) \cdot (2, 2, 0) = 12$
25143	f_{23}	p_5	p_1	$(p_1 - p_5) \cdot v_{23} = (14, 55, -3) \cdot (0, 4, 32) = 124$
25134	f_{23}	p_5	p_1	$(p_1 - p_5) \cdot v_{23} = (14, 55, -3) \cdot (0, 4, 32) = 124$
21354	f_{24}	p_1	p_5	$(p_5 - p_1) \cdot v_{24} = (-14, -55, 3) \cdot (4, 0, 32) = 40$
21345	f_{25}	p_1	p_5	$(p_5 - p_1) \cdot v_{25} = (-14, -55, 3) \cdot (0, -4, 32) = 316$
21534	f_{34}	p_5	p_3	$(p_3 - p_5) \cdot v_{34} = (39, -30, -10) \cdot (2, -2, 0) = 138$
21543	f_{25}	p_1	p_3	$(p_3 - p_1) \cdot v_{25} = (25, -85, -7) \cdot (0, -4, 32) = 116$
21453	f_{24}	p_1	p_5	$(p_5 - p_1) \cdot v_{24} = (-14, -55, 3) \cdot (4, 0, 32) = 40$
21435	f_{25}	p_1	p_5	$(p_5 - p_1) \cdot v_{25} = (-14, -55, 3) \cdot (0, -4, 32) = 316$
14325	f_{12}	p_1	p_4	$(p_4 - p_1) \cdot v_{12} = (49, -93, -21) \cdot (4, 0, -32) = 868$
14352	f_{12}	p_1	p_4	$(p_4 - p_1) \cdot v_{12} = (49, -93, -21) \cdot (4, 0, -32) = 868$
14235	f_{12}	p_1	p_4	$(p_4 - p_1) \cdot v_{12} = (49, -93, -21) \cdot (4, 0, -32) = 868$
14253	f_{12}	p_1	p_4	$(p_4 - p_1) \cdot v_{12} = (49, -93, -21) \cdot (4, 0, -32) = 868$
14523	f_{12}	p_1	p_4	$(p_4 - p_1) \cdot v_{12} = (49, -93, -21) \cdot (4, 0, -32) = 868$
14532	f_{12}	p_1	p_4	$(p_4 - p_1) \cdot v_{12} = (49, -93, -21) \cdot (4, 0, -32) = 868$
13425	f_{12}	p_1	p_3	$(p_3 - p_1) \cdot v_{12} = (25, -85, -7) \cdot (4, 0, -32) = 324$
13452	f_{12}	p_1	p_3	$(p_3 - p_1) \cdot v_{12} = (25, -85, -7) \cdot (4, 0, -32) = 324$
13245	f_{12}	p_1	p_3	$(p_3 - p_1) \cdot v_{12} = (25, -85, -7) \cdot (4, 0, -32) = 324$
13254	f_{12}	p_1	p_3	$(p_3 - p_1) \cdot v_{12} = (25, -85, -7) \cdot (4, 0, -32) = 324$
13524	f_{12}	p_1	p_3	$(p_3 - p_1) \cdot v_{12} = (25, -85, -7) \cdot (4, 0, -32) = 324$
13542	f_{12}	p_1	p_3	$(p_3 - p_1) \cdot v_{12} = (25, -85, -7) \cdot (4, 0, -32) = 324$
12345	f_{12}	p_1	p_2	$(p_2 - p_1) \cdot v_{12} = (26, -76, 3) \cdot (4, 0, -32) = 8$
12354	f_{12}	p_1	p_2	$(p_2 - p_1) \cdot v_{12} = (26, -76, 3) \cdot (4, 0, -32) = 8$
12435	f_{12}	p_1	p_2	$(p_2 - p_1) \cdot v_{12} = (26, -76, 3) \cdot (4, 0, -32) = 8$
12453	f_{12}	p_1	p_2	$(p_2 - p_1) \cdot v_{12} = (26, -76, 3) \cdot (4, 0, -32) = 8$
12543	f_{12}	p_1	p_2	$(p_2 - p_1) \cdot v_{12} = (26, -76, 3) \cdot (4, 0, -32) = 8$
12534	f_{12}	p_1	p_2	$(p_2 - p_1) \cdot v_{12} = (26, -76, 3) \cdot (4, 0, -32) = 8$
15324	f_{24}	p_5	p_2	$(p_2 - p_5) \cdot v_{24} = (40, -21, 0) \cdot (4, 0, 32) = 160$
15342	f_{25}	p_5	p_2	$(p_2 - p_5) \cdot v_{25} = (40, -21, 0) \cdot (0, -4, 32) = 84$
15234	f_{34}	p_2	p_3	$(p_3 - p_2) \cdot v_{34} = (-1, -9, -10) \cdot (2, -2, 0) = 16$
15243	f_{15}	p_1	p_3	$(p_3 - p_1) \cdot v_{15} = (25, -85, -7) \cdot (1, -3, 16) = 168$
15423	f_{24}	p_5	p_2	$(p_2 - p_5) \cdot v_{24} = (40, -21, 0) \cdot (4, 0, 32) = 160$
15432	f_{25}	p_5	p_2	$(p_2 - p_5) \cdot v_{25} = (40, -21, 0) \cdot (0, -4, 32) = 84$

We show now that the polyhedral fan does not contain a cycle in any direction. To this end, we check all simple cycles in the dual graph, which has one vertex per cell and one edge per facet. If the halfspaces defined by a cycle (with the corresponding orientation) have empty intersection, then there is no cycle in visibility involving the facets in the cycle. The graph associated to $\mathcal{F}$ is depicted in Figure D.2.

Note first that the halfspaces associated to cycles around a ray have empty intersection. This excludes the cycles $\langle C_1, C_2, C_3\rangle$,$\langle C_2, C_3, C_4\rangle$, $\langle C_2, C_4, C_5\rangle$ and $\langle C_1, C_2, C_5\rangle$.

We give next certificates for the remaining cycles. To show that the halfspaces have empty

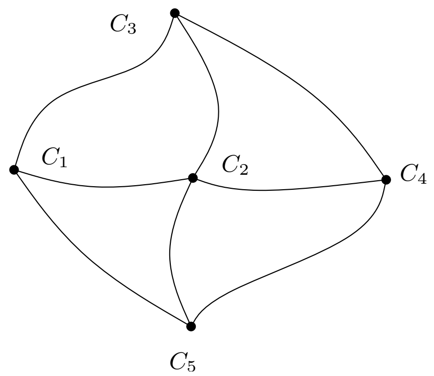


Figure D.2: Dual graph of $\mathcal{F}$.

intersection we provide a solution to the dual problem (in the sense of Lemma 4.5).

$$\begin{aligned}
\langle C_1, C_3, C_4, C_5\rangle &: v_{13} + \frac{1}{4}v_{34} + v_{45} - \frac{1}{2}v_{15} = 0\\
\langle C_2, C_5, C_1, C_3\rangle &: \frac{3}{4}v_{25} - \frac{1}{4}v_{15} + \frac{1}{2}v_{13} - \frac{1}{4}v_{23} = 0\\
\langle C_2, C_1, C_3, C_4\rangle &: -\frac{1}{2}v_{12} + v_{13} + v_{34} - \frac{1}{2}v_{24} = 0\\
\langle C_2, C_3, C_4, C_5\rangle &: \frac{1}{2}v_{23} + v_{34} + v_{45} - \frac{3}{4}v_{25} = 0\\
\langle C_2, C_4, C_5, C_1\rangle &: \frac{1}{2}v_{24} + v_{45} - v_{15} + \frac{1}{4}v_{12} = 0\\
\langle C_2, C_1, C_3, C_4, C_5\rangle &: -\frac{1}{2}v_{12} + v_{13} + v_{34} + v_{45} - \frac{3}{4}v_{25} = 0\\
\langle C_2, C_3, C_4, C_5, C_1\rangle &: \frac{1}{2}v_{23} + v_{34} + v_{45} - v_{15} + \frac{1}{4}v_{12} = 0\\
\langle C_2, C_4, C_5, C_1, C_3\rangle &: \frac{1}{2}v_{24} + v_{45} - v_{15} + \frac{1}{2}v_{13} - \frac{1}{4}v_{23} = 0\\
\langle C_2, C_5, C_1, C_3, C_4\rangle &: \frac{3}{4}v_{25} - \frac{1}{4}v_{15} + \frac{1}{2}v_{13} + \frac{1}{4}v_{34} - \frac{1}{4}v_{24} = 0
\end{aligned}$$

Bibliography

[1] A. Abdulkadiroğlu and T. Sönmez. Random serial dictatorship and the core from random endowments in house allocation problems. *Econometrica*, 66(3):689–701, 1998.

[2] P. K. Agarwal, A. Efrat, and M. Sharir. Vertical decomposition of shallow levels in 3-dimensional arrangements and its applications. *SIAM Journal on Computing*, 29:39–50, 1999.

[3] P. K. Agarwal and M. Sharir. Arrangements and Their Applications. In *Handbook of Computational Geometry*, pages 49–119. Elsevier Science Publishers B.V. North-Holland, 1998.

[4] O. Aichholzer, F. Aurenhammer, S.-W. Cheng, N. Katoh, G. Rote, M. Taschwer, and Y.-F. Xu. Triangulations intersect nicely. *Discrete & Computational Geometry*, 16(4):339–359, 1996.

[5] D. Akca. *Least Squares 3D Surface Matching.* PhD thesis, Institut für Geodäsie und Photogrammetrie an der Eidgenössischen Technischen Hochschule, Zürich, 2007.

[6] D. Alberts. A Generalization of the Hungarian Method for the c-Assignment Problem. Diplomarbeit, Freie Universität Berlin, 1992.

[7] H. Alt, N. Blum, K. Mehlhorn, and M. Paul. Computing a maximum cardinality matching in a bipartite graph in time $O(n^{1.5}\sqrt{m/\log n})$. *Information Processing Letters*, 37(4):237–240, Feb. 1991.

[8] H. Alt and L. J. Guibas. Discrete Geometric Shapes: Matching, Interpolation, and Approximation: A Survey. In *Handbook of Computational Geometry*, pages 121–154. 1996.

[9] H. Alt, K. Mehlhorn, H. Wagener, and E. Welzl. Congruence, similarity, and symmetries of geometric objects. *Discrete & Computational Geometry*, 3(1):237–256, 1988.

[10] A. Asinowski, B. Keszegh, and T. Miltzow. Counting one-sided exchange stable matchings. *ArXiv e-prints*, 1401.5354, Jan. 2014.

[11] F. Aurenhammer. A criterion for the affine equivalence of cell complexes in $\mathbb{R}^d$ and convex polyhedra in $\mathbb{R}^{d+1}$. *Discrete & Computational Geometry*, 2(1):49–64, 1987.

[12] F. Aurenhammer. Power diagrams: properties, algorithms and applications. *SIAM Journal on Computing*, 16(1):78–96, Feb. 1987.

[13] F. Aurenhammer, F. Hoffmann, and B. Aronov. Minkowski-type theorems and least-squares clustering. *Algorithmica*, 20:61–76, 1998.

[14] F. Aurenhammer, R. Klein, and D.-T. Lee. *Voronoi Diagrams and Delaunay Triangulations.* World Scientific, 2013.

[15] D. Avis, K. Fukuda, and S. Picozi. On canonical representations of convex polyhedra. In A. M. Cohen, X.-S. Gao, and N. Takayama, editors, *Mathematical Software, Proceedings of the First International Congress of Mathematical Software*, pages 350–360. World Scientific Publishing, 2002.

[16] M. Balzer and O. Deussen. Voronoi treemaps. In *IEEE Symposium on Information Visualization* , page 7, 2005.

[17] S. Basu, R. Pollak, and M.-F. Roy. On the number of cells defined by a family of polynomials on a variety. *Mathematika*, 43(01):120–126, 1996.

[18] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, Apr. 2002.

[19] R. Ben-Avraham, M. Henze, R. Jaume, B. Keszegh, O. Raz, M. Sharir, and I. Tubis. Minimum partial-matching and hausdorff rms-distance under translation: Combinatorics and algorithms. In A. Schulz and D. Wagner, editors, *Algorithms - ESA 2014*, volume 8737 of *Lecture Notes in Computer Science*, pages 100–111. Springer Berlin Heidelberg, 2014.

[20] R. Ben-Avraham, M. Henze, O. E. Raz, R. Jaume, M. Sharir, B. Keszegh, and I. Tubis. Minimum Partial-Matching and Hausdorff RMS-Distance under Translation: Combinatorics and Algorithms. In A. S. Schulz and D. Wagner, editors, *Proc. 22th Annual European Symposium*, volume 8737 of *LNCS*, pages 100–111, 2014.

[21] C. Berge. Two theorems in graph theory. In *Proc. of the National Academy of Sciences of the United States of America*, volume 43, pages 842–844, 1957.

[22] P. J. Besl and N. D. McKay. A method for registration of 3D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, Feb. 1992.

[23] L. J. Billera, P. Filliman, and B. Sturmfels. Constructions and complexity of secondary polytopes. *Advances in Mathematics*, 83(2):155–179, 1990.

[24] A. Bishnu, S. Das, S. C. Nandy, and B. B. Bhattacharya. Simple algorithms for partial point set pattern matching under rigid motion. *Pattern Recognition*, 39(9):1662–1671, 2006.

[25] E. Boros and P. L. Hammer. On clustering problems with connected optima in Euclidean spaces. *Discrete Mathematics*, 75(1–3):81–88, 1989.

[26] P. Bose, L. J. Guibas, A. Lubiw, M. H. Overmars, D. L. Souvaine, and J. Urrutia. The floodlight problem. *International Journal of Computational Geometry and Applications*, 7(1/2):153–163, 1997.

[27] R. Burkard, M. Dell'Amico, and S. Martello. *Assignment Problems.* Society for Industrial and Applied Mathematics, 2009.

[28] R. E. Burkard and F. Rendl. Lexicographic bottleneck problems. *Operations Research Letters*, 10(5):303–308, 1991.

[29] S. Cabello, P. Giannopoulos, and C. Knauer. On the parameterized complexity of d-dimensional point set pattern matching. *Information Processing Letters*, 105(2):73–77, 2008.

[30] S. Cabello, P. Giannopoulos, C. Knauer, and G. Rote. Matching point sets with respect to the Earth Mover's Distance. *Computational Geometry: Theory and Applications*, 39(2):118–133, 2008.

[31] V. Capoyleas, G. Rote, and G. J. Woeginger. Geometric clusterings. *Journal of Algorithms*, 12(2):341–356, 1991.

[32] M. Cary, A. Rudra, A. Sabharwal, and E. Vee. Floodlight illumination of infinite wedges. *Computational Geometry: Theory and Applications*, 43(1):23–34, Jan. 2010.

[33] T. M. Chan. Optimal output-sensitive convex hull algorithms in two and three dimensions. *Discrete & Computational Geometry*, 16:361–368, 1996.

[34] T. M. Chan. Random sampling, halfspace range reporting, and construction of ($\leq k$)-levels in three dimensions. *SIAM Journal on Computing*, pages 586–595, 1999.

[35] B. Chazelle. An optimal convex hull algorithm in any fixed dimension. *Discrete & Computational Geometry*, 10(1):377–409, 1993.

[36] B. Chazelle, L. Guibas, and D. Lee. The power of geometric duality. *BIT Numerical Mathematics*, 25(1):76–90, 1985.

[37] B. Chazelle and J. Matoušek. Derandomizing an output-sensitive convex hull algorithm in three dimensions. *Computational Geometry: Theory and Applications*, 5(1):27–32, 1995.

[38] K. L. Clarkson and P. W. Shor. Applications of random sampling in computational geometry, II. *Discrete & Computational Geometry*, 4:387–421, 1989.

[39] S. Cohen and L. Guibas. The Earth Mover's Distance under transformation sets. In *Proceedings of the International Conference on Computer Vision*, volume 2 of *ICCV '99*, pages 1076–1083. IEEE Computer Society, 1999.

[40] R. Connelly. Rigidity and energy. *Inventiones Mathematicae*, pages 11–33, 1982.

[41] H. Crapo and W. Whiteley. Spaces of stresses, projections and parallel drawings for spherical polyhedra. *Contributions to Algebra and Geometry*, 35(2):259–281, 1994.

[42] F. D. Croce, V. T. Paschos, and A. Tsoukias. An improved general procedure for lexicographic bottleneck problems. *Operations Research Letters*, 24(4):187–194, 1999.

[43] J. Czyzowicz, E. Rivera-Campo, and J. Urrutia. Optimal floodlight illumination of stages. In *Canadian Conference on Computational Geometry*, pages 393–398. University of Waterloo, 1993.

[44] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars. *Computational Geometry: Algorithms and Applications*. Springer-Verlag TELOS, Santa Clara, CA, USA, 3rd ed. edition, 2008.

[45] J. De Loera, J. Rambau, and F. Santos. *Triangulations*. Springer, 2010.

[46] J. Dietel, H.-D. Hecker, and A. Spillner. A note on optimal floodlight illumination of stages. *Information Processing Letters*, 105(4):121–123, 2008.

[47] H. Edelsbrunner. *Algorithms in Combinatorial Geometry*, volume 10 of *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, Berlin, Heidelberg, 1987.

[48] H. Edelsbrunner. An acyclicity theorem for cell complexes in d dimension. *Combinatorica*, 10(3):251–260, 1990.

[49] H. Edelsbrunner and L. J. Guibas. Topologically sweeping an arrangement. *Journal of Computer and System Sciences*, 38(1):165–194, 1989.

[50] H. Edelsbrunner and R. Seidel. Voronoi diagrams and arrangements. *Discrete & Computational Geometry*, 1(1):25–44, 1986.

[51] J. Edmonds and R. M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM*, 19(2):248–264, Apr. 1972.

[52] A. Efrat, A. Itai, and M. J. Katz. Geometry helps in bottleneck matching and related problems. *Algorithmica*, 31(1):1–28, 2001.

[53] T. Eiter and H. Mannila. Distance measures for point sets and their computation. *Acta Informatica*, 34:103–133, 1997.

[54] H. Everett, J.-M. Robert, and M. van Kreveld. An optimal algorithm for the ($\leq k$)-levels, with applications to separation and transversal problems. In *Proceedings of the Ninth Annual Symposium on Computational Geometry*, pages 38–46, New York, NY, USA, 1993. ACM.

[55] E. Ezra, M. Sharir, and A. Efrat. On the ICP algorithm. In *Proceedings of the Twenty-second Annual Symposium on Computational Geometry*, SCG '06, pages 95–104, New York, NY, USA, 2006. ACM.

[56] L. Fejes Tóth. Illumination of convex discs. *Acta Mathematica Academiae Scientiarum Hungaricae*, 29(3-4):355–360, 1977.

[57] P. C. Fishburn. Lexicographic orders, utilities and decision rules: a survey. *Management Science*, 20(11):pp. 1442–1471, 1974.

[58] M. L. Fredman and R. E. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM*, 34(3):596–615, July 1987.

[59] H. N. Gabow and R. E. Tarjan. Algorithms for two bottleneck optimization problems. *Journal of Algorithms*, 9(3):411–417, Sept. 1988.

[60] D. Gale and L. S. Shapley. College admissions and the stability of marriage. *American Mathematical Monthly*, 69(1):9–15, 1962.

[61] V. Galperin and G. Galperin. Osveshchenije ploskosti prozhektorami (in Russian). *Kvant*, 11:28–30, 1981.

[62] J. Galtier, F. Hurtado, M. Noy, S. Perennes, and J. Urrutia. Simultaneous edge flipping in triangulations. *International Journal of Computational Geometry and Applications*, 13(2), Apr. 2003.

[63] I. Gelfand, M. Kapranov, and A. Zelevinsky. *Discriminants, Resultants, and Multidimensional Determinants*. Modern Birkhäuser Classics. Birkhäuser Boston, 2008.

[64] N. Gelfand and S. Rusinkiewicz. Geometrically stable sampling for the ICP algorithm. In *Proc. International Conference on 3D Digital Imaging and Modeling*, pages 260–267, 2003.

[65] W. Grimson, R. Kikinis, F. Jolesz, and P. Black. Image-guided surgery. *Scientific American*, 280(6):62–69, 06 1999.

[66] D. Gusfield and R. W. Irving. *The Stable Marriage problem: Structure and Algorithms.* MIT Press, Cambridge, MA, USA, 1989.

[67] S. Har-Peled and B. Sadri. How fast is the k-means method? *Algorithmica*, 41(3):185–202, January 2005.

[68] M. Henze and R. Jaume. Bottleneck partial-matching Voronoi diagrams. In *Proc. 30th European Workshop on Computational Geometry*, 2014.

[69] M. Henze and R. Jaume. Bottleneck Partial-Matching Voronoi Diagrams and Applications. In H.-K. Ahn and C.-S. Shin, editors, *The 25th International Symposium on Algorithms and Computation*, volume 8889 of *LNCS*, pages 714–725, 2014.

[70] M. Henze, R. Jaume, and B. Keszegh. On the complexity of the partial least-squares matching Voronoi diagram. In *Proc. 29th European Workshop on Computational Geometry*, pages 193–196, 2013.

[71] J. E. Hopcroft and R. M. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM Journal on Computing*, 2(4):225–231, 1973.

[72] R. W. Irving and P. Leather. The complexity of counting stable marriages. *SIAM Journal on Computing*, 15(3):655–667, Aug. 1986.

[73] H. Ito, H. Uehara, and M. Yokoyama. NP-Completeness of stage illumination problems. In J. Akiyama, M. Kano, and M. Urabe, editors, *Discrete and Computational Geometry*, volume 1763 of *Lecture Notes in Computer Science*, pages 158–165. Springer Berlin Heidelberg, 2000.

[74] K. Iwama and S. Miyazaki. A survey of the stable marriage problem and its variants. In *Proc. International Conference on Informatics Education and Research for Knowledge-Circulating Society*, pages 131–136, Washington, DC, USA, 2008. IEEE Computer Society.

[75] R. Jaume and G. Rote. The finest regular coarsening and recursively-regular subdivisions. *ArXiv e-prints*, 1310.4372, Oct. 2013.

[76] B. Johnson and B. Shneiderman. Tree-maps: a space-filling approach to the visualization of hierarchical information structures. In *Proceedings of the 2nd Conference on Visualization '91*, VIS '91, pages 284–291, Los Alamitos, CA, USA, 1991. IEEE Computer Society Press.

[77] D. G. Kirkpatrick and R. Seidel. The ultimate planar convex hull algorithm. *SIAM Journal on Computing*, 15(1):287–299, 1986.

[78] D. A. Klain. The Minkowski problem for polytopes. *Advances in Mathematics*, 185(2):270–288, 2004.

[79] V. Klee. On the complexity of d-dimensional Voronoi diagrams. *Archiv der Mathematik*, 34(1):75–80, 1980.

[80] D. E. Knuth. *Mariages stables et leurs relations avec d'autres problèmes combinatoires.* Les Presses de l'Université de Montréal, Montreal, Quebec., 1976.

[81] H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.

[82] J. C. Maxwell. On reciprocal figures and diagrams of forces. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 27:250–261, 1864.

[83] P. McMullen. The maximum numbers of faces of a convex polytope. *Mathematika*, 17:179–184, 12 1970.

[84] A. Nocaj and U. Brandes. Computing Voronoi treemaps: faster, simpler, and resolution-independent. *Computer Graphics Forum*, 31:855–864, June 2012.

[85] I. Paschinger. *Konvexe Polytope und dirichletsche Zellenkomplexe.* PhD thesis, Institut für Mathematik, University of Salzburg, Austria, 1982.

[86] J. M. Phillips and P. K. Agarwal. On bipartite matching under the RMS distance. In *Canadian Conference on Computational Geometry*, pages 143–146, 2006.

[87] F. Preparata and M. Shamos. *Computational Geometry: an Introduction.* Texts and monographs in computer science. Springer-Verlag, 1988.

[88] F. P. Preparata. An optimal real-time algorithm for planar convex hulls. *Communications of the ACM*, 22(7):402–405, July 1979.

[89] A. P. Punnen and K. Nair. Improved complexity bound for the maximum cardinality bottleneck bipartite matching problem. *Discrete Applied Mathematics*, 55(1):91–93, 1994.

[90] L. Ramshaw and R. E. Tarjan. On minimum-cost assignments in unbalanced bipartite graphs. Technical report, HP Labs Technical Report HPL-2012-40 (June 2012), 2012.

[91] P. Rezende and D. Lee. Point set pattern matching in d-dimensions. *Algorithmica*, 13(4):387–404, 1995.

[92] G. Rote. Two applications of point matching. In *Proc. 25th European Workshop on Computational Geometry*, Brussels, Belgium, March 2009.

[93] G. Rote. Partial least-squares point matching under translations. In *Proc. 26th European Workshop on Computational Geometry*, pages 249–251, 2010.

[94] G. Rote. Personal communication, 2014.

[95] B. Roth and W. Whiteley. Tensegrity frameworks. *Transactions of the American Mathematical Society*, 265(2):419–446, 1981.

[96] Y. Rubner, C. Tomasi, and L. J. Guibas. The Earth Mover's Distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, Nov. 2000.

[97] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*, pages 145–152, 2001.

[98] F. Santos. Non-connected toric Hilbert schemes. *Mathematische Annalen*, 332:645–665, 2005.

[99] M. I. Shamos and D. Hoey. Closest-point problems. In *FOCS*, pages 151–162, 1975.

[100] L. S. Shapley and H. Scarf. On cores and indivisibility. *Journal of Mathematical Economics*, 1(1):23–37, 1974.

[101] M. Sharir. The Clarkson-Shor technique revisited and extended. *Combinatorics, Probability & Computing*, 12(2):191–201, Mar. 2003.

[102] M. Sharir. Private communication, 2013.

[103] P. T. Sokkalingam and Y. P. Aneja. Lexicographic bottleneck combinatorial problems. *Operations Research Letters*, 23(1–2):27–33, 1998.

[104] N. Tomizawa. On some techniques useful for solution of transportation network problems. *Networks*, 1(2):173–194, 1971.

[105] J. Urrutia. Art Gallery and Illumination Problems. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, chapter 22, pages 973–1027. North-Holland, Amsterdam, 2000.

[106] P. Vaidya. Geometry helps in matching. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, STOC '88, pages 422–425, New York, NY, USA, 1988. ACM.

[107] T. D. Wu, S. C. Schmidler, T. Hastie, and D. L. Brutlag. Modeling and superposition of multiple protein structures using affine transformations: Analysis of the globins. *Pacific Symposium on Biocomputing*, 3:507–518, 1998.

[108] W. Zeng, G. Zhu, and Y. Li. Point matching estimation for moving object tracking based on Kalman filter. In H. Miao and G. Hu, editors, *ACIS-ICIS*, pages 1115–1119. IEEE Computer Society, 2009.

[109] G. Ziegler. *Lectures on Polytopes*. Graduate Texts in Mathematics. Springer New York, 1995.

[110] K. Zikan and T. M. Silberberg. The Frobenius Metric in Image Registration. In *Computer Vision and Image Processing*, pages 385–420. Elsevier Science, 1992.

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit ohne fremde Hilfe und nur mit den angegebenen Quellen verfasst habe. Die Stellen, die ich dem Wortlaut oder dem Sinn nach anderen Werken entnommen habe, sind durch Angabe der Quellen kenntlich gemacht.

Berlin,