Johannes EDELSBRUNNER

Flip-graphs of triangulations and the secondary polytope

DIPLOMA THESIS

written to obtain the academic degree of a Master of Science (MSc)

Diploma Study Technical Mathematics

Graz University of Technology

Supervisor:
Assoc.Prof. Dipl.-Ing. Dr.techn. Oswin AICHHOLZER

Institute for Software Technology

Graz, March 2012

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am ....
(Unterschrift)

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

....
date (signature)

Abstract

This thesis focuses on the connectedness of the flip-graph of triangulations and the related secondary polytope. A comprehensive overview of the most important structures used for construction and analysis of the secondary polytope is given, the relation to the flip-graph of regular triangulations is shown, and various other properties of the secondary polytope are presented. The definition of a constrained secondary polytope, which is able to relate non-regular triangulations to the flip-graph of regular triangulations, is established. This gives new possibilities to investigate non-regular triangulations in the flip-graph of all triangulations. For various examples of point-configurations, secondary polytopes and subdivision posets are shown and illustrated with figures. An algorithm for the computation of flip-paths between regular triangulations, based on linear optimization on the secondary polytope, is given. The length of the produced flip-paths and the runtime of the algorithm are evaluated, and it turns out that the algorithm yields the best results for 2-dimensional point-configurations in convex position.

Zusammenfassung

Diese Arbeit beschäftigt sich mit dem Flipgraphen von Triangulierungen, speziell mit der Frage, ob dieser zusammenhängend ist, sowie mit dem verwandten Secondary Polytope. Zuerst wird eine Übersicht der wichtigsten Strukturen zur Konstruktion und Untersuchung des Secondary Polytopes gegeben, und danach die Beziehung zwischen dem Secondary Polytope und dem Flipgraphen von Triangulierungen herausgearbeitet. Zusätzlich werden weitere Eigenschaften des Secondary Polytopes untersucht und eine Abwandlung des Secondary Polytopes, das Constrained Secondary Polytope, definiert. Dieses setzt nicht-reguläre Triangulierungen mit dem Flipgraphen von regulären Triangulierungen in Beziehung. Damit können Flips zu oder von nicht-regulären Triangulierungen untersucht werden. Unterstützend werden mehrere Beispiele von Punktkonfigurationen gegeben und deren Secondary Polytopes und Subdivision Posets untersucht und illustriert. Abschließend wird ein Algorithmus für die Berechnung von Flippfaden zwischen regulären Triangulierungen präsentiert, welcher auf linearer Optimierung auf dem Secondary Polytope basiert. Dabei werden die Längen der produzierten Flippfade und die Laufzeit des Algorithmus analysiert. Es zeigt sich, dass der Algorithmus die besten Resultate für 2-dimensionale Punktkonfigurationen in konvexer Lage liefert.

Acknowledgments

I want to thank my supervisor Prof. Dr. Oswin Aichholzer for giving me the opportunity to work on the topic of this thesis. Although this thesis took longer to write than expected, and time was getting short at the end, I am happy that we could finish it in the presented constitution.

Parts of the issues treated here have come from a seminar in computational geometry of Prof. Dr. Franz Aurenhammer and he also deserves my thanks.

Very grateful I am to Dr. Thomas Hackl for taking the time to read my drafts and giving me advice during extended discussions.

Thanks goes also to my parents and my girlfriend for their support and patience until the end of the work.

Contents

1 Introduction

In computer graphics, three-dimensional modeling, numerical methods, and many other mathematics-related fields triangulations play an important role. For a given set of points in 2 dimensions, a triangulation is the decomposition of their convex hull into triangles. Every corner of such a triangle must be one of the given points, and two triangles are not allowed to overlap. But triangulations also exist in higher dimensions. For instance, for a given set of points in 3 dimensions a triangulation is the decomposition into tetrahedra. In general, a triangulation is a decomposition into simplices for an arbitrary-dimensional set of points.

For a given set of points, one can now ask how to transform one triangulation into another. The operations therefor are called flips. The most simple flip (edge exchange flip) is performed by taking out one edge of a 2-dimensional triangulation (so that the two triangles that were incident to the edge are now "joined" to a convex quadrangle) and the second possible edge is inserted. But this is of course only possible when the quadrangle is convex, and that already shows that flips cannot always be performed. There are also two other kinds of flips in 2 dimensions (point inserting/point removing flip) and various other kinds in other dimensions.

The question is now, does there exist a sequence of flips that transforms the start triangulation into the target triangulation?

For dimension 2 this is true, which is shown in [9, chapter 1.2]. For higher dimensions, however, there were counter-examples constructed. See [15] for an example in dimension 5, [7, chapter 7.3, chapter 7.4] for examples in dimension 5 and 6, or [14] for examples in dimension 6 and 234. There is no answer yet for the especially interesting dimension 3.

The question for the existence of such a sequence can be transformed into another question. If we see this problem as a graph, we relate each vertex of the graph to one triangulation and each flip to one edge. So if one can travel from one vertex to the other, this means that there exists a flip from one related triangulation to the other. This graph is called the flip-graph of triangulations. The existence of a sequence of flips from a start triangulation to a target triangulation now implies the existence of a path from a start vertex to a target vertex. If we want to get from an arbitrary triangulation to another arbitrary triangulation, this just means that the flip-graph must be connected.

1.1 The connectedness of the flip-graph

How can one find out whether the flip-graph is connected? One can try to start with a triangulation and try to flip successively to other triangulations. But how can we know if we have reached all triangulations? Moreover, solving the problem only for the triangulations of one set of points is unsatisfying, we want it to be solved for a whole class of point sets. As it turns out, the dimension of the set of points is crucial. Because of that there are in general results for all point sets of a particular dimension.

If we are in 2 dimensions, investigating triangulations and flips is comparatively easy. Triangulations can be drawn without problems, flips can be performed by switching an edge, and connections can be seen with relative ease. In 3 dimensions it is already much more complicated, since the building blocks of triangulations are now tetrahedra, which are 3-dimensional objects. Drawing a not completely ordinary 3-dimensional triangulation means drawing various tetrahedra stacked over each other. A clear and unambiguous visualization can be hard to achieve. In higher dimensions the visualization gets virtually impossible. This leads to the problem that trying out point constellations, triangulations, and flips is easy for the 2-dimensional case, but hard in higher dimensions. But while dimension 2 is already well understood, many unanswered questions exist in higher dimensions.

When it is not possible to experiment with examples, one must find other ways to analyze the situation. Therefore, in order to answer the question whether a flip-graph is connected, a multitude of properties for triangulations and structures related to the flip-graph have been defined.

Perhaps the most important property is the one of regularity of a triangulation. A d-dimensional triangulation is called regular, when there exists a certain higher-dimensional polytope, for which a certain projection back to d dimensions results in that particular triangulation. Otherwise it is called non-regular. It turns out that the regular flip-graph (the flip-graph containing only the regular triangulations) is always connected (independent from the dimension of the set of points). This is described in [7, chapter 5.3]. So the non-regular triangulations are obviously the ones which might be problematic.

An important structure related to the flip-graph is the secondary polytope. It is a high-dimensional polytope whose vertices are in a one-to-one correspondence to the regular triangulations. Additionally, the edges correspond to flips between these triangulations. So the so-called "1-skeleton", the graph of all vertices and edges from the secondary-polytope, is the regular flip-graph of triangulations. From this,

it can obviously also be deduced that the regular flip-graph is connected. Moreover, there is also a one-to-one correspondence between all faces of the secondary polytope and the regular subdivisions. A subdivision is pretty similar to a triangulation in that it decomposes the convex hull of the given set of points, unlike triangulations not necessarily into triangles, but into convex polyhedra. As before, it can be regular or non-regular.

One deficient of the secondary polytope is that one can give numerous statements about regular triangulations (or subdivisions), but not about non-regular ones. Therefore, this work introduces the notion of constrained secondary polytopes, which are polytopes that lie in the secondary polytope, consist partially of their faces, and can have non-regular triangulations as their vertices.

1.2 Finding paths in the flip-graph

After answering the question whether there exists a flip-path from one triangulation to another, the next question is how to find such a path, and furthermore how to find the shortest path.

The secondary polytope already includes the regular flip-graph. So, for regular triangulations, the secondary polytope is a good starting point for the search of such a path. Since it is a polytope, various optimization methods can be tried to get from one vertex to the other. For non-regular triangulations, working with constrained secondary polytopes might also help to find such paths.

An algorithm for finding paths between regular triangulations is presented in this work. It uses linear optimization on the secondary polytope. The runtime of the algorithm and the quality of the generated flip-paths are also evaluated.

1.3 Related work

Gelfand, Kapranov and Zelevinsky have introduced the secondary polytope and have shown its relation to the flip-graph of triangulations. Some of their works are in Russian, but the book [11] is in English and contains the secondary polytope and results for it. The paper [4] gives a study of secondary polytopes and shows additional options for its construction. An extensive overview of triangulations and related topics is given in [7]. Included are regularity, secondary polytopes, flip-paths, and many other topics.

Concerned with flip-paths between regular triangulations is paper [13]. For 2-dimensional point-configurations in convex position the flip-graph has many con-

nections with other mathematical fields. These are shown and investigated in [17]. A survey of flips in planar graphs (including a broader definition of flip, also existing in other structures than triangulations) is given in [6].

1.4 Organization of work

Because triangulations, secondary polytopes, and all the other structures are geometric objects, the attempt to visualize them is only natural. The concepts are often much easier to understand with the addition of graphics, and so graphical representations are used in many cases.

The work is divided into several sections which are:

Section 1 is the current section and gives an introduction to the topic. The motivation for the thesis is explained and related work is referenced.

Section 2 introduces the basic definitions and structures. As it might already be clear from the introduction above, there are many structures that can be used to analyze triangulations and their relations. Exact definitions for point-configurations (essentially sets of points), triangulations, subdivisions, and more are given in order to avoid ambiguities later on.

Section 3 introduces the more advanced structures that ultimately lead to the secondary polytope. The properties of the secondary polytope, its relation to triangulations and subdivisions, and how non-regular triangulations can be related to the secondary polytope, are shown.

Section 4 gives a list of examples of interesting point-configurations. For every example visualizations of the mentioned structures are given. They should enhance the intuition and understanding for these structures.

Section 5 presents an algorithm related to the problem of finding a flip-path between regular triangulations. It is based on linear optimization on the secondary polytope.

Section 6 summarizes the presented work and its results.

2 General Definitions

In order to analyze triangulations and flips thereon, we will need several structures. Because of that, the first part of this work is dedicated to general definitions, which will be used to build these structures. The definitions are widely inspired by the book "Triangulations" from De Loera, Rambau, and Santos [7], which gives an extensive overview of triangulations and all kinds of related structures. We will restate the relevant definitions, try to make some of them more precise, and work out new ones.

In this work (in contrast to [7]) we choose a rather constructive approach to introduce the needed structures. Not the more common and more known structures are defined first, but the more elementary ones. Most definitions will use the previous definitions, in order to have an exact specification for the defined structure. This results in the first part of this work being dedicated mainly to definitions.

In Sections 2.1 and 2.2 we will introduce the fundamental geometric and combinatorial objects that serve as building-blocks for later. In Section 2.3 we take points in an arbitrary-dimensional space and assign labels to them. Section 2.4 defines triangulations (and their generalizations: subdivisions) on the labeled points. And in the subsequent sections, structures and properties for triangulations and subdivisions will be shown.

At first we begin with a recapitulation of some fundamental geometric operations. For the following definitions, let X be a subset of $\mathbb{R}^m$.

Definition 2.0.1 (Linear span)**.**

$$\operatorname{span}(X) = \{\sum \lambda_i x_i | x_i \in X, \lambda_i \in \mathbb{R}\}$$

This is the intersection of all subspaces containing X and therefore the lowest-dimensional subspace of $\mathbb{R}^m$ containing X. Because it is a subspace, it also contains the origin.

Definition 2.0.2 (Affine hull)**.**

$$\operatorname{aff}(X) = \{\sum \lambda_i x_i | x_i \in X, \lambda_i \in \mathbb{R}, \sum \lambda_i = 1\}$$

This is the intersection of all affine spaces containing X and therefore the lowest-dimensional affine space of $\mathbb{R}^m$ containing X. It is a subset of the linear span.

Definition 2.0.3 (Conical hull)**.**

$$\operatorname{coni}(X) = \{\sum \lambda_i x_i | x_i \in X, \lambda_i \in \mathbb{R}, \lambda_i \geq 0\}$$

This is the intersection of all convex cones containing X. It is also a subset of the linear span.

Definition 2.0.4 (Convex hull)**.**

$$\text{conv}(X) = \{\sum \lambda_i x_i | x_i \in X, \lambda_i \in \mathbb{R}, \sum \lambda_i = 1, \lambda_i \geq 0\}$$

This is the intersection of all convex sets containing X. It is a subset of the linear span, the affine hull, and the conical hull.

A fundamental combinatorial structure that will be used is the graph. It consists of a set of objects (called the "vertices") together with a set of links (called the "edges") between the objects. An introduction to graph theory can be found in [5].

Definition 2.0.5 (Graph)(adapted from [5, p. 1], [5, p. 5])**.** An *undirected graph* is an ordered pair (V, E), where V is a set, and E is a set consisting of subsets of V with two elements. A *directed graph* is an ordered pair (V, E), where V is a set, and E is a set consisting of ordered pairs of V with two elements. The elements of V are called *vertices*, and the elements of E are called *edges*.

The difference between undirected and directed graphs is that the edges of directed graphs have a built-in direction. This also means that for two vertices in an undirected graph there is only one edge possible, whereas for two vertices in a directed graph there are two edges possible.

Another combinatorial structure that will be used repeatedly is the partially ordered set (also called "poset"). It consists of a set together with a binary relation. The binary relation determines how the elements of the set are related to each other (e.g. one element is "smaller" than another, or one element is subset of another, ...). For an overview of partially ordered sets see [18, chapter 3]. Posets can also be seen as directed graphs, since the binary relation can determine the directed edges.

Definition 2.0.6 (Partial order)(in analogy to [18, p. 97])**.** A *partial order* is a binary relation on a set that is reflexive, antisymmetric, and transitive.

Definition 2.0.7 (Poset)(in analogy to [18, p. 97])**.** A *poset* is a set together with a partial order on that set. A *maximal element* in a poset is an element that can be compared to every other element and is greater than every other element. Analog, a *minimal element* in a poset is an element that can be compared to every other element and is smaller than every other element.

2.1 Geometric objects

Triangulations are essentially geometric objects. In 2-dimensional space, one can think of triangulations as consisting of triangles. But the precise definition will also include the so-called faces of triangles. In higher dimensions the triangles are generalized to simplices. We will also define subdivisions, which can be seen as generalizations of triangulations. Subdivisions consist of polyhedra and not necessarily simplices. Overall, polyhedra are probably the most general and basic building-blocks of the structures we will define. A comprehensive introduction to polyhedra and related geometric objects can be found in [20].

Definition 2.1.1 (Polyhedron)([20, p. 4])**.** A *polyhedron* is the intersection of finitely many closed half-spaces in $\mathbb{R}^m$. The dimension d of the polyhedron is the dimension of the lowest-dimensional affine subspace, where the polyhedron is contained in (labeled by the notation: d-polyhedron).

By this definition a polyhedron is convex, because it is the intersection of half-spaces.

Definition 2.1.2 (Face (polyhedron))([7, p. 43])**.** A *face* F of a polyhedron P is every subset $F \subseteq P$ where an arbitrary linear functional ψ is maximized. We write $F \leq P$. Every face of a polyhedron is a polyhedron itself. We write *k-face* for a k-dimensional face. The empty set is also defined as a face of every polyhedron. Naming conventions for faces are:

- 0-faces are called *vertices*
- 1-faces are called *edges*
- $(d-1)$-faces are called *facets*
- the (-1)-face is the empty set and is called *empty face*
- the d-face is the polyhedron itself and is called *trivial face*

Definition 2.1.3 (Polytope)([20, p. 4])**.** A *polytope* is the convex hull of a finite set of points in $\mathbb{R}^m$.

Theorem 2.1.4 (see [20, p. 29])**.** A polytope is a bounded polyhedron.

The dimension d of a polytope is given by the dimension of the affine hull of its vertices. Faces of polytopes are polytopes themselves.

Definition 2.1.5 (Simplex)(see [20, p. 7])**.** A *simplex* is a d-polytope that has exactly $d+1$ vertices.

Each k-face of a simplex is a k-simplex itself and always has $k+1$ vertices. This implies that a d-simplex has the property that the convex hull of its vertices always spans a $(d-1)$-dimensional space, if one vertex is omitted. Because each subset of vertices spans a face, a d-simplex has $\binom{d+1}{k+1}$ k-faces (see [9, p. 45]).

Definition 2.1.6 (Face poset)(adapted from [20, p. 57])**.** The *face poset* of a polyhedron is the poset of it's faces with respect to the subset relation.

Note that in literature the face poset is generally called face lattice. A lattice is a special kind of poset that fulfills additional properties. But for this work the definition of a poset is sufficient.

Definition 2.1.7 (Geometric polyhedral complex)(adapted from [7, p. 45])**.** A *geometric polyhedral complex* $\mathscr{C}$ is a set of polyhedra where each face of a polyhedron is also element of the set, and the intersection of two polyhedra is a face of both. Formally:

- $F \leq P \in \mathscr{C} \Rightarrow F \in \mathscr{C}$ (Closure Property)
- $P_1, P_2 \in \mathscr{C} \Rightarrow P_1 \cap P_2 \leq P_1, P_2$ (Intersection Property)

Definition 2.1.8 (Geometric simplicial complex)([7, p. 45])**.** A *geometric simplicial complex* is a polyhedral complex consisting of only simplices.

2.2 Combinatorial objects

The geometric objects defined in the last section might be sufficient for describing triangulations, but not for describing subdivisions. Triangulations and subdivisions are not only geometric objects, but also combinatorial objects. After having introduced geometric complexes, we now define complexes which are of combinatorial nature:

Definition 2.2.1 (Abstract complex)**.** For a set of labels $\mathcal{L}$, an *abstract complex* $\mathscr{A}$ is a set consisting of subsets of $\mathcal{L}$, with the property that the intersection of elements of $\mathscr{A}$ is contained in $\mathscr{A}$. Formally:

- $S_1, S_2 \in \mathscr{A} \Rightarrow S_1 \cap S_2 \in \mathscr{A}$ (Abstract Intersection Property)

Note that the empty set is per definition contained in the abstract complex.

Definition 2.2.2 (Abstract simplicial complex)([7, p. 83])**.** For a set of labels $\mathcal{L}$, an *abstract simplicial complex* $\mathscr{A}$ is a set of subsets of $\mathcal{L}$ where each subset of elements of $\mathscr{A}$ is in $\mathscr{A}$. Formally:

- $S_{sub} \subseteq S \in \mathscr{A} \Rightarrow S_{sub} \in \mathscr{A}$

It is straightforward to show that an abstract simplicial complex is an abstract complex, the property of the former implies the property of the latter.

2.3 Point configurations

Every triangulation is defined on a set of points. When working with point sets, it is convenient to have labeled points, and maybe to allow having multiple points on the same location. This also helps when we reference a point. Therefore, we take a set of labels and define a point-configuration as follows:

Definition 2.3.1 (Point-configuration)([7, p. 47])**.** A *point-configuration* $\mathcal{A}$ is a function $\mathcal{L} \to \mathbb{R}^m$ where $\mathcal{L}$ is a set of labels and $\mathbb{R}^m$ is the space of coordinates. A point is the tuple $(l \in \mathcal{L}, \mathcal{A}(l) \in \mathbb{R}^m)$, the label of the point is l, and the coordinates of the point are $\mathcal{A}(l)$. The dimension d of the point-configuration is the dimension of the affine hull of the points (written as d-dimensional point-configuration).

For convenience we sometimes refer to a point, but mean its label or its coordinates, according to the situation. Equally, we may refer to a point-configuration, but mean the set of labels or the set of coordinates of its points.

Point-configurations will always be considered in "general position". That means that for a d-dimensional point-configuration no more than $k + 1$ points lie on a k-dimensional hyperplane for $0 \leq k \leq d - 1$. This simplifies some problems and eliminates many special cases.

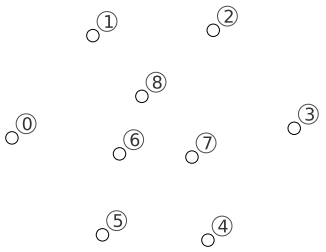


Figure 1: *Point-configuration.* 9 points in the plane (with their labels).

There is also a kind of generalization for the concept of a point-configuration, namely the vector-configuration (see [7, p. 77]). There, one simply has vectors instead of points and triangulations are still possible. The corresponding object to a triangle defined by 3 points in a point-configuration is a cone, defined by 3 vectors in the vector-configuration. If one can lay a half-plane through the origin and all vectors are on one side of it, one can offset the plane a bit into this direction (where the vectors are) and stretch each vector so that its head lies in the plane. This way, one gets a point-configuration (from the vector heads) and a set of triangles (corresponding to the cones) that form a triangulation. This explains why vector-configurations are a generalization of point-configurations. However,

for the scope of this work, point-configurations are sufficient and we will only deal with them.

After having defined geometric and combinatorial objects, we now bring them both together in the definition of cells of a point-configuration. The geometric part of a cell works with the points of the point-configuration, and the combinatorial part works with the labels of the point-configuration.

Definition 2.3.2 (Cell)**.** We define a *cell* on a point-configuration $\mathcal{A}$ as a tuple $(l \subseteq \mathcal{L}, P \subset \mathbb{R}^m)$ where $P = conv(\mathcal{A}(l))$. We call l the combinatorial part and P the geometric part of the cell. The dimension k of a cell is the dimension of its convex hull, and the cell is called *k-cell*. The trivial cell consists of all labels and their convex hull. Naming conventions for cells are:

- 0-cells are called *vertices*
- 1-cells are called *edges*
- d-cells are called *maximal cells*
- the cell defined by the empty set is the (-1)-cell and is called *empty cell*
- the cell defined by all labels of $\mathcal{A}$ is called *trivial cell*

A k-cell is a *simplicial cell*, if it has exactly $k+1$ labels and its convex hull is a simplex. We say that a cell C_a is a *subcell* of another cell C_b (written as $C_a \subseteq C_b$), if the labels of C_a form a subset of the labels of C_b.

As with point-configurations, we may in future refer to a cell and mean its labels, its convex hull, or the points corresponding to its labels. Note that a cell can contain labels of points that are not vertices of the cell's convex hull. The points can even lie in the relative interior of the convex hull.

For a cell C of a point-configuration $\mathcal{A}$ we write $\mathcal{A}_C$ for the point-configuration consisting only of the points of C.

Definition 2.3.3 (Face (cell))**.** A cell $F = (l_F, P_F)$ is a *face* of another cell $C = (l_C, P_C)$ (written as $F \leq C$), if:

- the geometric part of F is a face of the geometric part of C
- the combinatorial part of F consists of all labels of C whose points lie in the geometric part of F

Formally:

$$F \leq C \iff P_F \leq P_C \wedge l_F = \{l \in l_C | \mathcal{A}(l) \in P_F\} \tag{1}$$

The combinatorial part of the definition is crucial: Given a face of a cell, a label of that cell, which lies in the convex hull of the face, must be part of the face. The label cannot be omitted, even if it is not a vertex of the face's convex hull.

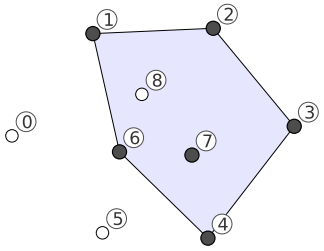


Figure 2: *Cell.* A cell of the point-configuration in Figure 1. The cell consists of the labels {1,2,3,4,6,7}. The convex hull is shown in blue, and the points corresponding to the labels of the cell are marked in black. Note that point 8 lies inside the convex hull of the cell, but is not part of the cell.

2.4 Triangulations and subdivisions

We come to the formal definition of a triangulation. But first we will define the more general concept of a subdivision. In contrast to a triangulation which consists exclusively of simplices, a subdivision consists of convex polyhedra. Since every simplex is a polyhedron, every triangulation is also a subdivision.

Because we have labels and coordinates in a point-configuration, we can bring geometric and abstract complexes together in the formal definition of subdivisions of point-configurations. The already defined cells are acting as building-blocks of a subdivision. The labels of the cells are used to ensure that a subdivision fulfills the properties of an abstract complex, and the convex hulls of the cells are used to ensure that the properties of a geometric complex are fulfilled.

Definition 2.4.1 (Polyhedral subdivision)(adapted from [7, p. 62])**.** A *polyhedral subdivision* $\mathscr{S}$ of a point-configuration $\mathcal{A}$ is a set of cells with the following properties: each face of a cell is contained in $\mathscr{S}$; two cells intersect in a common face; the union of all cells equals the convex hull of $\mathcal{A}$. Formally:

- $F \leq C \in \mathscr{S} \Rightarrow F \in \mathscr{S}$ (Closure Property)
- $C_1, C_2 \in \mathscr{S} \Rightarrow C_1 \cap C_2 \leq C_1, C_2$ (Intersection Property)

- $\bigcup_{C \in \mathscr{S}} C = \text{conv}(\mathcal{A})$ (Union Property)

If all labels of $\mathcal{A}$ are used, the subdivision is called *full*. If the subdivision consists only of the trivial cell and its faces, the subdivision is itself called trivial subdivision. The set of all polyhedral subdivisions of $\mathcal{A}$ is denoted $\mathit{Subdivisons}(\mathcal{A})$.

With this definition, a polyhedral subdivision is a geometric polyhedral complex on the coordinates of the point-configuration and an abstract complex on the labels. Note that the polyhedra of the polyhedral subdivision are polytopes, since they are bounded by the point-configuration. From now on we will call a polyhedral subdivision just "subdivision".

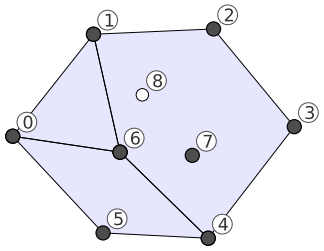


Figure 3: *Polyhedral subdivision.* A polyhedral subdivision of the point-configuration in Figure 1. The cells must cover the whole convex hull. Only point 8 is not part of the subdivision.

Now we finally come to the definition of a triangulation, which is, as already mentioned, a special kind of subdivision.

Definition 2.4.2 (Triangulation (simplicial subdivision))(in analogy to [7, p. 1])**.** A *triangulation* (or *simplicial subdivision*) is a polyhedral subdivision where each cell is a simplicial cell. The set of all triangulations of a point-configuration $\mathcal{A}$ is denoted $\mathit{Triangulations}(\mathcal{A})$.

Note that the word "triangulation" is usually used for the 2-dimensional case, because the highest-dimensional simplices are triangles. But "triangulation" is also used for an arbitrary-dimensional simplicial subdivision (often called d-dimensional-triangulation).

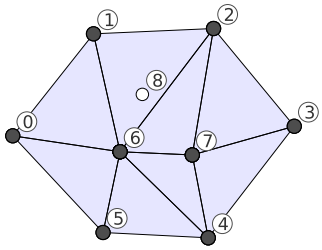


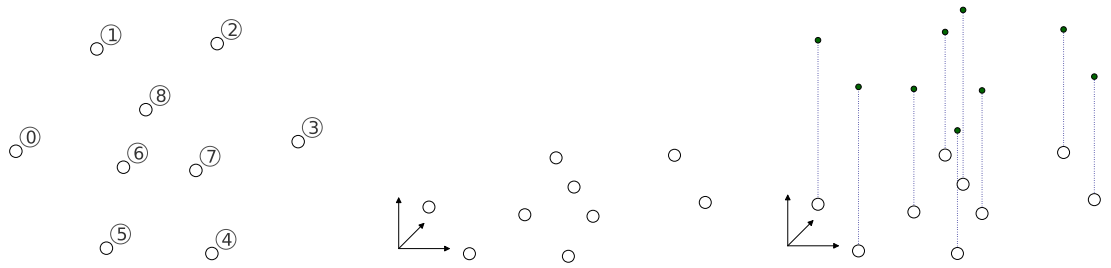
Figure 4: *Triangulation.* A triangulation of the point-configuration in Figure 1. Each cell is a triangle (in general a simplex). Point 8 is again not part of it.

2.5 Height-functions for subdivisions

A concept which has proven to be very helpful in analyzing subdivisions and especially triangulations is that of a height-function for a point-configuration. Each of the points gets assigned a height and will be lifted according to this height. This will lead to the important classifications of regular triangulations and regular subdivisions.

Definition 2.5.1 (Height-function)([7, p. 55])**.** A *height-function* for a point-configuration $\mathcal{A}$ is a function: $w : \mathcal{A} \to \mathbb{R}$.

Definition 2.5.2 (Lifted point-configuration)(partially from [7, p. 55])**.** The *lifted point-configuration* for a point-configuration $\mathcal{A}$, lifted by a height-function w, is the function: $\mathcal{A}^w : \mathcal{L} \to \mathbb{R}^{m+1}$ with $l \mapsto \begin{pmatrix} \mathcal{A}(l) \\ w(l) \end{pmatrix}$.



(a) Original point-configuration

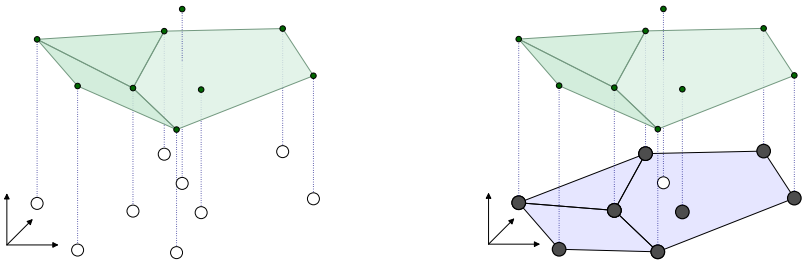
(b) Point-configuration in xy-plane of $\mathbb{R}^3$

(c) Lifted point-configuration in $\mathbb{R}^3$

Figure 5: *Lifted point configuration.* A lifting of the point-configuration in Figure 1. Each point is associated with a certain height. A lifted point has the coordinates of the original point, plus the height as the last coordinate.

Having the points lifted along an additional dimension via the height-function w gives us an interesting possibility: We can compute the convex hull of the lifted point-configuration. Now we are interested in the lower part of the boundary of the convex hull. To be exact, we look at the trivial cell of the lifted point-configuration (it corresponds to the convex hull). Now we take the lower faces of the trivial cell (That are those which intersect a line parallel to the new dimension-axis lower than any other face). When we project these lower faces back to $\mathbb{R}^m$, we get a subdivision of the point-configuration.

Definition 2.5.3 (Subdivision-function)(adapted from [7, p. 60])**.** We define the *subdivision-function* $\text{sub}_{\mathcal{A}}(w)$ for a point-configuration $\mathcal{A}$ and a given height-function w as the set of cells, which is obtained by projecting down all lower faces of the trivial cell of $\mathcal{A}^w$ from $\mathbb{R}^{m+1}$ to $\mathbb{R}^m$ by the first m coordinates.
We also define the *constrained subdivision-function* $\text{sub}_{\mathcal{A}}(w, C)$, constrained for a cell C. $\text{sub}_{\mathcal{A}}(w, C) := \text{sub}_{\mathcal{A}_C}(w)$.
At last, we define the *constrained subdivision-function* $\text{sub}_{\mathcal{A}}(w, \mathscr{S}_0)$, constrained for a subdivision $\mathscr{S}_0$. $\text{sub}_{\mathcal{A}}(w, \mathscr{S}_0)$ is the set consisting of all cells from $\text{sub}_{\mathcal{A}}(w, C_0)$ for every cell $C_0 \in \mathscr{S}_0$.



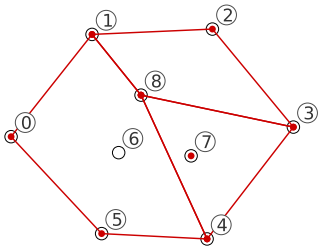
(a) Lower convex hull.

(b) Subdivision obtained by projection of lower convex hull.

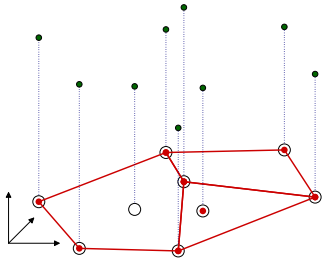
Figure 6: *Subdivision-function.* The subdivision-function for the lifted point-configuration in Figure 1. The lower convex hull of the lifted points is projected back down from $\mathbb{R}^3$ to $\mathbb{R}^2$.

The constrained version of the subdivision-function is an invention of this work. It will later on be needed to further relate subdivisions to each other. The constrained version can be seen as the union of normal subdivision-functions for subsets of the point-configuration. That the set obtained from the subdivision-function is indeed a subdivision of the point-configuration is proven in [7, p.67]. That this is also true for the constrained version is obvious, since the set obtained from the constrained

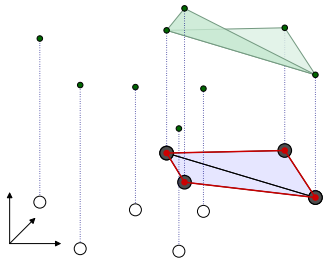
subdivision-function consists of sets that are obtained from a normal subdivision-function.



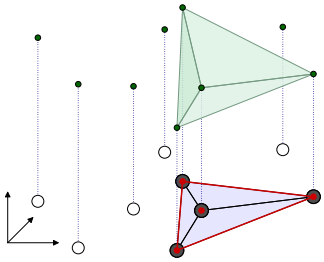
(a) Constraining subdivision.



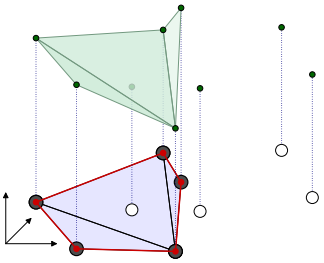
(b) Constraining subdivision and lifting of points.



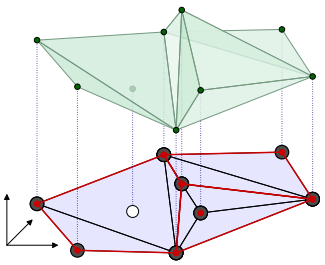
(c) Result of subdivision-function for constraining cell {1,2,3,8}.



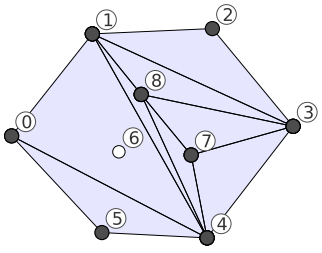
(d) Result of subdivision-function for constraining cell {3,4,7,8}.



(e) Result of subdivision-function for constraining cell {0,1,4,5,8}.



(f) Result of subdivision-function for the constraining subdivision in Figure 7a.



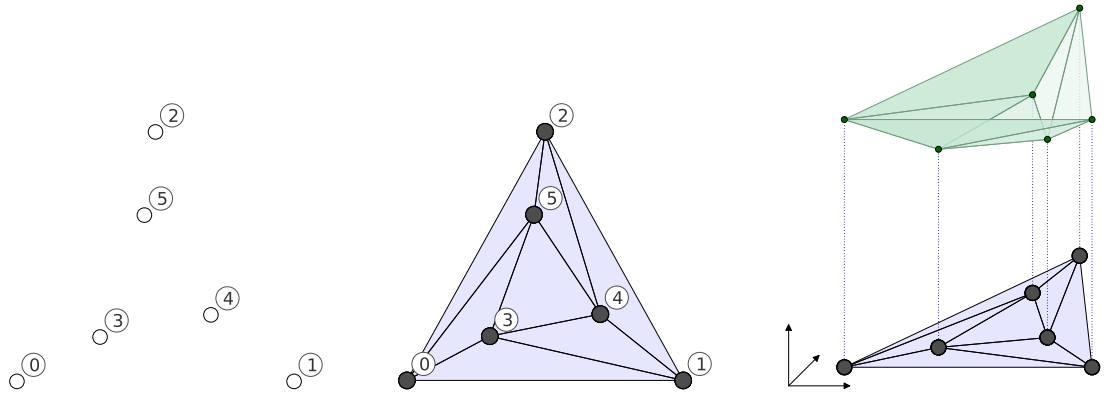
(g) Resulting subdivision.

Figure 7: *Constrained subdivision-function.* Constrained subdivision-function for the point-configuration in Figure 1. It is constrained for the subdivision in Figure 7a. According to the 3 cells of the constraining subdivision, there are 3 lower convex hulls as shown in Figure 7c, Figure 7d, and Figure 7e. The resulting subdivision for the constrained subdivision-function is shown in Figure 7g.

2.6 Regular subdivisions

Every subdivision that can be obtained from the subdivision-function is called regular. Regularity is a fruitful concept, because a large amount of analysis can be done with the height-functions that generate subdivisions. Whereas when a subdivision is not regular, the possibilities for analyzing it are limited.

Definition 2.6.1 (Regular subdivision)([7, p. 59])**.** A subdivision $\mathscr{S}$ of a point-configuration $\mathcal{A}$ is called *regular*, if there exists a height-function w, such that $\mathscr{S} = \mathrm{sub}_{\mathcal{A}}(w)$.



(a) Point configuration **(b)** Regular triangulation **(c)** Subdivision function

Figure 8: *Regular subdivision.* Regular subdivision in form of a triangulation (Figure 8b), for the point-configuration in Figure 8a. It can be obtained via the subdivision-function used on a height-function (Figure 8c). In this case, the height-function lifts all outer points to a common height and all inner points to a second, lower situated common height.

The existence of regular subdivisions is clear, because for any arbitrary height-function the subdivision-function already yields a regular subdivision. That there exist non-regular subdivisions is not immediately obvious. For a subdivision to be non-regular, one has to proof that there exists no height-function, such that the subdivision-function yields this particular subdivision. Examples will be given later on.

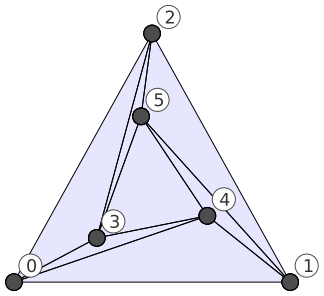
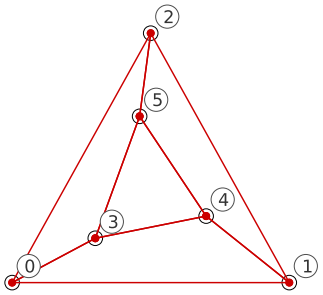


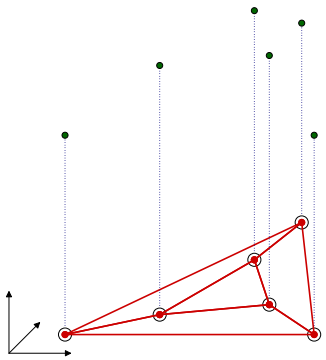
Figure 9: *Non-regular subdivision.* Non-regular subdivision of the point-configuration in Figure 8a. Here in form of a triangulation.

If a subdivision $\mathscr{S}$ is not regular there is no height-function such that the subdivision-function yields $\mathscr{S}$. But there can still be a height-function and a constraining subdivision such that the constraining subdivision-function yields $\mathscr{S}$. Therefore, $\mathscr{S}$ can still be kind of pseudo-regular in relation to a constraining subdivision.

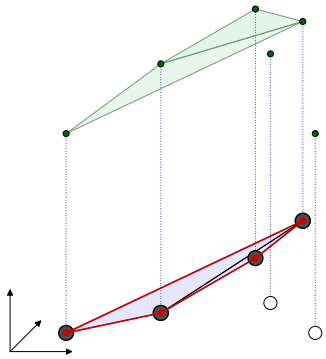
Definition 2.6.2 (Relative regular subdivision)**.** We call a subdivision $\mathscr{S}$ of a point-configuration $\mathcal{A}$ *relative regular* to another subdivision $\mathscr{S}_0$, if there exists a height-function w such that $\mathscr{S} = \mathrm{sub}_{\mathcal{A}}(w, \mathscr{S}_0)$.



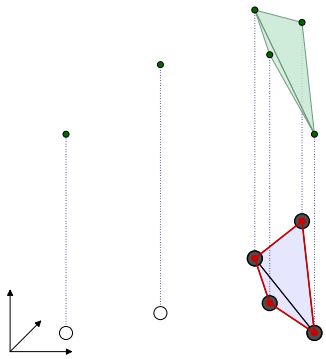
(a) Constraining subdivision.



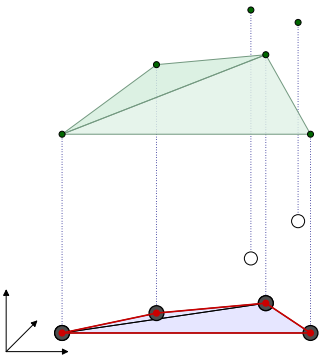
(b) Constraining subdivision and lifting of points.



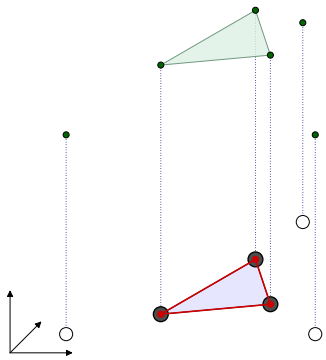
(c) Result of subdivision-function for constraining cell {0,2,3,5}.



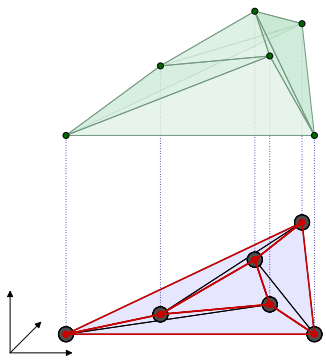
(d) Result of subdivision-function for constraining cell {1,2,4,5}.



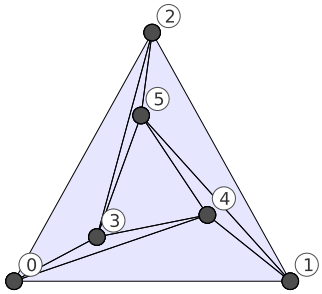
(e) Result of subdivision-function for constraining cell {0,1,3,4}.



(f) Result of subdivision-function for constraining cell {3,4,5}.



(g) Result of subdivision-function for the constraining subdivision in Figure 10a.



(h) Resulting subdivision.

Figure 10: *Relative regular subdivision.* Relative regular subdivision (Figure 10h) for the point-configuration in Figure 8a, relative regular to the subdivision in Figure 10a.

A subcategory of non-regular subdivisions are subdivisions that are cyclic with respect to the in-front-relation. Edelsbrunner has described the principle of acyclicity with respect to the in-front-relation in [8] and [9, p. 4].

Definition 2.6.3 (In-front-relation)(adapted from [9, p. 4])**.** Fix maximal cells C_a, C_b of a point-configuration $\mathcal{A} \subset \mathbb{R}^m$ and $x \in \mathbb{R}^m$. The *in-front-relation* states that C_a is in-front-of C_b with respect to x, if a half-line starting at x intersects C_a before C_b or simultaneously.

Note that this definition is well-defined. If $C_a \neq C_b$, there cannot be two half-lines staring at x, where one half-line intersects C_a before C_b, and the other half-line intersects C_b before C_a.

Theorem 2.6.4. The in-front-relation for a fixed $x \in \mathbb{R}^m$ is a reflexive, antisymmetric, and transitive binary relation on the set of all cells of a subdivision. The in-front-relation for $x \in \mathbb{R}^m$ is therefore a partial order.

Proof. trivial □

As in every partial order, in a subdivision's set of cells there can be cycles with respect to the in-front-relation. If such a cycle appears for a subdivision, the subdivision is not regular:

Definition 2.6.5 (In-front-relation cyclic subdivision)**.** A subdivision is called *cyclic with respect to the in-front-relation*, if there exists $x \in \mathbb{R}^m$, such that the in-front-relation regarding x contains a cycle.

Theorem 2.6.6. A subdivision that is cyclic with respect to the in-front-relation is not regular.

Proof. Take a subdivision $\mathscr{S}$ which is cyclic with respect to the in-front-relation, and fix $x \in \mathbb{R}^m$ for which $\mathscr{S}$ is cyclic. Now assume that $\mathscr{S}$ is regular. This means there exists a height-function (for which $\mathscr{S}$ is regular) and a corresponding lifted point-configuration. Each maximal cell C of $\mathscr{S}$ corresponds to a facet F of the convex hull of the lifted point-configuration. Consider a hyperplane through F. The hyperplane intersects a vertical line (parallel to the lifting-axis) through x in a specific height. Now take two maximal cells C_a, C_b, where C_a is in-front-of C_b. They have a common face C_{ab}, and corresponding faces F_a, F_b, and F_{ab} on the convex hull of the lifted point-configuration. Because the region around F_{ab} on the convex hull must be convex, the hyperplane trough F_a must intersect the vertical line through x in a higher position than the hyperplane trough F_b. Now take a sequence of cells C_0, C_1, …, C_k that is cyclic with respect to the in-front-relation. The intersection heights of the vertical line through x with the hyperplanes corresponding to C_0, C_1, …, C_k must now also be cyclic. But the

heights of these intersections are increasing values in $\mathbb{R}$ and can therefore not be cyclic, so this is a contradiction. □

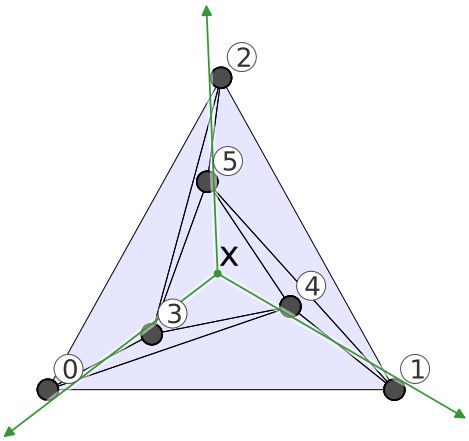


Figure 11: *In-front-relation cyclic subdivision.* The triangulation in Figure 9 is cyclic with respect to the in-front-relation. For x there is a cycle for the set consisting of all triangles except the middle one. The intersections of half-lines starting from x with the triangles determine the in-front-relation between the triangles. The 3 half-lines shown in the picture are sufficient to achieve a cycle with respect to the in-front-relation for the triangles.

2.7 Delaunay triangulations and subdivisions

We give a short overview of a very special triangulation. The Delaunay triangulation is in some sense the most natural triangulation (see [10, 3]). It exists for every point-configuration and has special properties. In the already mentioned problem of finding flip-paths, a common method is to flip towards the Delaunay triangulation (see [9, p. 13]). We specify it here for subdivisions, since they are the generalizations of triangulations. The Delaunay subdivision can be defined as the result of the subdivision-function when the points are lifted to the $(m+1)$-dimensional paraboloid:

Definition 2.7.1 (Delaunay subdivision (height-function definition))(see [3])**.** The *Delaunay subdivision* is the subdivision $\text{sub}_{\mathcal{A}}(w)$ for the height-function $w : \mathcal{A} \to \mathbb{R}, p \mapsto \|p\|^2$

Note that this subdivision does not change when the point-configuration is rotated or translated. The height-function does change, but the convex hull in $\mathbb{R}^{m+1}$ stays

combinatorial the same. One can see by the height-function, that the points are lifted onto the $(m+1)$-dimensional paraboloid. The paraboloid has the property that points in $\mathbb{R}^m$ lying on a common sphere (circle in 2 dimensions) lie on a common hyperplane in $\mathbb{R}^{m+1}$. If no other point is lying inside the sphere, the hyperplane will contain a facet of the convex hull in $\mathbb{R}^{m+1}$. This gives a second equivalent definition:

Definition 2.7.2 (Delaunay subdivision (empty sphere property))(see [3])**.** A *Delaunay subdivision* is a subdivision where each maximal cell has the following properties:

- there exists a sphere through all its vertices
- no point of the point-configuration lies in the interior of the sphere

Normally, one speaks of Delaunay triangulations (and not Delaunay subdivisions), because in general position no more than $m+1$ points lie on a $(m-1)$-sphere and they define a simplex. So if the points are in general position, the subdivision-function will yield a triangulation. The second definition is the usual one, but since we will work extensively with height functions, the first definition is adequate for us. It also shows implicitly that the Delaunay subdivision is regular. There is also a modified version of the Delaunay subdivision that adds weights to the height-function:

Definition 2.7.3 (Weighted Delaunay subdivision)**.** The *weighted Delaunay subdivision* is the subdivision $\text{sub}_{\mathcal{A}}(w)$ for the height-function $w : \mathcal{A} \to \mathbb{R}, p \mapsto \|p\|^2 - \overline{w}(p)$, where the weight-function $\overline{w}$ is any function $\mathcal{A} \to \mathbb{R}$ defining a weight for each point.

If we start with all weights being 0, we get the normal Delaunay subdivision. Modifying the weights a little bit will change the simplices in some regions. This way we can get from the Delaunay subdivision to other subdivisions. In fact we can reach every other regular subdivision, since each height-function can be generated. So the weighted Delaunay subdivisions are also the regular ones, but - so to say - viewed from a different angle.

Delaunay subdivisions also have a tight relation to the widely used Voronoi diagrams (see [1, 2] for introductions to Voronoi diagrams). Delaunay subdivisions and Voronoi diagrams are the dual graphs of each other.

2.8 Refinements

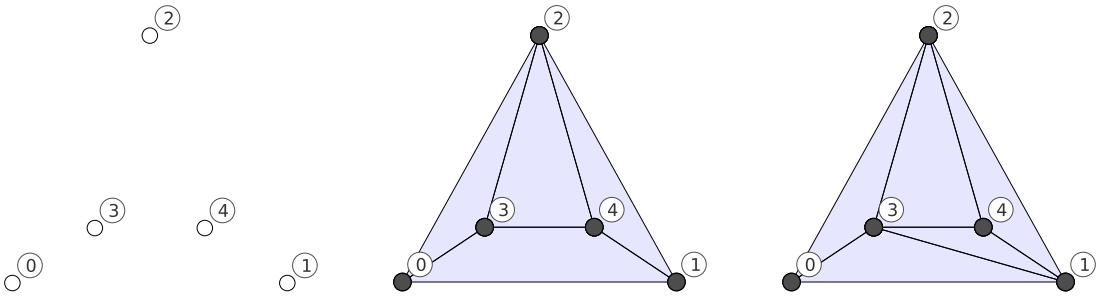
There are more interesting properties of subdivisions than just the one that subdivisions are generalizations of triangulations. A special relation between subdivisions exists (which includes the triangulations). If we insert an edge (that crosses

no other edge) into a 2-dimensional subdivision, we get another subdivision. This subdivision is in a way finer than the original one, because it has more polyhedra. Of course, it also works the other way. If we take out an edge of a 2-dimensional subdivision, and the new resulting area is a convex polyhedron, we get another subdivision. Now the new subdivision is coarser than the original one, because it has less polyhedra. The exact definition is the following one:

Definition 2.8.1 (Refinement and coarsening)(adapted from [7, p. 65])**.** Given two subdivisions $\mathscr{S}_{coarse}$ and $\mathscr{S}_{fine}$ of the same point-configuration, it is said that $\mathscr{S}_{coarse}$ is a *coarsening* of $\mathscr{S}_{fine}$ and $\mathscr{S}_{fine}$ is a *refinement* of $\mathscr{S}_{coarse}$, if each cell of $\mathscr{S}_{fine}$ is a subcell of a cell of $\mathscr{S}_{coarse}$. Formally:

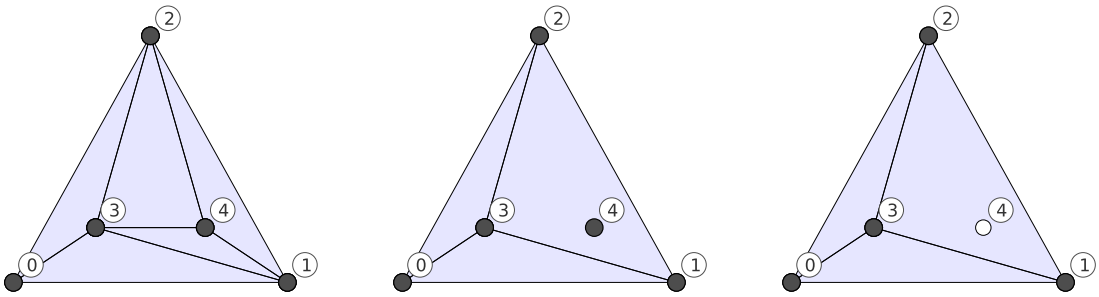
- $C_{fine} \in \mathscr{S}_{fine} \Rightarrow \exists C_{coarse} \in \mathscr{S}_{coarse} : C_{fine} \subseteq C_{coarse}$

We write $\mathscr{S}_{fine} \preceq \mathscr{S}_{coarse}$ (or $\mathscr{S}_{fine} \prec \mathscr{S}_{coarse}$ if they are not allowed to be equal). For a subdivision $\mathscr{S}_0$, the set of all refinements is written $\mathit{Refinements}(\mathscr{S}_0)$, the set of all coarsenings is written $\mathit{Coarsenings}(\mathscr{S}_0)$, and the set of all triangulations that are refinements is written $\mathit{Triangulations}(\mathscr{S}_0)$.



(a) Point configuration **(b)** Coarse subdivision **(c)** Fine subdivision

Figure 12: *Refinement subdivision and coarsening subdivision.* A point configuration and two subdivisions of it. Figure 12b is a coarsening of Figure 12c, and Figure 12c is a refinement of Figure 12b.



(a) Triangulation **(b)** Coarsening of Figure 13a **(c)** Not a coarsening of Figure 13a

Figure 13: *Refinement example.* Three subdivisions of the point-configuration in Figure 12a. Figure 13b is a coarsening of Figure 13a, but Figure 13c is not, because vertex 4 is not part of a cell and can therefore not be part of any cell of a refining subdivision.

Since we can compare two subdivisions whether one is a refinement of the other, we have a relation in the mathematical sense. This relation can be used to build a partially ordered set:

Theorem 2.8.2 ([7, p. 66])**.** For a point-configuration $\mathcal{A}$, the refinement relation gives a poset on $\mathit{Subdivisons}(\mathcal{A})$.

Definition 2.8.3 (Refinement poset)([7, p. 65])**.** The *refinement poset* of a point-configuration $\mathcal{A}$ is the poset obtained from $\mathit{Subdivisons}(\mathcal{A})$ via refinement relation. For a subdivision $\mathscr{S}_0$, the *constrained refinement poset* is the poset obtained from $\mathit{Refinements}(\mathscr{S}_0)$. The *height* of a refinement poset is the maximum length k of a chain of subdivisions $\mathscr{S}'_0 \prec \mathscr{S}'_1 \prec \ldots \prec \mathscr{S}'_k$ in the poset.

Note that for a subdivision $\mathscr{S}_0$, it holds that $\mathscr{S}_0 \in \mathit{Refinements}(\mathscr{S}_0)$ and $\mathscr{S}_0 \in \mathit{Coarsenings}(\mathscr{S}_0)$. For a point-configuration $\mathcal{A}$, the refinement poset equals the constrained refinement poset for the trivial subdivision of $\mathcal{A}$.

Theorem 2.8.4 (in analogy to [7, p. 66])**.** The refinement poset has a unique maximal element (the trivial subdivision), and every minimal element is a triangulation.

Definition 2.8.5 (Minimal refinement and coarsening)**.** A *minimal refinement* of a subdivision $\mathscr{S}$ is a subdivision $\mathscr{S}_{\mathit{fine}}$ so that there is no subdivision $\mathscr{S}'$ with $\mathscr{S}_{\mathit{fine}} \prec \mathscr{S}' \prec \mathscr{S}$. A *minimal coarsening* of a subdivision $\mathscr{S}$ is a subdivision $\mathscr{S}_{\mathit{coarse}}$ so that there is no subdivision $\mathscr{S}'$ with $\mathscr{S} \prec \mathscr{S}' \prec \mathscr{S}_{\mathit{coarse}}$.

For a poset there is the nice possibility to visualize it as a directed graph. This graph is shown in Figure 14 for the point-configuration in Figure 12a.

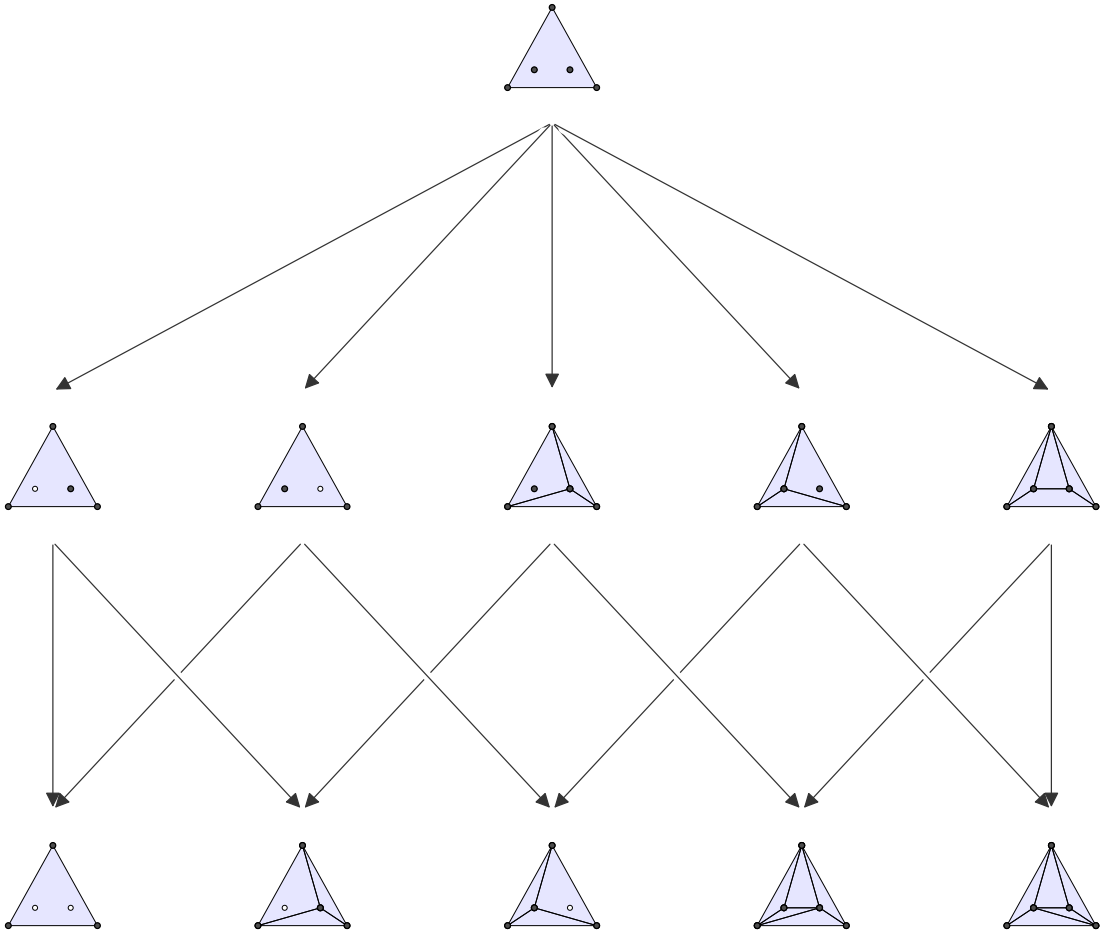


Figure 14: *Refinement poset.* Refinement poset for the point-configuration in Figure 12a. If $\mathscr{S}_{fine}$ is a minimal refinement of $\mathscr{S}_{coarse}$, there is an arrow drawn from $\mathscr{S}_{coarse}$ to $\mathscr{S}_{fine}$. In the first row is the trivial subdivision, and the last row consists of exclusively triangulations.

Theorem 2.8.6 (Relative regularity of refinements)**.** For subdivisions $\mathscr{S}_{fine} \preceq \mathscr{S}_{middle} \preceq \mathscr{S}_{coarse}$ follows:

1. If $\mathscr{S}_{fine}$ is relative regular to $\mathscr{S}_{coarse}$, then $\mathscr{S}_{fine}$ is relative regular to $\mathscr{S}_{middle}$.
2. If $\mathscr{S}_{fine}$ is not relative regular to $\mathscr{S}_{middle}$, then $\mathscr{S}_{fine}$ is not relative regular to $\mathscr{S}_{coarse}$.

Proof.

1. There exists a height-function w such that $\mathscr{S}_{fine} = \mathrm{sub}_{\mathcal{A}}(w, \mathscr{S}_{coarse})$. We will show that $\mathscr{S}_{fine} = \mathrm{sub}_{\mathcal{A}}(w, \mathscr{S}_{middle})$.

   Because of $\mathscr{S}_{fine} \preceq \mathscr{S}_{middle} \preceq \mathscr{S}_{coarse}$, we can take cells $C_{fine} \subseteq C_{middle} \subseteq C_{coarse}$ in the corresponding subdivisions.

   From $\mathscr{S}_{fine} = \mathrm{sub}_{\mathcal{A}}(w, \mathscr{S}_{coarse})$ follows $C_{fine} \in \mathrm{sub}_{\mathcal{A}}(w, C_{coarse})$.

   Consider the convex hull of the lifted point-configuration $\mathcal{A}^w_{C_{coarse}}$. Because of $C_{fine} \in \mathrm{sub}_{\mathcal{A}}(w, C_{coarse})$, C_{fine} has a corresponding face F_{fine} on the convex hull. Because C_{middle} is convex and a subset of C_{coarse}, F_{fine} must also be part of the convex hull of the lifted point-configuration $\mathcal{A}^w_{C_{middle}}$. It follows that $C_{fine} \in \mathrm{sub}_{\mathcal{A}}(w, C_{middle})$.

   Because C_{fine} can be chosen as an arbitrary subcell of C_{middle}, it follows that $\mathrm{sub}_{\mathcal{A}}(w, C_{middle}) \subseteq \mathrm{sub}_{\mathcal{A}}(w, C_{coarse})$.

   Because C_{middle} can be chosen as an arbitrary subcell of C_{coarse}, and C_{coarse} can be chosen arbitrary, it follows that $\mathrm{sub}_{\mathcal{A}}(w, \mathscr{S}_{middle}) = \mathrm{sub}_{\mathcal{A}}(w, \mathscr{S}_{coarse}) = \mathscr{S}_{fine}$.

2. This statement is equivalent to 1.

□

2.9 Flips

After the definition of triangulations, flips are the next central elements. A flip exchanges a part of a subdivision. In its most simple occurrence this can be the exchange of an edge with another edge in 2 dimensions. But it can also take out or insert a vertex from or into a subdivision. In higher dimensions there are even more possibilities. There is a survey of flips in planar graphs in [6].

If we think only about the simplices of a subdivision which are involved in the flip, leaving the other simplices aside, we find a surprisingly simple definition for a flip:

Definition 2.9.1 (Flip)(adapted from [7, p. 75])**.** For a point-configuration $\mathcal{A}$ take a subset $\mathcal{B} \subseteq \mathcal{A}$. If there exist exactly two triangulations $\mathscr{T}_a$, $\mathscr{T}_b$ for $\mathcal{B}$, we define a *flip* on $\mathcal{B}$ as the exchange of $\mathscr{T}_a$ with $\mathscr{T}_b$. The *reverse-flip* to the flip is the exchange of $\mathscr{T}_b$ with $\mathscr{T}_a$.

This gives us a simple description for flips in all dimensions. But flips can also be characterized. For instance, in 2 dimensions there is one edge exchanging flip, one vertex removal flip, and one vertex insertion flip. However, it turns out that looking at the highest-dimensional simplices gives a more general description of flips. In an edge exchanging flip one switches 2 triangles into 2 other triangles. So it is called a 2-2 flip. The vertex removal flip exchanges 3 triangles for 1 triangle. So it is called 3-1 flip. And the reversal, the vertex insertion flip, exchanges 1 triangle for 3 triangles. So it is called 1-3 flip. The same is possible for all other dimensions. For a more comprehensive and more detailed categorization of flips see [7, p. 119].

If we take $d+1$ affinely independent points and add one point in the affine subspace of the points, we get $d+2$ points for which exactly two d-dimensional triangulations are possible. It turns out that the flip which exchanges the 2 triangulations is always an a-b flip with $a \geq 1, b \geq 1$ and $a + b = d + 2$. The reverse-flip to an a-b flip is therefore always a b-a flip.

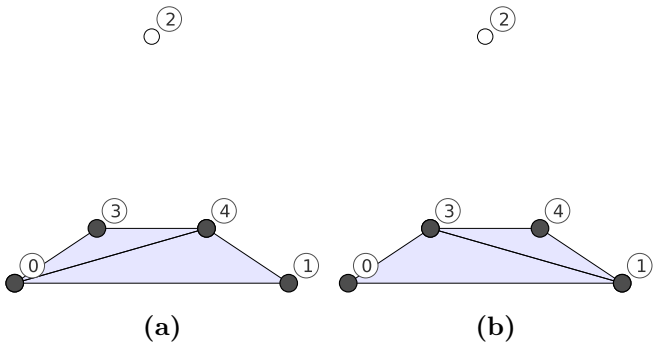


Figure 15: *Flip (edge exchange).* Edge flip in the point-configuration in Figure 12a. The 2 cells of Figure 15a are flipped into the 2 cells of Figure 15b. It corresponds to exchanging the edge (0,4) with the edge (1,3).

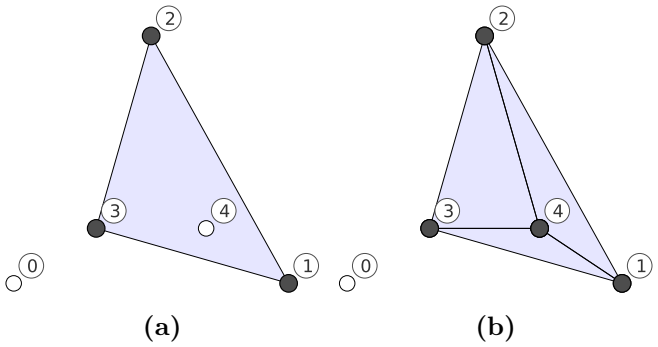


Figure 16: *Flip (vertex removal/vertex insertion).* Vertex removal and vertex insertion flip in the point-configuration in Figure 12a. The cell of Figure 16a is flipped into the 3 cells of Figure 16b. This corresponds to inserting the vertex 4. The reversal corresponds to removing the vertex 4.

Having described the flip only for a subset of a subdivision, we now say that we flip a subdivision $\mathscr{S}$ into another subdivision $\mathscr{S}'$, if we perform a flip on a subset of the points of $\mathscr{S}$ and obtain $\mathscr{S}'$. However, this is only possible when the triangulation in which the flip takes place (defined on the subset of points) is part of the subdivision $\mathscr{S}$. $\mathscr{S}$ and $\mathscr{S}'$ are then called flip-neighbors. After a flip was performed, in the resulting subdivision it is always possible to perform the reverse-flip.

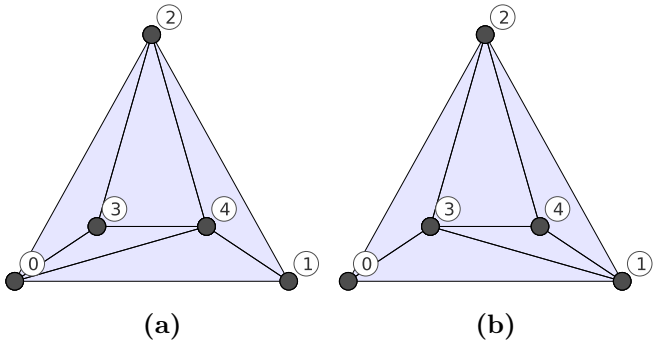


Figure 17: *Flip in a triangulation.* Flip in a triangulation of the point-configuration in Figure 12a.

2.10 Flip-graphs

As already mentioned, applying a flip on a subdivision yields a second subdivision. Applying a second flip to the second subdivision yields a third subdivision. This can be repeated as often as desired. So we can get from one subdivision to other subdivisions by a sequence of flips. Although flips are possible in subdivisions, we are mostly interested in performing flips in triangulations.

The relation between triangulations via flips can be analyzed with a graph (called the flip-graph). Each triangulation corresponds to a vertex in the graph, and each flip corresponds to an edge in the graph.

Definition 2.10.1 (Flip-graph)(see [13])**.** The *flip-graph* $\mathscr{G}$ of a point-configuration is a graph where each triangulation corresponds to a vertex in $\mathscr{G}$, and a flip from one triangulation into another corresponds to an edge in $\mathscr{G}$. The *regular flip-graph* $\mathscr{G}_{reg}$ is the subset of $\mathscr{G}$ consisting only of regular triangulations and regular flips (meaning flips between two regular triangulations, where the common minimal coarsening of both is also regular).

Note, there exist non-regular flips between regular triangulations (see [7, p. 234]).

The problem of determining whether the flip-graph of a point-configuration is connected or not is of great interest. When the flip-graph is connected, one can transform a triangulation into every other triangulation by a sequence of flips. But solving the problem for one point-configuration is not nearly as interesting as solving it for a whole class of point-configurations.

One important attribute of a point-configuration is its dimension d (the dimension of the affine hull of the points). It turns out that the flip-graph of every 2-dimensional point-configuration is connected. In [9, chapter 1.2] an algorithm for flipping an arbitrary triangulation of a 2-dimensional point-configuration into the Delaunay triangulation is given. When one flips from a triangulation $\mathscr{T}_a$ to another triangulation $\mathscr{T}_b$ it is always possible to flip back from $\mathscr{T}_b$ to $\mathscr{T}_a$ (via the reverse-flips). Flipping from any triangulation to any other triangulation is now always possible, because one can flip from the first triangulation to the Delaunay triangulation and then to the second triangulation. In [7, chapter 3.4.1] exists a similar proof where every triangulation is flipped into the pulling triangulation (see [7, chapter 3.2.1] for the definition of the pulling triangulation). Unfortunately, in higher dimensions there is no algorithm known which could flip an arbitrary triangulation into the Delaunay triangulation (or any other triangulation). See [7, chapter 3.6.2] for further details.

In fact, there are examples which show that some flip-graphs in higher dimensions are not connected. In [14], there is a construction of a triangulation of a 6-dimensional point-configuration with 324 points where no flips are possible, and also the construction of a 234-dimensional point-configuration with 552 points where the flip-graph is not connected. Smaller examples in 5 and 6 dimensions are given in [7, chapter 7.3, chapter 7.4]. For dimensions 3 and 4 there are currently no known examples that would show the disconnectedness of any flip-graph. The question whether all flip-graphs in these dimensions are connected is still open.

Regularity is an important attribute of triangulations. An important result is, that the regular flip-graph of a point-configuration in any dimension is connected. This can be seen in the next section on the secondary polytope and is also described in [13].

3 Secondary Polytopes

After having defined the more general structures in the last chapter, we now come to specific ones, which will ultimately lead to the secondary polytope. The secondary polytope will be very helpful in analyzing subdivisions and flips between subdivisions. There exists a bijection between the faces of the secondary polytope and the regular subdivisions of a point-configuration.

In Section 4 we will give a lot of examples of point-configurations, their secondary polytopes, and their refinement posets. During the study of the current section the reader should feel free to look ahead into Section 4 to get examples of the structures described here.

In the following sections one can find definitions for structures related to the subdivisions of a point configuration $\mathcal{A}$. The definitions for these structures equal the usual definitions in literature. Additionally, we also define many of them constrained for a subdivision $\mathscr{S}_0$. The constrained versions consider only the refining subdivisions of $\mathscr{S}_0$ ($\mathit{Refinements}(\mathscr{S}_0)$). It follows that a constrained structure, if it is constrained for the trivial subdivision, equals the general structure.

The constrained versions of the structures are inventions of this work, serving the purpose of relating the non-regular subdivisions to the other subdivisions. From secondary polytopes only information about regular subdivisions can be received, but constrained secondary polytopes can also give information about non-regular subdivisions (and their relations to other subdivisions).

3.1 The space of height functions

A height function w yields a height for every point in a point-configuration. If we interpret this as a tuple of heights (one for each point p_i), we can write a height-vector $\vec{w} = (w(p_1), \ldots, w(p_n)) \in \mathbb{R}^n$. Each height-vector corresponds to a subdivision according to the subdivision function. We write $\mathrm{sub}_{\mathcal{A}}(\vec{w}) := \mathrm{sub}_{\mathcal{A}}(w)$. So we can partition the space $\mathbb{R}^n$ into regions (called secondary cones) where $\mathrm{sub}(\vec{w})$ yields the same subdivision. These regions have some interesting properties.

Definition 3.1.1 (Pointed cone)**.** A subspace of a vector space is called *cone* $\mathcal{C}$ when for every $x \in \mathcal{C}$ and every $\lambda \in \mathbb{R}^+$, it follows that $\lambda x \in \mathcal{C}$. It is *pointed* if it contains the origin.

Definition 3.1.2 (Fan)**.** A *fan* is a geometric polyhedral complex which consists of convex pointed cones and covers the whole vector space.

Definition 3.1.3 (Open secondary cone)(parts from [7, p. 221])**.** Given a subdivision $\mathscr{S}$. The region $\mathcal{C}^o_{\mathcal{A}}(\mathscr{S})$ of $\mathbb{R}^n$ where for each $\vec{w} \in \mathbb{R}^n$: $\text{sub}_{\mathcal{A}}(\vec{w}) = \mathscr{S}$ is called *open secondary cone* of $\mathscr{S}$. For a constraining subdivision $\mathscr{S}_0$, the region $\mathcal{C}^o_{\mathcal{A}}(\mathscr{S}, \mathscr{S}_0)$ with $\text{sub}_{\mathcal{A}}(\vec{w}, \mathscr{S}_0) = \mathscr{S}$ is called *constrained open secondary cone* for $\mathscr{S}_0$.

Definition 3.1.4 (Closed secondary cone)(parts from [7, p. 221])**.** Given a subdivision $\mathscr{S}$. The region $\mathcal{C}_{\mathcal{A}}(\mathscr{S})$ of $\mathbb{R}^n$ where for each $\vec{w} \in \mathbb{R}^n$, $\text{sub}_{\mathcal{A}}(\vec{w}) = \mathscr{S}_{coarse}$, where $\mathscr{S}_{coarse}$ is equal to $\mathscr{S}$ or a coarsening of $\mathscr{S}$, is called *closed secondary cone* of $\mathscr{S}$. For a constraining subdivision $\mathscr{S}_0$, the region $\mathcal{C}_{\mathcal{A}}(\mathscr{S}, \mathscr{S}_0)$ with $\text{sub}_{\mathcal{A}}(\vec{w}, \mathscr{S}_0) = \mathscr{S}_{coarse}$ is called *constrained closed secondary cone* for $\mathscr{S}_0$.

Definition 3.1.5 (Secondary fan)([4])**.** The collection of all closed secondary cones of a point-configuration is called *secondary fan* and denoted $\Sigma-\text{fan}(\mathcal{A})$. For a subdivision $\mathscr{S}_0$, the collection of all constrained closed secondary cones is called *constrained secondary fan* for $\mathscr{S}_0$.

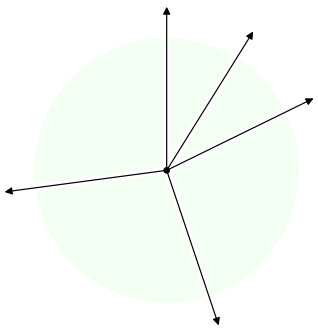


Figure 18: *Secondary fan.* Secondary fan of the point-configuration in Figure 12a p.22. Every ray corresponds to a subdivision of the point-configuration. Every cone (spanned by two rays) corresponds to a triangulation of the point-configuration. The origin corresponds to the trivial subdivision.

Definition 3.1.6 (Secondary cone poset)**.** We define the *secondary cone poset* as the poset of all closed secondary cones of a point-configuration for the subset relation. For a subdivision $\mathscr{S}_0$, the poset of constrained closed secondary cones is called *constrained secondary cone poset* for $\mathscr{S}_0$.

Theorem 3.1.7 (see [7, p. 229])**.** Given a point-configuration, the regular refinement poset is isomorph to the secondary cone poset.

3.2 Secondary polytopes

Now we have everything to define secondary polytopes. Secondary cones already correspond to regular subdivisions of the point-configuration. The secondary fan combines the cones to one structure and therefore relates the regular subdivisions to each other. The secondary polytope is now defined by the secondary fan and thus inherits the ability to relate the regular subdivisions to each other.

Definition 3.2.1 (Normal cone)(in analogy to [7, p. 47])**.** The *inner normal cone* of a polyhedron P for a point $x \in P$ is the set:

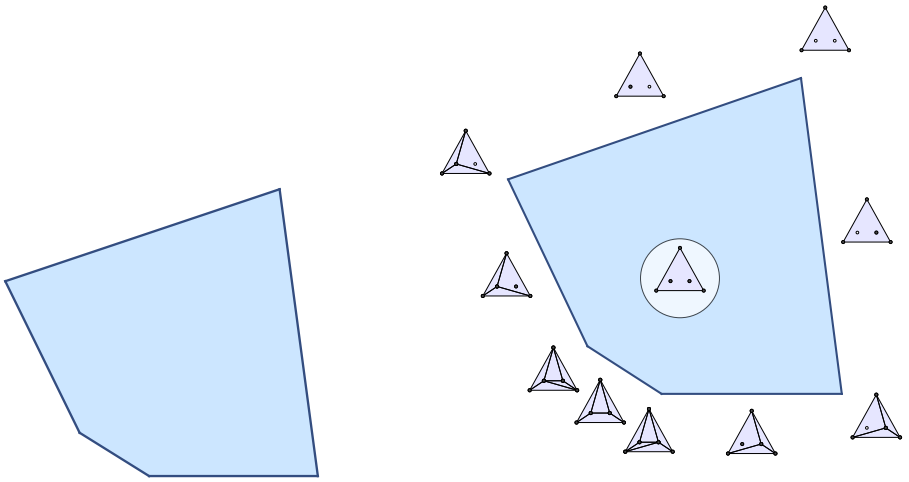
$$\mathrm{N}_P(x) := \{v \in \mathbb{R}^m | < v, x > \leq < v, y > \forall y \in P\} \tag{2}$$

The *outer normal cone* of a polyhedron P for a point $x \in P$ is the negative of the corresponding inner normal cone.

The normal cones have the property that two points $x_a, x_b \in P$ lie in the relative interior of the same face of P, if and only if $\mathrm{N}_P(x_a) = \mathrm{N}_P(x_b)$. This allows us to write $\mathrm{N}_P(F) := \mathrm{N}_P(x)$ for an face F of P and an arbitrary point x from the relative interior of F. Therefore, there exists a bijection between the normal cones and the faces of a polyhedron.

Definition 3.2.2 (Normal fan)(see [7, p. 47])**.** The *inner normal fan* (*outer normal fan*) of a polytope P is the fan consisting of all inner normal cones (outer normal cones) of P.

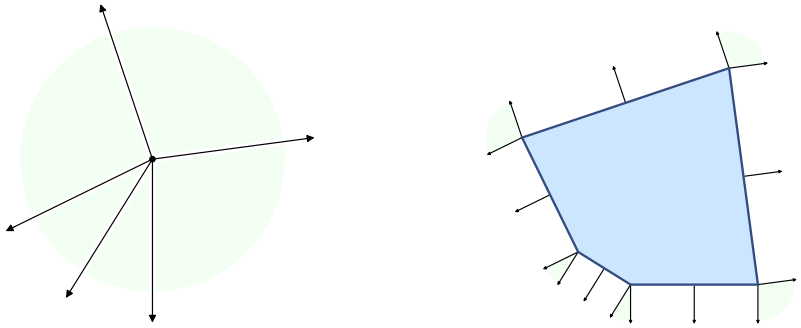
Definition 3.2.3 (Secondary polytope)([4])**.** A *secondary polytope* of a point-configuration is every polytope whose inner normal fan equals the secondary fan. For a constraining subdivision $\mathscr{S}_0$, the *constrained secondary polytope* for $\mathscr{S}_0$ is every polytope whose inner normal fan equals the constrained secondary fan for $\mathscr{S}_0$.



(a) Secondary polytope.

(b) Secondary polytope with corresponding subdivisions.

Figure 19: *Secondary polytope.* Secondary polytope of the point-configuration in Figure 12a p.22. In Figure 12a every face of the secondary polytope has an icon with the subdivision corresponding to the face.



(a) Secondary fan mirrored at origin.

(b) Secondary polytope with outer normal cones.

Figure 20: *Correspondence between secondary fan and secondary polytope.* The secondray fan (Figure 20a), mirrored at the origin, equals the outer normal fan of the secondary polytope (Figure 20b). The outer secondary cones (not the inner ones) are shown, because they can be seen more easily from the shape of the secondary polytope.

The secondary polytope of a 2-dimensional convex point-configuration is equal to a polytope called *associahedron*. The name associahedron comes from the fact that the vertices of the associahedron correspond to all different ways of bracketing a string. See [12] or [20, p. 18] for more.

3.3 GKZ secondary polytopes

After having defined a secondary polytope as *any* polytope whose inner normal fan equals the secondary fan, the question of how to explicitly construct a secondary polytope arises. There exist various methods to construct the secondary polytope (see [4]). We will use the probably most simple method, which is the one that works with GKZ-vectors (Gelfand-Kapranov-Zelevinsky-vectors) (see [4, chapter 2], [7, chapter 5], [11, chapter 7]). Using this method, we relate one GKZ-vector to each triangulation of the point-configuration. The convex hull of all GKZ-vectors will be a polytope that fulfills the requirements for being a secondary polytope.

Definition 3.3.1 (GKZ-Vector)([7, p. 215])**.** Take a d-dimensional point-configuration $\mathcal{A}$ with n points on a set of labels $\mathcal{L}$. Let us assume that $\mathcal{L}$ is the set $\{1, \ldots, n\}$. The *GKZ-vector* of a triangulation $\mathscr{T}$ is the vector:

$$\phi_{\mathcal{A}}(\mathscr{T}) := \sum_{l \in \mathcal{L}} \sum_{C \in \mathscr{T}, l \in C} (\mathrm{vol}(C) \cdot e_l) \in \mathbb{R}^n \tag{3}$$

Where e_l is the l-th canonical basis vector and $\mathrm{vol}(C)$ is the d-dimensional volume of C. If C is not d-dimensional, $\mathrm{vol}(C) = 0$.

Definition 3.3.2 (GKZ secondary polytope)([4, chapter 2])**.** The *GKZ secondary polytope* of a point-configuration is the polytope:

$$\Sigma{-}\mathrm{poly}(\mathcal{A}) := \mathrm{conv}(\{\phi_{\mathcal{A}}(\mathscr{T}) | \mathscr{T} \in \mathit{Triangulations}(\mathcal{A})\}) \tag{4}$$

The *constrained GKZ secondary polytope*, constrained for a subdivision $\mathscr{S}_0$, is the polytope:

$$\Sigma{-}\mathrm{poly}(\mathcal{A}, \mathscr{S}_0) := \mathrm{conv}(\{\phi_{\mathcal{A}}(\mathscr{T}) | \mathscr{T} \in \mathit{Triangulations}(\mathscr{S}_0)\}) \tag{5}$$

That $\Sigma{-}\mathrm{poly}(\mathcal{A})$ is a secondary polytope will be shown in the following theorems. We are often only interested in any secondary polytope, and the GKZ construction is an easy way to obtain a secondary polytope. Therefore, we will refer to the GKZ secondary polytope just with "the secondary polytope". Note again that $\Sigma{-}\mathrm{poly}(\mathcal{A}) = \Sigma{-}\mathrm{poly}(\mathcal{A}, \mathscr{S}_0)$ when $\mathscr{S}_0$ is the trivial subdivision.

Definition 3.3.3 (Characteristic section)([7, p. 230])**.** For a triangulation $\mathscr{T}$ and a height-vector $\vec{w}$, the *characteristic section* $g_{\vec{w},\mathscr{T}} : \text{conv}(\mathcal{A}) \to \mathbb{R}$ is defined for every point p_i that is part of a cell of $\mathscr{T}$ as $g_{\vec{w},\mathscr{T}}(p_i) \mapsto \vec{w}_i$, and as affine extension on $\text{conv}(C)$ for every cell $C \in \mathscr{T}$. The *lifted characteristic section* is defined as the function $G_{\vec{w},\mathscr{T}} : \text{conv}(\mathcal{A}) \to \mathbb{R}^{m+1}$ with $x \mapsto \begin{pmatrix} x \\ g_{\vec{w},\mathscr{T}}(x) \end{pmatrix}$.

Lemma 3.3.4 (Lifting-lemma)(adapted and extended from [7, p. 230])**.** For a triangulation $\mathscr{T}$ and a height-vector $\vec{w}$ follows:

$$\mathscr{T} \preceq \text{sub}_{\mathcal{A}}(\vec{w}) \Rightarrow g_{\vec{w},\mathscr{T}} \leq g_{\vec{w},\mathscr{T}'} \; \forall \mathscr{T}' \in \textit{Triangulations}(\mathcal{A}). \tag{6}$$

For a constraining subdivision $\mathscr{S}_0$ this can be restated as:

$$\mathscr{T} \preceq \text{sub}_{\mathcal{A}}(\vec{w}, \mathscr{S}_0) \Rightarrow g_{\vec{w},\mathscr{T}} \leq g_{\vec{w},\mathscr{T}'} \; \forall \mathscr{T}' \in \textit{Triangulations}(\mathscr{S}_0). \tag{7}$$

Proof. From the definition of the convex hull follows that for every cell C, the lifted cell $G_{\vec{w},\mathscr{T}}(C)$ lies in the convex hull of the lifted point-configuration $\text{conv}(\mathcal{A}^w)$. Since $\text{sub}_{\mathcal{A}}(\vec{w})$ is the projection of the lower convex hull of $\text{conv}(\mathcal{A}^w)$, $G_{\vec{w},\mathscr{T}}(C)$ must also lie in the lower convex hull (otherwise not every cell of $\mathscr{T}$ could be the subset of a cell of $\text{sub}_{\mathcal{A}}(\vec{w})$). This implies that $g_{\vec{w},\mathscr{T}} \leq g_{\vec{w},\mathscr{T}'}$ for all $\mathscr{T}' \in \textit{Triangulations}(\mathcal{A})$.
For a constraining subdivision $\mathscr{S}_0$ the proof is analog, by successively taking one cell $C_0 \in \mathscr{S}_0$ and replacing $\mathcal{A}$ with $\mathcal{A}_{C_0}$. □

The lifting lemma is stated and proven in [7, p. 230] also in the other direction, but we will only need this direction here.

Theorem 3.3.5 (adapted from [7, p. 231])**.** The inner normal fan of the secondary polytope $\Sigma\text{−poly}(\mathcal{A})$ equals the secondary fan $\Sigma\text{−fan}(\mathcal{A})$.

Proof. We show that for each triangulation $\mathscr{T}$, the closed secondary cone $\mathcal{C}_{\mathcal{A}}(\mathscr{T})$ from the secondary fan is subset of the inner normal cone $\text{N}_{\Sigma-\text{poly}(\mathcal{A})}(\phi_{\mathcal{A}}(\mathscr{T}))$ from the inner normal fan of the secondary polytope. Showing that one cone is subset of the other is sufficient to proof the equality of both fans, because every fan fills out its underlying space completely. So we take $\mathscr{T} \in \textit{Triangulations}(\mathcal{A})$ and have to show:

$$\mathcal{C}_{\mathcal{A}}(\mathscr{T}) \overset{!}{\subseteq} \text{N}_{\Sigma-\text{poly}(\mathcal{A})}(\phi_{\mathcal{A}}(\mathscr{T})) \tag{8}$$

Take any height-vector $\vec{w} \in \mathcal{C}_{\mathcal{A}}(\mathscr{T})$. From the definition of a normal cone (Definition 3.2.1) follows that the following equation must be fulfilled:

$$< \vec{w}, \phi_{\mathcal{A}}(\mathscr{T}) > \overset{!}{\leq} < \vec{w}, x > \;\; \forall x \in \Sigma\text{−poly}(\mathcal{A}) \tag{9}$$

For this to be true it is sufficient to take x from the set of all vertices of $\Sigma-\text{poly}(\mathcal{A})$ (because they are the extremal points of $\Sigma-\text{poly}(\mathcal{A})$). Since the vertices of $\Sigma-\text{poly}(\mathcal{A})$ are among the GKZ-vectors, the above equation equals:

$$< \vec{w}, \phi_{\mathcal{A}}(\mathscr{T}) > \overset{!}{\leq} < \vec{w}, \phi_{\mathcal{A}}(\mathscr{T}') > \ \forall \mathscr{T}' \in \textit{Triangulations}(\mathcal{A}) \tag{10}$$

For a simplicial cell C from a triangulation, $G_{\vec{w},\mathscr{T}'}(C)$ is a cell in the lifted point-configuration that consists of the same labels as C. Now we can rewrite the scalar-product as follows:

$$\begin{aligned}
& < \vec{w}, \phi_{\mathcal{A}}(\mathscr{T}') > && (11)\\
=& \sum_{j\in J} \vec{w}_j \phi_{\mathcal{A}}(\mathscr{T}')_j && (12)\\
=& \sum_{j\in J} \sum_{C\in\mathscr{T}':j\in C} \vec{w}_j \text{vol}(C) && (13)\\
=& \sum_{C\in\mathscr{T}'} \sum_{j\in C} \vec{w}_j \text{vol}(C) && (14)\\
=& \sum_{C\in\mathscr{T}'} \text{vol}(C) \sum_{j\in C} \vec{w}_j && (15)\\
=& (d+1) \sum_{C\in\mathscr{T}'} \text{vol}(C) \left(\frac{1}{d+1}\right) \sum_{j\in C} \vec{w}_j && (16)\\
=& (d+1) \int_{\text{conv}(\mathcal{A})} g_{\vec{w},\mathscr{T}'}(x)\,\mathrm{d}x. && (17)
\end{aligned}$$

With the conversion:

$$\begin{aligned}
& \sum_{C\in\mathscr{T}'} \text{vol}(C) \left(\frac{1}{d+1}\right) \sum_{j\in C} \vec{w}_j && (18)\\
=& \sum_{C\in\mathscr{T}'} \text{vol}(C) * \textit{height of barycenter of } G_{\vec{w},\mathscr{T}'}(C) && (19)\\
=& \sum_{C\in\mathscr{T}'} \int_C g_{\vec{w},\mathscr{T}'}(x)\,\mathrm{d}x && (20)\\
=& \int_{\text{conv}(\mathcal{A})} g_{\vec{w},\mathscr{T}'}(x)\,\mathrm{d}x. && (21)
\end{aligned}$$

From Lemma 3.3.4 now follows that the integral is minimized for any $\mathscr{T}'$ with $\mathscr{T}' \preceq \text{sub}_{\mathcal{A}}(\vec{w})$, especially for $\mathscr{T}' = \mathscr{T}$ since $\mathscr{T} \preceq \text{sub}_{\mathcal{A}}(\vec{w})$ because we have chosen $\vec{w} \in \mathcal{C}_{\mathcal{A}}(\mathscr{T})$. □

From Definition 3.2.3 follows that $\Sigma-\text{poly}(\mathcal{A})$ is indeed a secondary polytope of $\mathcal{A}$. Now we show the same for the constrained version:

Theorem 3.3.6. For a constraining subdivision $\mathscr{S}_0$, the inner normal fan of the constrained secondary polytope $\Sigma-\text{poly}(\mathcal{A}, \mathscr{S}_0)$ equals the constrained secondary fan $\Sigma-\text{fan}(\mathcal{A}, \mathscr{S}_0)$.

Proof. The proof is equal to the proof of Theorem 3.3.5 after the replacement of *Triangulations*$(\mathcal{A})$ with *Triangulations*$(\mathscr{S}_0)$, $\Sigma-\text{poly}(\mathcal{A})$ with $\Sigma-\text{poly}(\mathcal{A}, \mathscr{S}_0)$, and $\mathcal{C}_{\mathcal{A}}(\mathscr{T})$ with $\mathcal{C}_{\mathcal{A}}(\mathscr{T}, \mathscr{S}_0)$. □

Theorem 3.3.7 (see [7, p. 217])**.** The face poset of the secondary polytope is isomorph to the secondary cone poset, and thus isomorph to the regular refinement poset of the point-configuration.

Corollary 3.3.8 (corollary of [7, p. 217])**.** There exists a bijection between the faces of the secondary polytope and the regular subdivisions.

So we can assign a regular subdivision to every face of the secondary polytope. For a subdivision $\mathscr{S}_0$ all relative regular subdivisions can also be assigned to faces of the constrained secondary polytope, constrained for $\mathscr{S}_0$.

Corollary 3.3.9 (corollary of [7, p. 217])**.** The dimension of the secondary polytope equals the height of the regular refinement poset.

Theorem 3.3.10 ([7, p. 218])**.** The secondary polytope of a d-dimensional point-configuration $\mathcal{A}$ with n points has dimension $(n - d - 1)$. Formally:

$$\dim(\Sigma-\text{poly}(\mathcal{A})) = n - d - 1. \tag{22}$$

Theorem 3.3.11 (partly similar in [7, p. 233], described in [13])**.** The 1-skeleton (the graph with only the vertices and edges of a polytope) of the secondary polytope equals the regular flip-graph of the point-configuration.

Corollary 3.3.12 ([7, p. 233])**.** The regular flip-graph of a d-dimensional point-configuration is connected and on each regular triangulation at least $(n - d - 1)$ flips can be performed.

Proof. The 1-skeleton of a polytope is connected. From Theorem 3.3.11 follows that the flip-graph is connected. Since the secondary polytope has dimension $(n-d-1)$ (as stated in 3.3.10), every vertex has at least $(n-d-1)$ incident edges. This means that for a regular triangulation $(n-d-1)$ flips can be performed. □

The constrained secondary polytopes are not only other types of secondary polytopes. It seems obvious (partly from the examples in Section 4) that a face of a secondary polytope is a constrained secondary polytope itself. Unfortunately, the proof for this could not be found, and so we can only state the following conjecture:

Conjecture 3.3.13. Let $\mathscr{S}$ be the corresponding subdivision to a face F of the secondary polytope. F is the convex hull of all vertices, whose corresponding triangulations are refinements of $\mathscr{S}$. Therefore F is $\Sigma-\text{poly}(\mathcal{A}, \mathscr{S})$.

We will show that a constrained secondary polytope is characterized by other, special secondary polytopes. More exactly, the constrained secondary polytope is the Minkowski-sum of the other secondary polytopes. An overview of Minkowski-sums can be found in [19]. We briefly give the definition and some statements of Minkowski-sums.

Definition 3.3.14 (Minkowski-sum)([20, p. 28])**.** The *Minkowski-sum* of two sets $A, B \subseteq \mathbb{R}^m$ is defined as:

$$A + B := \{a + b \mid a \in A, b \in B\}. \tag{23}$$

The following two technical lemmas are proven here since no reference was found:

Lemma 3.3.15. Let A, B be finite subsets of $\mathbb{R}^m$. Then $\text{conv}(A) + \text{conv}(B) = \text{conv}(A + B)$.

Proof. Let $P_A = \text{conv}(A)$, $P_B = \text{conv}(B)$, $P_{A+B} = \text{conv}(A+B)$, and $P' = P_A + P_B$.

- P' is convex because: $\forall p, q \in P' : \exists p_a, q_a \in P_A \ \wedge \ p_b, q_b \in P_B : p = p_a + p_b \ \wedge \ q = q_a + q_b$. Every point in $\text{conv}(\{p, q\})$ can now be generated as the sum of a point in $\text{conv}(\{p_a, p_b\})$ and a point in $\text{conv}(\{q_a, q_b\})$.

- P' is a polytope because: An extremal point of a convex set S is a point $x \in S$ such that $\nexists x_1, x_2 \in S : x_1 \neq x_2 \neq x \wedge x \in \text{conv}(\{x_1, x_2\})$. The extremal points of a polytope are the vertices. For all $p \in P'$ exist $a \in P_A, b \in P_B$ such that $p = a + b$. Assume that a is not extremal. Then $\exists a_1, a_2 \in P_A : a_1 \neq a_2 \neq a \ \wedge \ a \in \text{conv}(\{a_1, a_2\})$. Since $a_1, a_2 \in P_A$ it follows that $a_1 + b, a_2 + b \in P'$. Because $a \in \text{conv}(\{a_1, a_2\})$ it follows that $a + b \in \text{conv}(\{a_1 + b, a_2 + b\})$. This means that $p = a + b$ cannot be an extremal point. So every extremal point of P' is the sum of a vertex in P_A and a vertex in P_B. Therefore the set of extremal points of P' is finite and P' is the convex hull of a finite set of points and thus a polytope.

- $P' = P_{A+B}$ because: P' is the convex hull of its vertices and, as shown in the last point, every vertex of P' is the sum of a vertex in P_A and a vertex in P_B. The vertices of P_A and P_B are among A and B. Therefore $P' = \text{conv}(A+B)$.

□

Lemma 3.3.16. For two polyhedra P, P' holds that $\dim(P + P') \leq \dim(P) + \dim(P')$.

Proof. P, P', and $P + P'$ lie in their corresponding affine hulls $\text{aff}(P)$, $\text{aff}(P')$, and $\text{aff}(P + P')$. The polyhedra also have the same dimension as their corresponding affine hulls. Take a basis $\mathscr{B}_P$ of $\text{aff}(P)$ and a basis $\mathscr{B}_{P'}$ of $\text{aff}(P')$. $\text{aff}(P + P')$ is spanned by $\mathscr{B}_P \cup \mathscr{B}_{P'}$. Therefore, $\dim(P+P') = \dim(\text{aff}(P+P')) \leq \dim(\text{aff}(P))+ \dim(\text{aff}(P')) = \dim(P) + \dim(P')$. □

This leads us to the following theorem about constrained secondary polytopes:

Theorem 3.3.17. For a constraining subdivision $\mathscr{S}_0$, $\Sigma-\text{poly}(\mathcal{A}, \mathscr{S}_0)$ equals the Minkowski-sum of $\Sigma-\text{poly}(\mathcal{A}_{C_0})$ for all $C_0 \in \mathscr{S}_0$.

Proof. As a polytope, the secondary polytope is defined as the convex hull of its vertices. Every vertex is a GKZ-vector of a triangulation. We will show that every GKZ-vector of $\Sigma-\text{poly}(\mathcal{A}, \mathscr{S}_0)$ is the sum of one GKZ-vector of $\Sigma-\text{poly}(\mathcal{A}_{C_0})$ for every $C_0 \in \mathscr{S}_0$.
Let $\mathscr{T}$ be a triangulation which is a refinement of $\mathscr{S}_0$. For a cell $C_0 \in \mathscr{S}_0$ we define $\mathscr{T}|_{C_0} = \{C \in \mathscr{T} \mid C \subseteq C_0\}$. $\mathscr{T}|_{C_0}$ is therefore a triangulation of $\mathcal{A}_{C_0}$. We can now rewrite the GKZ-vector of $\mathscr{T}$ in the following way:

$$\begin{aligned}
&\phi_{\mathcal{A}}(\mathscr{T}) && (24)\\
=&\sum_{l\in\mathcal{L}} \sum_{C\in\mathscr{T}, l\in C} \text{vol}(C)\ e_l && (25)\\
=&\sum_{l\in\mathcal{L}} \sum_{C_0\in\mathscr{S}_0} \sum_{C\in\mathscr{T}|_{C_0}, l\in C} \text{vol}(C)\ e_l && (26)\\
=&\sum_{C_0\in\mathscr{S}_0} \sum_{l\in\mathcal{L}} \sum_{C\in\mathscr{T}|_{C_0}, l\in C} \text{vol}(C)\ e_l && (27)\\
=&\sum_{C_0\in\mathscr{S}_0} \phi_{\mathcal{A}}(\mathscr{T}|_{C_0}). && (28)
\end{aligned}$$

Fix a $C_0 \in \mathscr{S}_0$. Since $\mathscr{T}$ can be any refinement of $\mathscr{S}_0$, any triangulation of $\mathcal{A}_{C_0}$ can equal $\mathscr{T}|_{C_0}$. So every sum of one GKZ-vector of $\Sigma-\text{poly}(\mathcal{A}_{C_0})$ for every $C_0 \in \mathscr{S}_0$ is also a GKZ-vector of $\Sigma-\text{poly}(\mathcal{A}, \mathscr{S}_0)$.
Let $C_1, \ldots, C_k$ be all the cells of $\mathscr{S}_0$. The arguments above show that:

$$\begin{aligned}
&\{\phi_{\mathcal{A}}(\mathscr{T}) \mid \mathscr{T} \in \textit{Triangulations}(\mathscr{S}_0)\} && (29)\\
=&\{\sum_{i=1}^{k} \phi_{\mathcal{A}}(\mathscr{T}_i) \mid \mathscr{T}_i \in \textit{Triangulations}(\mathcal{A}_{C_i})\}. && (30)
\end{aligned}$$

From Lemma 3.3.15 follows:

$$\text{conv}(\{\phi_{\mathcal{A}}(\mathscr{T}) \mid \mathscr{T} \in \textit{Triangulations}(\mathscr{S}_0)\}) \tag{31}$$

$$=\text{conv}(\{\sum_{i=1}^{k} \phi_{\mathcal{A}}(\mathscr{T}_i) \mid \mathscr{T}_i \in \textit{Triangulations}(\mathcal{A}_{C_i})\}) \tag{32}$$

$$=\sum_{i=1}^{k} \text{conv}(\{\phi_{\mathcal{A}}(\mathscr{T}_i) \mid \mathscr{T}_i \in \textit{Triangulations}(\mathcal{A}_{C_i})\}). \tag{33}$$

□

Theorem 3.3.18. Let $C_1, \ldots, C_k$ be all the cells of a subdivision $\mathscr{S}_0$. It holds that:

$$\dim(\Sigma\text{−poly}(\mathcal{A}, \mathscr{S}_0)) \leq \sum_{i=1}^{k} \dim(\Sigma\text{−poly}(\mathcal{A}_{C_i})). \tag{34}$$

Proof. This follows immediately from Theorem 3.3.17 and Lemma 3.3.16. □

3.4 Future work on constrained secondary polytopes

Unfortunately, we could not give any new answers to the question of the connectedness of the flip-graph. But an overview of structures which are useful for studying the flip-graph and definitions of additional structures that should also be useful for studying the flip-graph have been given.

The last sections have shown the build-up and properties of the secondary polytope. Additionally, the definition of the constrained secondary polytope was established. Some results for constrained secondary polytopes were given, but there remain many more to be discovered. Here are a few open issues:

Carry over results from secondary polytopes to constrained secondary polytopes

It seems obvious that most of the properties of secondary polytopes also hold for constrained secondary polytopes. Therefore, the theorems found for secondary polytopes should also be examined for their validity for constrained secondary polytopes.

The flip-graph and constrained secondary polytopes

Constrained secondary polytopes help with the understanding of the relationships between subdivisions and triangulations, since with their help one can relate the non-regular subdivisions to the other subdivisions. This is important when one studies the flip-graph of triangulations. For all regular triangulations of a point-configuration the flip-graph is connected. But there can be non-regular triangulations that can't be reached via flips from some other triangulations (as described in 2.10). For such non-regular triangulations, analyzing their coarsenings and the constrained secondary polytopes and constrained refinement posets, constrained for this coarsenings, might lead to results concerning the nature of these non-regular triangulations.

Complexes of cells

Constrained secondary polytopes can replace secondary polytopes completely, because when the constraining subdivision is the trivial subdivision, the constrained secondary polytope equals the secondary polytope. Therefore, one could simply define "secondary polytopes" for a constraining subdivision, and work with one instead of two definitions. The definition of the constrained secondary polytope was chosen, because we wanted to use the common definition of secondary polytopes in literature.

Moreover, all the other structures defined for a constraining subdivision equal the standard structures in literature, when the constraining subdivision is the trivial subdivision. Giving only definitions of structures with constraining subdivisions would reduce the amount of definitions.

But the theory may be generalized even more, if we don't regard subdivisions but complexes of cells. A complex of cells $\mathscr{C}$ would be a subdivision without the Union Property. By this definition, a triangulation, a subdivision, and even a single cell would be a complex of cells. On $\mathscr{C}$ we could define a "secondary cell polytope" $\Sigma-\text{cpoly}(\mathscr{C})$ which is the convex hull of all GKZ-vectors of refining complexes of $\mathscr{C}$ where every cell is a simplex. If $\mathscr{C}$ consists only of the trivial cell, $\Sigma-\text{cpoly}(\mathscr{C})$ would be the secondary polytope. If $\mathscr{C}$ is a subdivision $\mathscr{S}$, $\Sigma-\text{cpoly}(\mathscr{C})$ would be the constrained secondary polytope, constrained for $\mathscr{S}$.

Since even a subset of a triangulation would be a complex of cells, any region of a triangulation could be analyzed separately. The union of two complexes of cells is, under the condition that all cells intersect properly, again a complex of

cells. The secondary cell polytope of the union of two cell complexes $\mathscr{C}_a$, $\mathscr{C}_b$ would be the Minkowski-sum of the secondary cell polytopes of both complexes ($\Sigma-\text{cpoly}(\mathscr{C}_a \cup \mathscr{C}_b) = \Sigma-\text{cpoly}(\mathscr{C}_a) + \Sigma-\text{cpoly}(\mathscr{C}_b)$) (in analogy to 3.3.17).

So secondary cell polytopes would replace the other secondary polytopes, and they could also be used to build other secondary cell polytopes via Minkowski-sums.

4 Interesting Point Configurations

In the last chapters we defined many structures for the analysis of triangulations and flips. There are also many relations between this structures. In order to understand them better one should probably try to imagine their geometric shape by examining their properties. But concluding the shapes from some properties is sometimes cumbersome. Concrete examples are of great help when it comes to get an intuition about the structure and for deriving properties. This is of course only possible when there exists a good way to visualize the examples. Primarily the dimension of the object should not be too high, or it will not be visualizable in a satisfying way.

The following examples all show secondary polytopes and refinement posets on simple, small, 2-dimensional point-configurations. Unfortunately, for more complicated examples, the refinement posets are getting too big and too complicated. Especially the dimension of the secondary polytopes is getting higher than 3. The dimension of the secondary polytope is overall a critical point for the analysis. In dimension 2 and lower, the secondary polytopes are rather simple, not that interesting, and no non-regular subdivisions can occur for the point-configuration. In dimension 3, the shape of the secondary polytopes gets more complex, they are still visualizable, and non-regular subdivisions can occur for the point-configuration. In dimension 4 or higher, the abilities to visualize secondary polytopes are limited. Therefore, in all but one example, the secondary polytopes will be 3-dimensional.

The constraints mentioned above lead to examples for point-configurations with a maximum of 6 points in the 2-dimensional plane (since the dimension of the secondary polytope is $n - d - 1$ according to Theorem 3.3.10). Nevertheless, the examples already show a lot of properties of the structures.

All graphics have been computed from the point-configurations coordinates (depicted in each case at the beginning). For a secondary polytope this means that the illustration is exact and not only a sketch. At the vertices of each secondary polytope there are icons, which show the triangulation corresponding to that vertex. In the first example, there is also an icon near each face, which depicts the corresponding subdivision. All regular subdivisions are shown in blue, and all non-regular ones in red. The Delaunay subdivision is marked with a thick blue circle. Additionally, sometimes subdivisions are marked with a dotted circle. The special properties of these marked subdivisions are described in the respective figure caption. The presentation of 3-dimensional secondary polytopes has an effect of illumination in order to highlight their 3-dimensional structure. Some refinement posets are split into parts because otherwise they would be too big.

4.1 2-dimensional point-configurations in convex position

We begin with point-configurations in convex position. Convexity implies that every point must be part of every subdivision, and that no non-regular subdivisions can occur. The points are aligned equally on a circle in order to achieve the highest possible symmetry for the secondary polytope. Note that this only increases the aesthetics of the presentation. Even if the points were arranged in a different way, as long as they are in convex position, the combinatorics of their subdivisions would be the same. It follows that the refinement posets would stay the same, and the secondary polytopes would be deformed, but they would combinatorially stay the same. The Delaunay subdivisions, however, is for this positioning of points always the trivial subdivision, but would be another subdivision for another positioning.

4.1.1 5 points in convex position

The first point-configuration consists of 5 points equally aligned on a circle (Figure 21). On this point-configuration exist 5 triangulations and altogether 11 subdivisions.

The secondary polytope (Figure 22) is 2-dimensional and has the shape of a pentagon. Each triangulation has 2 edges in the inside of the convex hull of the point-configuration (also visible in Figure 23, last row). This means that for each triangulation there are 2 flips possible (since no vertex insertion or vertex removal flip is possible). That can also be seen in the secondary polytope, where 2 edges are incident to each vertex.

The refinement poset (Figure 23) is relatively simple and has height 2, which equals the dimension of the secondary polytope.

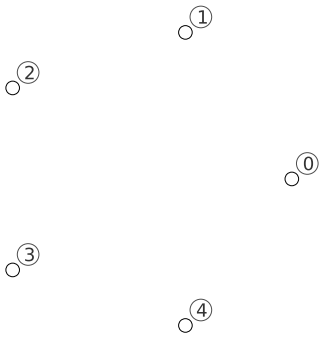


Figure 21: *5 points in convex position.* Point-configuration with 5 points on a circle.

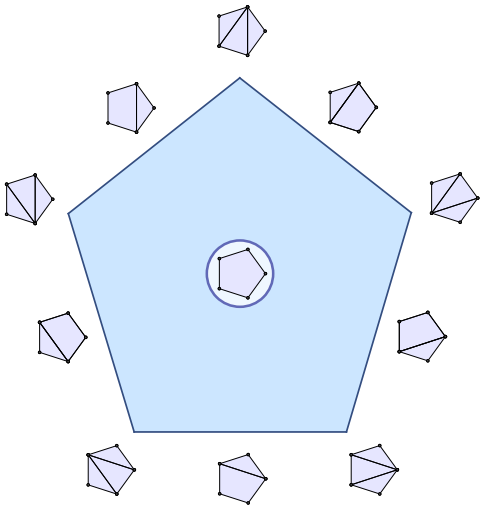


Figure 22: *Secondary polytope of 5 points in convex position.* Secondary polytope of the point-configuration in Figure 21 (with subdivisions corresponding to faces).

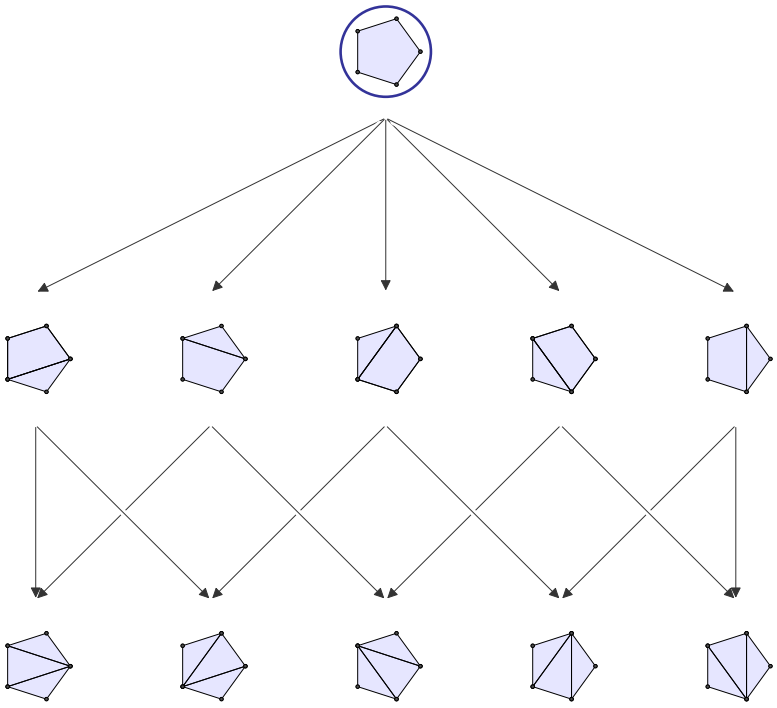


Figure 23: *Refinement poset of 5 points in convex position.* Refinement poset of the point-configuration in Figure 21.

4.1.2 6 points in convex position

Now we increase the number of points by one. The point-configuration now consists of 6 points equally aligned on a circle (Figure 24). This results in an increase of triangulations to 14 and of all subdivisions to 45.

The secondary polytope (Figure 25) is now 3-dimensional and has 6 facets in the shape of a pentagon and 3 facets in the shape of a parallelogram. Each of the pentagonal facets is similar to the secondary polytope of 5 points in convex position. If we add one point and an incident triangle to each subdivision of the 5 points, we get every subdivision corresponding to a face of one pentagonal facet. A facet F in the form of a parallelogram also has an interesting analogy. It has a corresponding subdivision $\mathscr{S}$. The faces of F contain two pairs of parallel edges. Parallel edges correspond to the same flip in different triangulations. So, in all triangulations that are refinements of $\mathscr{S}$ only 2 different flips are possible (if you count the reverse-flips additionally, 4 different flips are possible). This can also be seen in $\mathscr{S}$, since it is a subdivision where the only edge that is not on the convex hull of the point-configuration is a diagonal from one point to the opposite point. Each of the 2 cells in $\mathscr{S}$ can be refined in 2 different ways, which means that there is 1 flip possible (2 flips if you count the reverse-flips additionally). Since there are 2 cells in $\mathscr{S}$, the constrained secondary polytope, constrained for $\mathscr{S}$, is the Minkowski-sum of the secondary polytopes of the two cells (as stated in 3.3.17).

The refinement poset (Figure 26) already increases radically. Its height is 3, equaling the dimension of the secondary polytope.

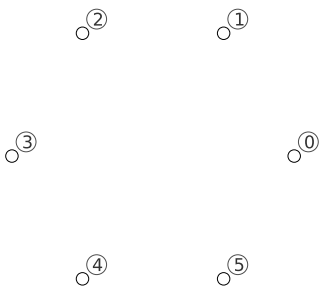


Figure 24: *6 points in convex position.* Point-configuration with 6 points on a circle.

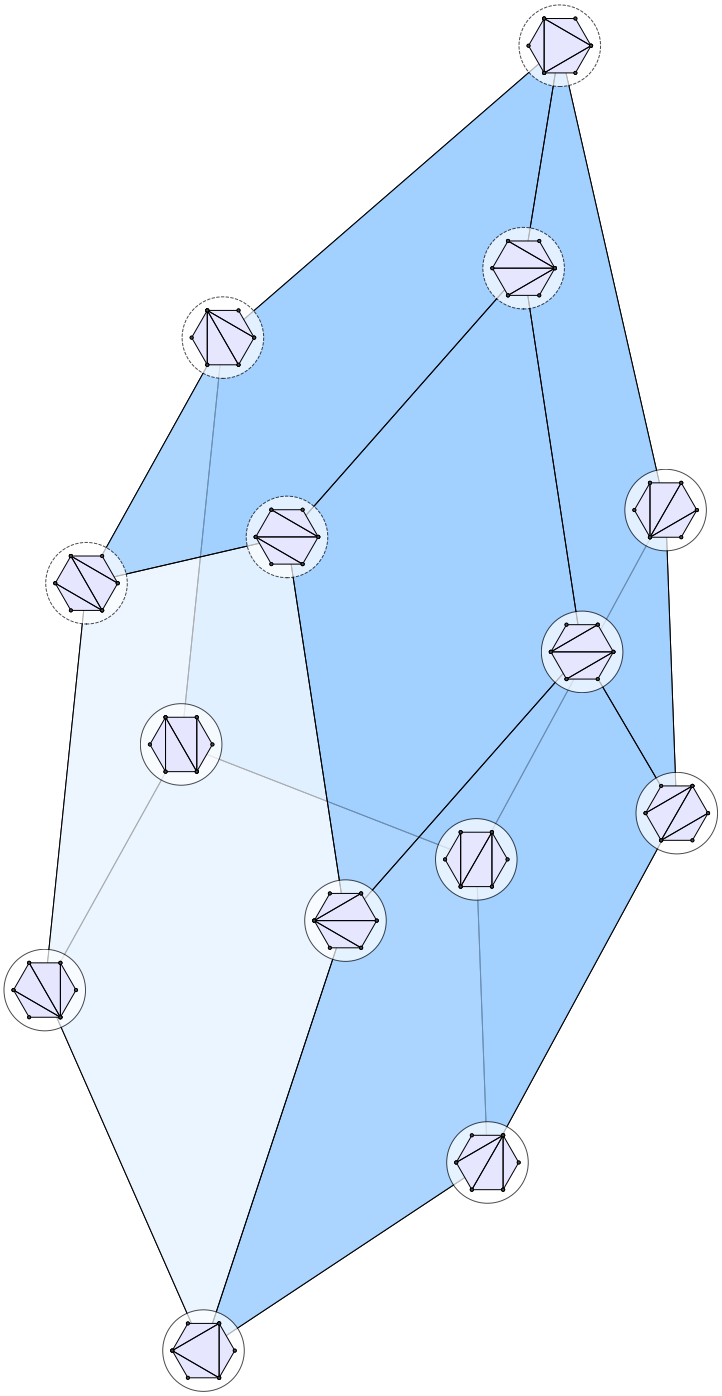


Figure 25: *Secondary polytope of 6 points in convex position.* Secondary polytope of the point-configuration in Figure 24. The 5 triangulations marked with a dotted circle correspond to the 5 triangulations of the 5 points in convex position in Figure 21, if you glue the upper right point and its incident triangle to every triangulation of the 5 points.

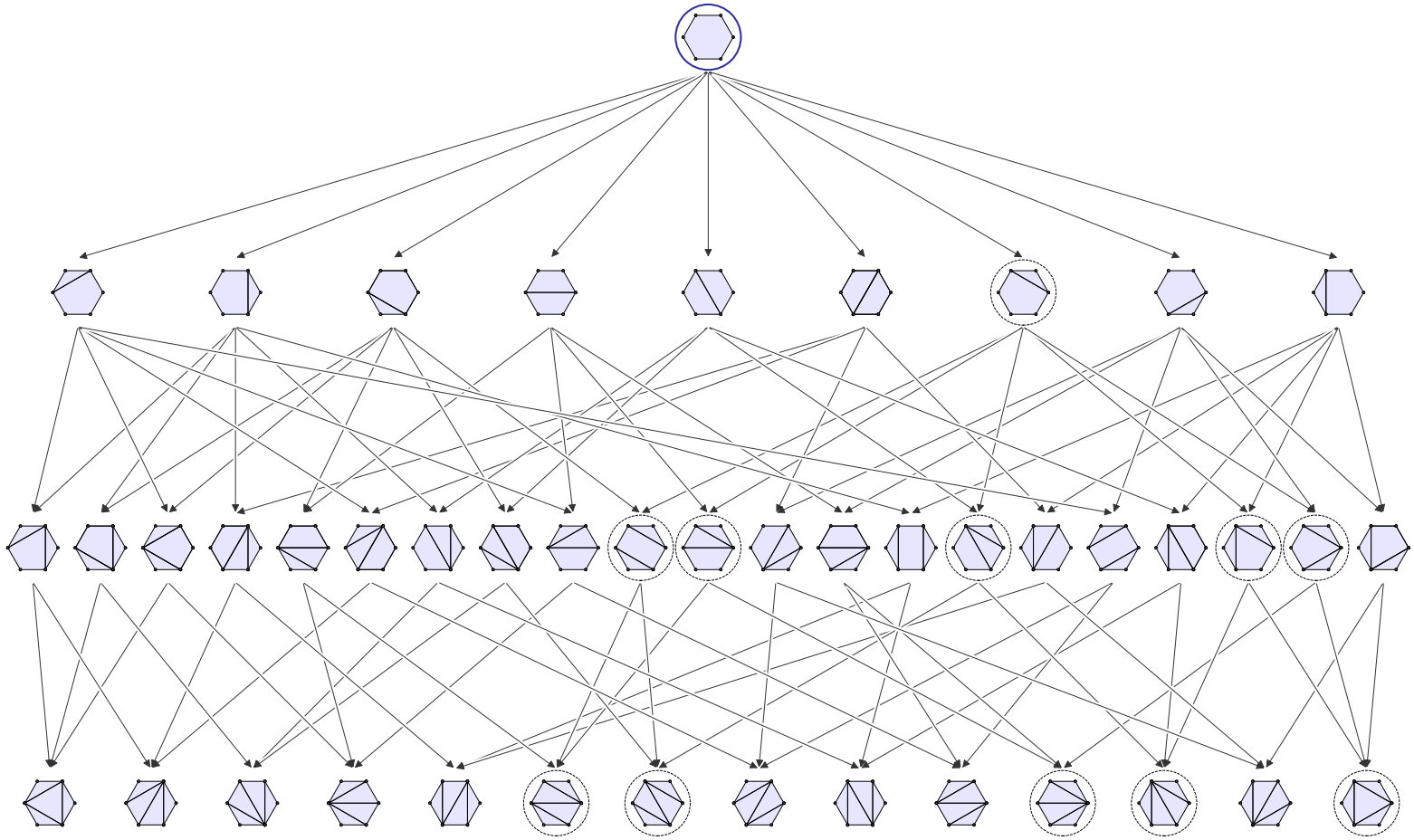


Figure 26: *Refinement poset of 6 points in convex position.* Refinement poset of the point-configuration in Figure 24. The subdivisions marked with a dotted circle equal the subdivisions of the 5 points in convex position in Figure 21, if you glue the upper right point and its incident triangle to every subdivision of the 5 points. So, the refinement poset in Figure 23 is isomorph to the poset containing only the marked subdivisions of this figure.

4.2 2-dimensional point-configurations

Now we omit the requirement that all points must lie in convex position. This means that not every subdivision must use all points of the point-configuration and that non-regular subdivisions may occur. If we translate the points, the positioning of the points in respect to each other changes. Therefore, also the set of all subdivisions, the refinement poset, and the secondary polytope change. These properties are in contrast to the properties of a 2-dimensional point-configuration in convex position (see Section 4.1). However, we still give examples of point-configurations with a high amount of symmetry in order to achieve symmetry in the secondary polytopes.

4.2.1 5 points in convex position with 1 central point

This point-configuration can also be seen as an extension to the 5 points in convex position from Section 4.1.1. The additional point is now inserted in the center of the point-configuration (Figure 27). Now the number of triangulations rises to 16 and that of all subdivisions to 53.

The secondary polytope (Figure 28) has one pentagonal facet (on the backside of the shown secondary polytope) which is the same as the secondary polytope of 5 points in convex position in Figure 22. All the subdivisions in this facet omit the central point. Among the other faces there are faces in form of a parallelogram which are Minkowski-sums of constrained secondary polytopes (analog to the faces in form of a parallelogram from the example in Section 4.1.2).

The refinement poset (Figure 29) has again height 3.

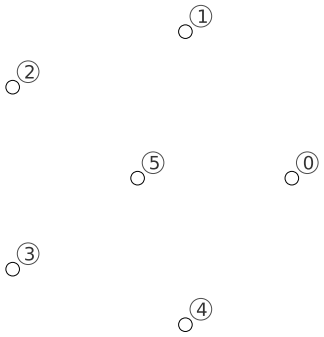


Figure 27: *5 points in convex position with 1 central point.* Point-configuration with 5 points on a circle and 1 point in the center.

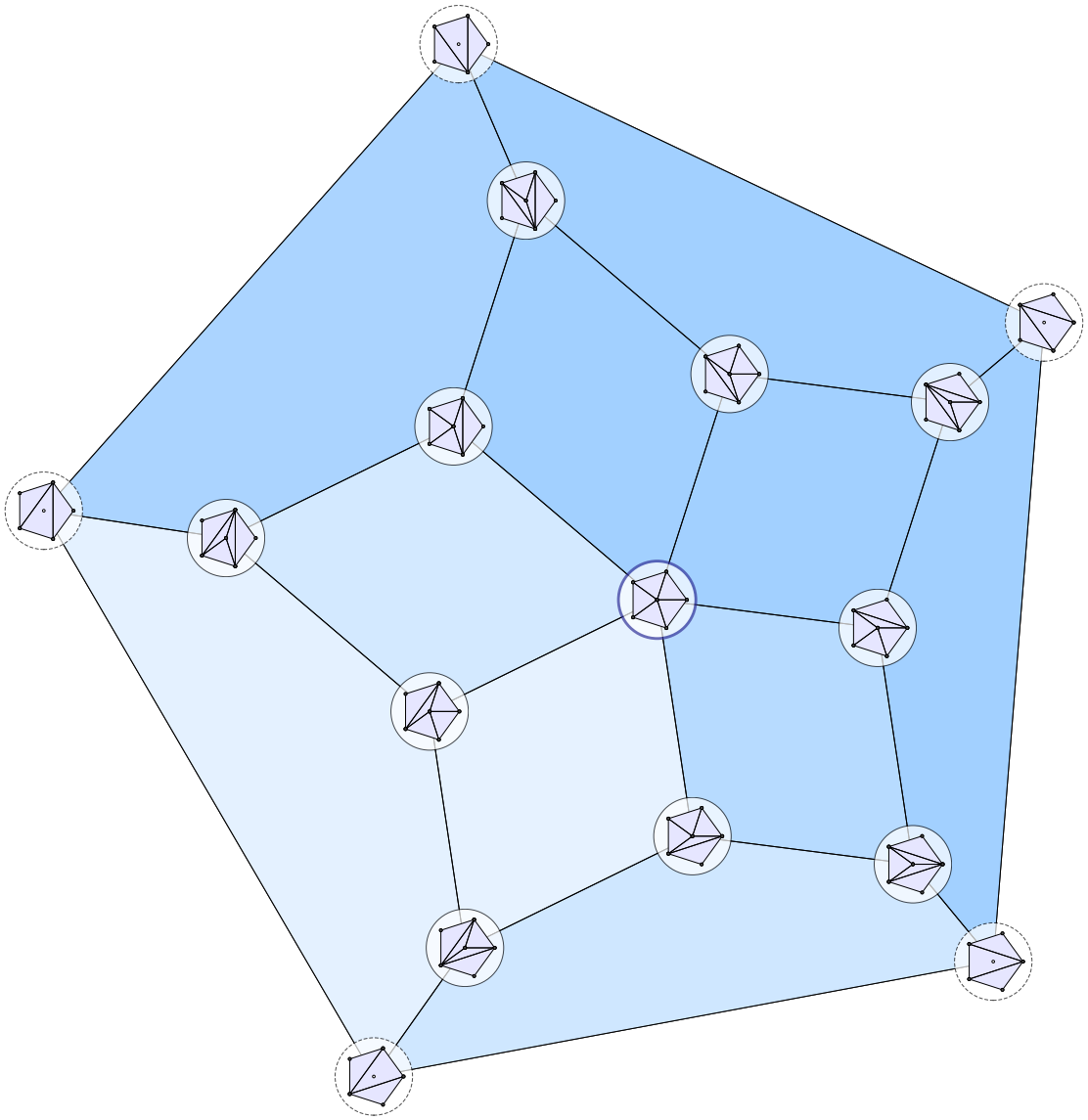


Figure 28: *Secondary polytope of 5 points in convex position with 1 central point.* Secondary polytope of the point-configuration in Figure 27. The 5 triangulations marked with a dotted circle are equal to the 5 triangulations of the 5 points in convex position in Figure 21. The facet on the backside (which contains the vertices corresponding to the marked triangulations) equals the secondary polytope of the 5 points (see Figure 22).

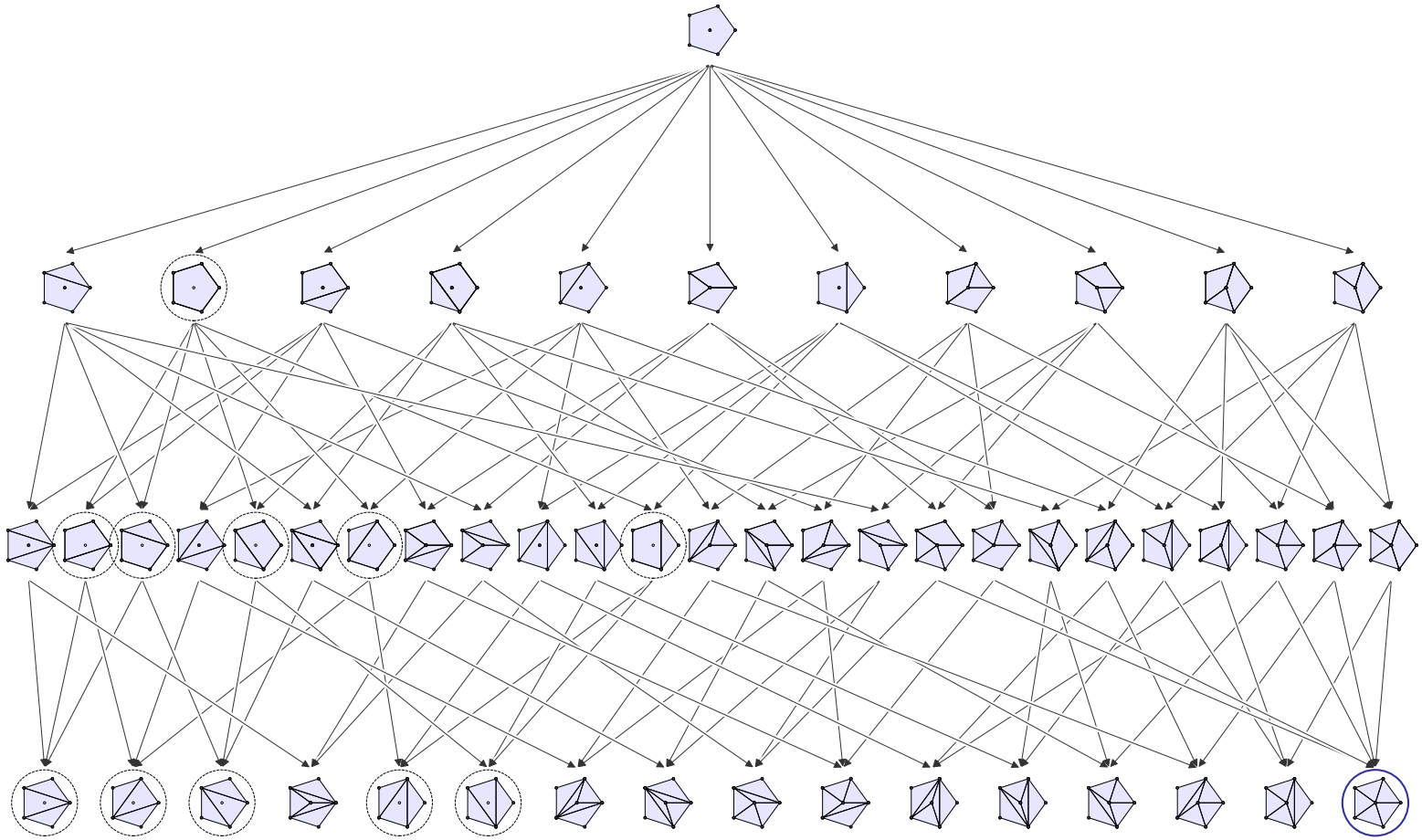


Figure 29: *Refinement poset of 5 points in convex position with 1 central point.* Refinement poset of the point-configuration in Figure 27. The subdivisions marked with a dotted circle equal the subdivisions of the 5 points in convex position in Figure 21. So, the subdivision poset from the 5 points (Figure 23) can be seen as a sub-poset of the subdivision poset in this figure.

4.2.2 Mother of all examples

The so-called "mother of all examples" is an important and famous example. It is described extensively in [7, chapter 7.1]. The mother of all examples only consists of 6 points in the plane (Figure 30), but it has already non-regular subdivisions. Therefore, it is especially interesting for the study of non-regular subdivisions and their relationship to the secondary polytope, because the secondary polytope is still 3-dimensional and thus easy to visualize and analyze. The point-configuration consists of 3 points on a circle and another 3 points on a smaller circle. The number of triangulations is 18 (16 regular and 2 non-regular) and the number of overall subdivisions is 65 (51 regular and 14 non-regular).

The secondary polytope (Figure 31) is 3-dimensional and has one hexagonal facet F. F corresponds to the subdivision $\mathscr{S}$ containing 3 quadrangular cells and 1 triangular cell (as can also be seen in Figure 32a). $\mathscr{S}$ is a full subdivision and therefore all refinements of $\mathscr{S}$ are also full. Moreover all full triangulations are refinements of $\mathscr{S}$. There are several non-regular subdivisions whose constrained secondary polytopes lie in $\mathscr{S}$ (as shown in Figure 32).

The refinement poset is split up into two parts in order to improve clarity. One part contains only the regular subdivisions (Figure 33), and one part only the refinements of $\mathscr{S}$ (Figure 34) which include all non-regular subdivisions. The refinement poset of regular subdivisions has again height 3, but the refinement poset (including the non-regular subdivisions) has now height 4.

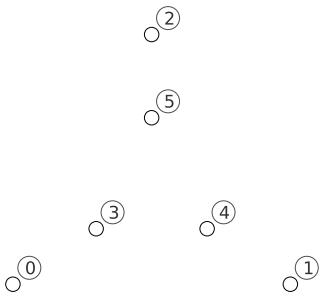


Figure 30: *Mother of all examples.* Point-configuration with 3 points on a big circle and 3 points on a smaller, concentric circle.

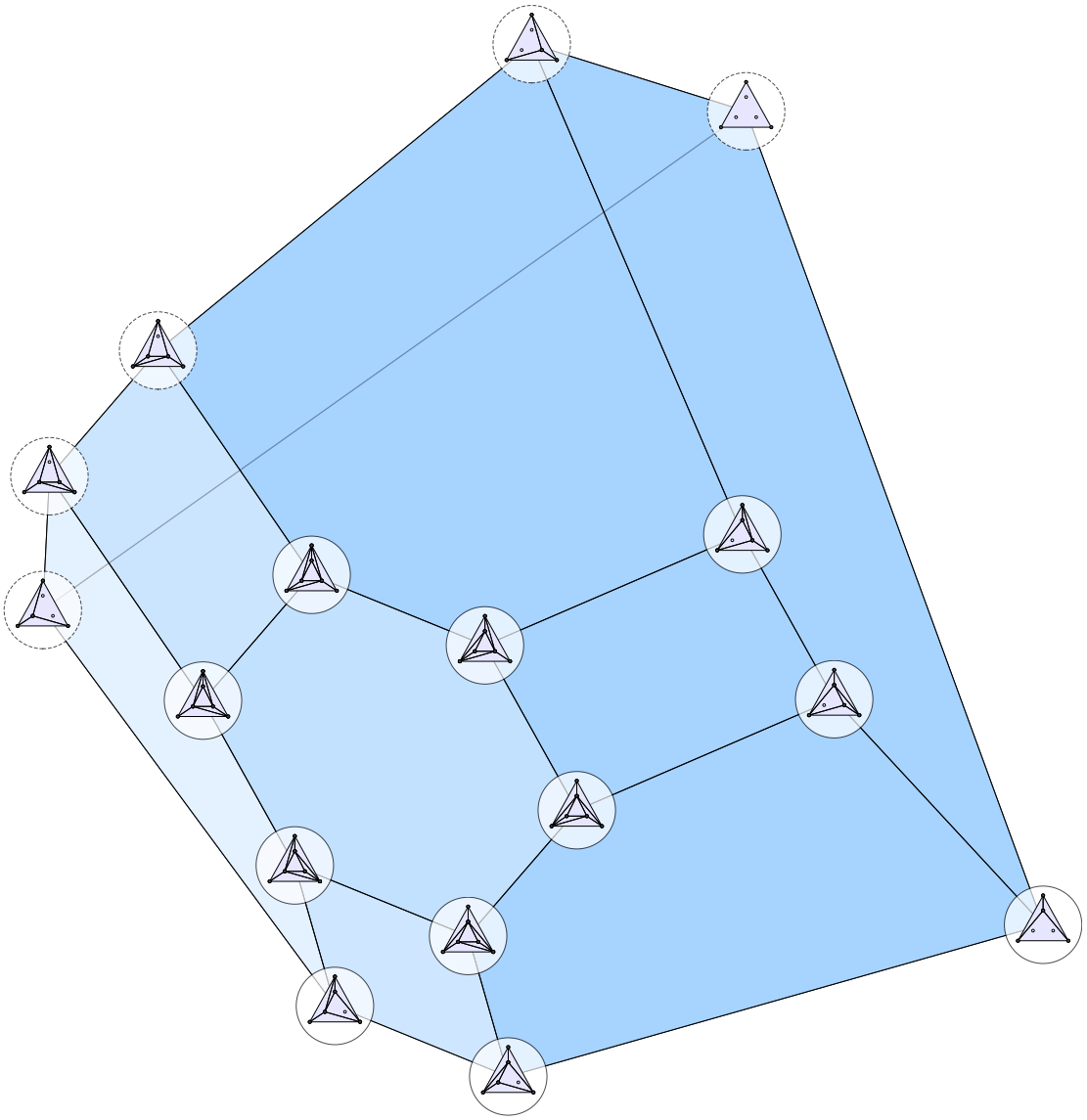
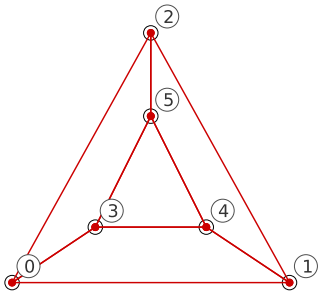
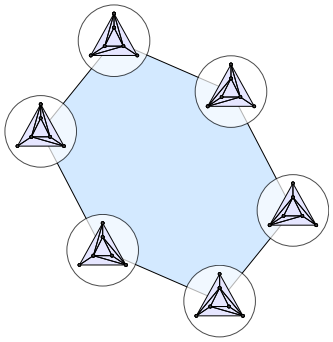


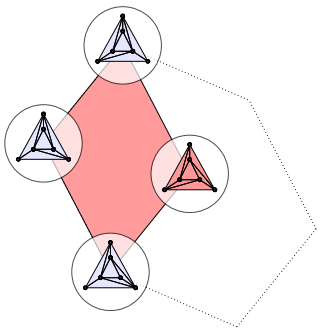
Figure 31: *Secondary polytope of the mother of all examples.* Secondary polytope of the point-configuration in Figure 30. The 5 triangulations marked with a dotted circle are equal to the 5 triangulations of the point-configuration in Figure 12a. The facet, containing all vertices corresponding to the 5 triangulations marked with a dotted circle, equals the secondary polytope in Figure 19 p.33.



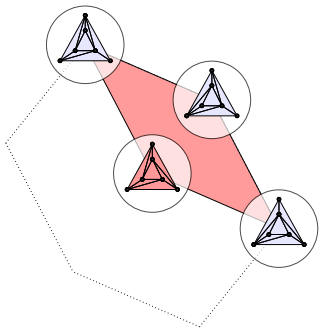
(a) Constraining subdivision



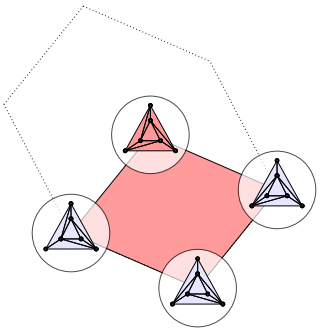
(b) Constrained secondary polytope



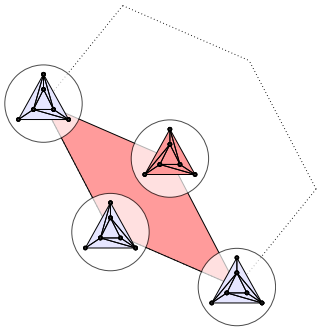
(c) Constrained secondary polytope 1



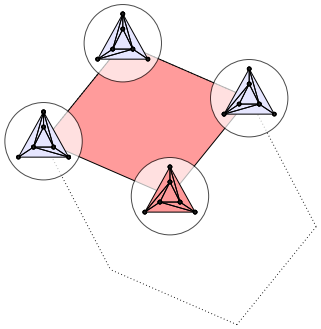
(d) Constrained secondary polytope 2



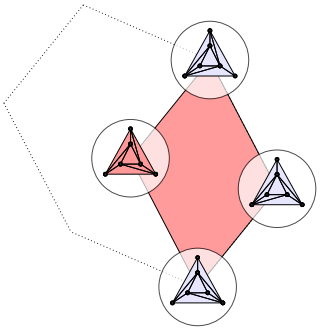
(e) Constrained secondary polytope 3



(f) Constrained secondary polytope 4



(g) Constrained secondary polytope 5



(h) Constrained secondary polytope 6

Figure 32: *Constrained secondary polytopes of the mother of all examples.* Figure 32b shows the constrained secondary polytope, constrained for the constraining subdivision in Figure 32a. There are several constrained secondary polytopes of non-regular subdivisions (painted red in Figure 32c - Figure 32h) that have the same dimension as Figure 32b, but also lie inside of it. There are 2 different non-regular triangulations that have GKZ-vectors lying in the relative interior of Figure 32b. Together with the 6 other triangulations whose GKZ-vectors are contained in Figure 32b they make up all 8 full triangulations.

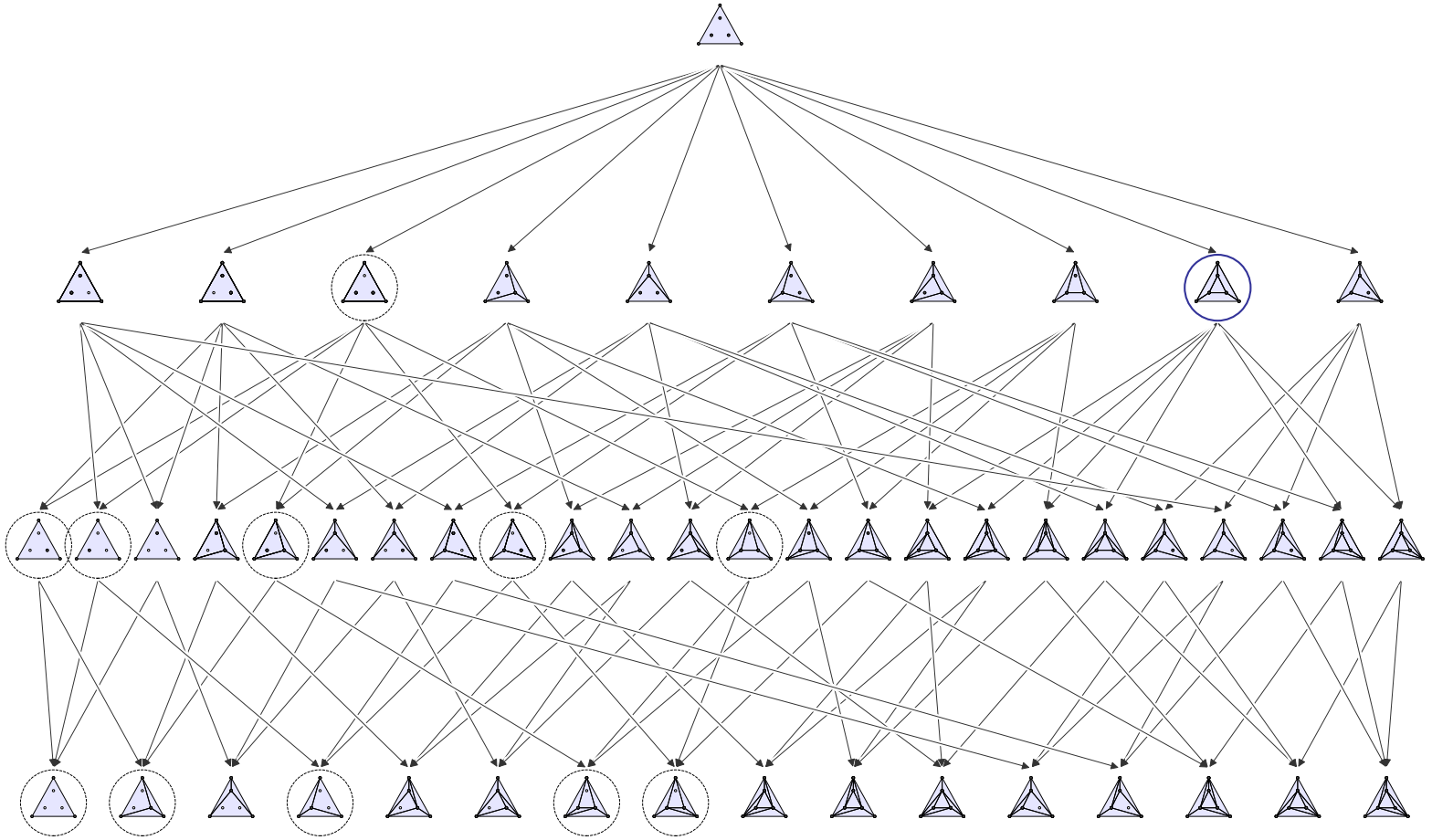


Figure 33: *Regular refinement poset of the mother of all examples.* Refinement poset of the regular subdivisions of the point-configuration in Figure 30. Non-regular subdivisions are not included. The subdivisions marked with a dotted circle equal the subdivisions of the point-configuration in Figure 12a p.22. Therefore, the refinement poset in Figure 14 p.24 is a sub-poset of the refinement poset in this figure.

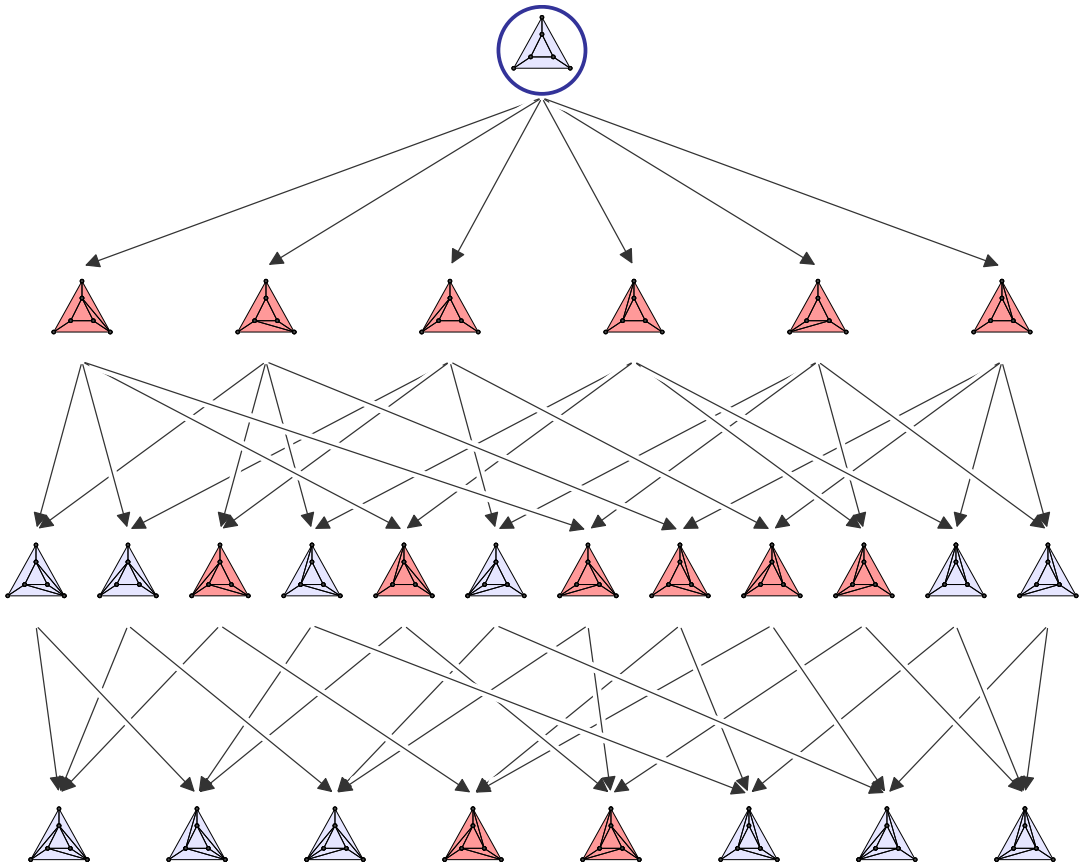


Figure 34: *Constrained refinement poset for the mother of all examples.* The constrained refinement poset for the point-configuration in Figure 30, constrained for the subdivision in Figure 32a. It contains all non-regular subdivisions (shown in red) of the point-configuration in Figure 30.

4.2.3 Twisted mother of all examples

This is a variation of the mother of all examples from Section 4.2.2. The 3 points lying on the smaller circle are now slightly rotated counter-clockwise (Figure 35). This has interesting effects on the secondary polytope and also on the property of regularity of a subdivision. One of the two non-regular triangulations of the mother of all examples now becomes regular. The other one stays non-regular and becomes cyclic with respect to the in-front-relation (as shown in Figure 11 p.20). The number of triangulations and subdivisions stays the same, but the distribution of regular and non-regular ones changes. The number of triangulations is again 18 (now 17 regular and 1 non-regular) and the number of overall subdivisions is again 65 (now 57 regular and 8 non-regular).

The secondary polytope (Figure 36) is almost equal to the secondary polytope of the mother of all examples (see Figure 31). But all GKZ-vectors of triangulations have moved slightly, and so the hexagonal facet is no longer present. One of the two GKZ-vectors of the former non-regular triangulations is now a vertex of the secondary polytope and therefore the corresponding triangulation is regular. The other one has moved into the relative interior of the secondary polytope (before it was in the relative interior of the hexagonal facet). There are also other subdivisions that have become regular and one subdivision $\mathscr{S}$ that has become non-regular. $\mathscr{S}$ is specially interesting and can be seen in Figure 37a. The constrained secondary polytope, constrained for $\mathscr{S}$ in Figure 37b is a cube. When we transform the point-configuration back to the point-configuration of the mother of all examples in Figure 30, the cube gets compressed until it becomes the already described hexagonal facet. With this procedure the change of the regularity property of some subdivisions can easily be seen.

The refinement poset is again split up into two parts. Both parts, the one containing only the regular subdivisions (Figure 38) and the one containing the refinements of $\mathscr{S}$ (Figure 39), are combinatorially equivalent to the parts of the mother of all examples. Overall, the refinement posets of the mother of all examples and of the twisted mother of all examples are combinatorially equal.

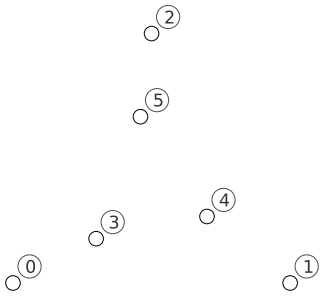


Figure 35: *Twisted mother of all examples.* Point-configuration with 3 points on a big circle and 3 points on a smaller, concentric circle. The inner 3 points are slightly rotated counter-clockwise.

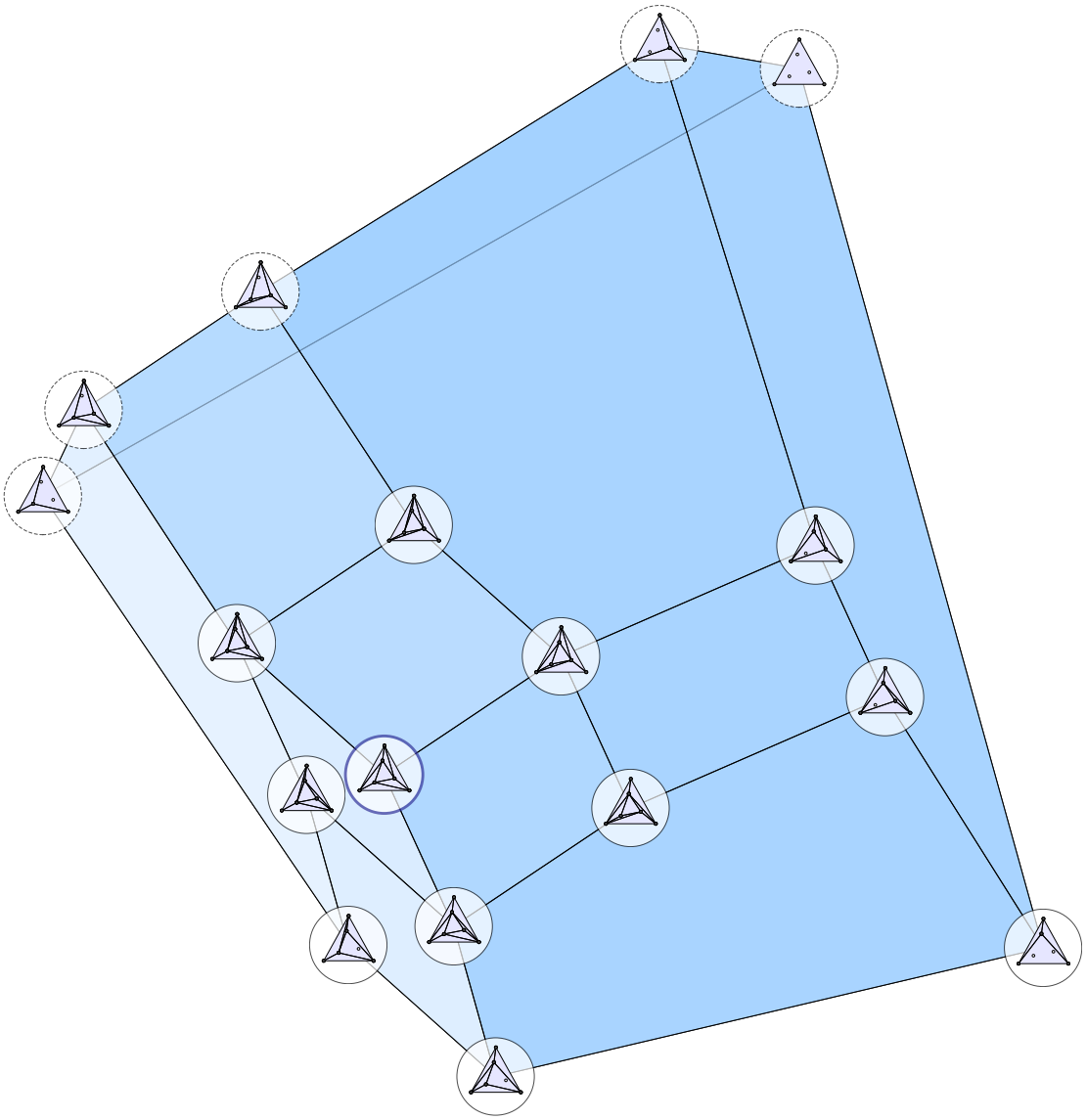
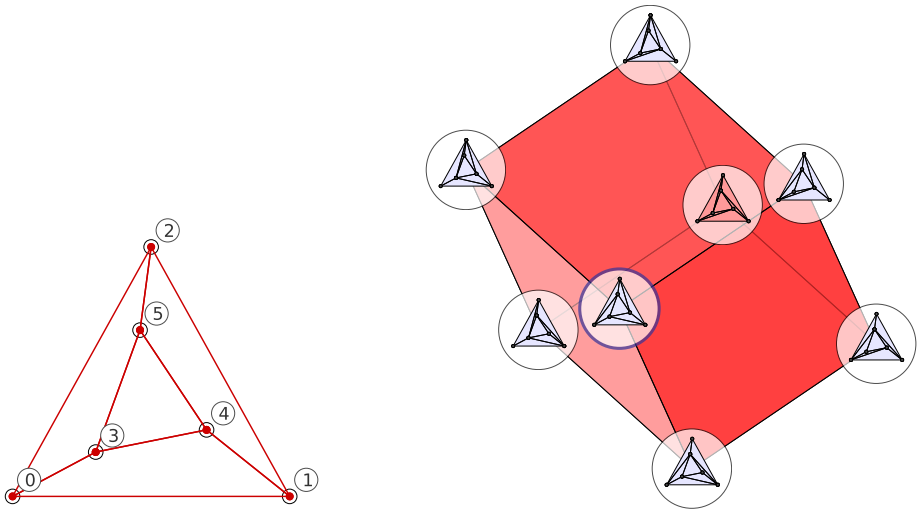


Figure 36: *Secondary polytope of the twisted mother of all examples.* Secondary polytope of the point-configuration in Figure 35. All facets, except the hexagonal facet, are still present. The hexagonal facet has split up into 3 facets in form of a parallelogram. The Delaunay triangulation (marked with a blue circle) has become regular and has now a corresponding vertex on the secondary polytope. The 5 triangulations marked with a dotted circle are combinatorially equal to the 5 triangulations of the point-configuration in Figure 12a p.22 (only the points have moved slightly). The facet, containing all vertices corresponding to the 5 triangulations marked with a dotted circle, is similar with the secondary polytope in Figure 19 p.33.



(a) Constraining subdivision **(b)** Constrained secondary polytope

Figure 37: *Constrained secondary polytope of the twisted mother of all examples.* Figure 37b shows the constrained secondary polytope for the constraining subdivision in Figure 37a. The subdivision in Figure 37a is non-regular. In Figure 37b every face visible from the front is also a face of the secondary polytope in Figure 36, and is therefore regular. All other faces (including the trivial one that corresponds to the subdivision in Figure 37a) are non-regular.

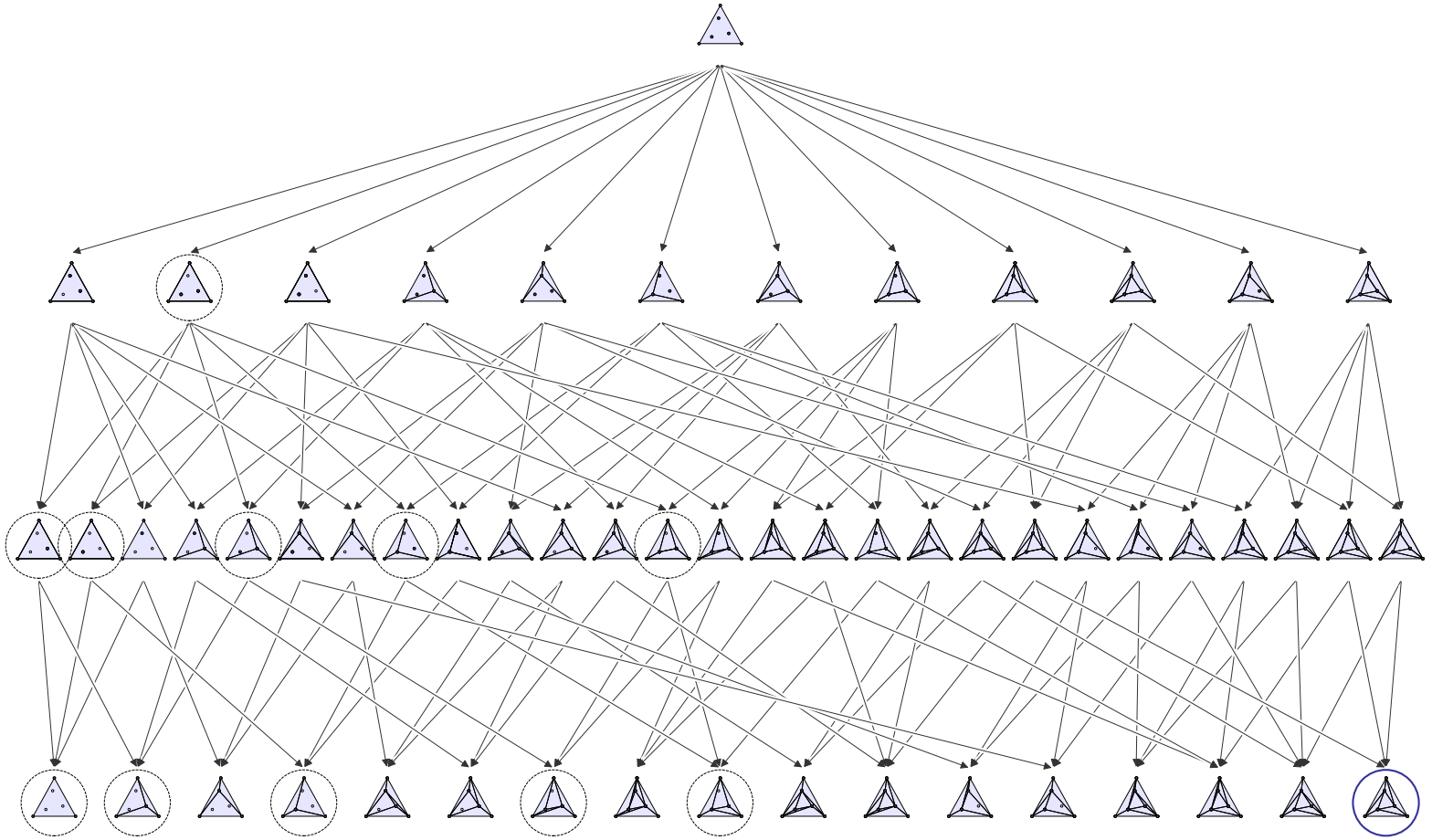


Figure 38: *Regular refinement poset of the twisted mother of all examples.* Refinement poset of the regular subdivisions of the point-configuration in Figure 35. Non-regular subdivisions are not included. The subdivisions marked with a dotted circle are combinatorially equal to the subdivisions of the point-configuration in Figure 12a p.22. Therefore the refinement poset in Figure 14 p.24 is isomorph to a sub-poset of the refinement poset in this figure.

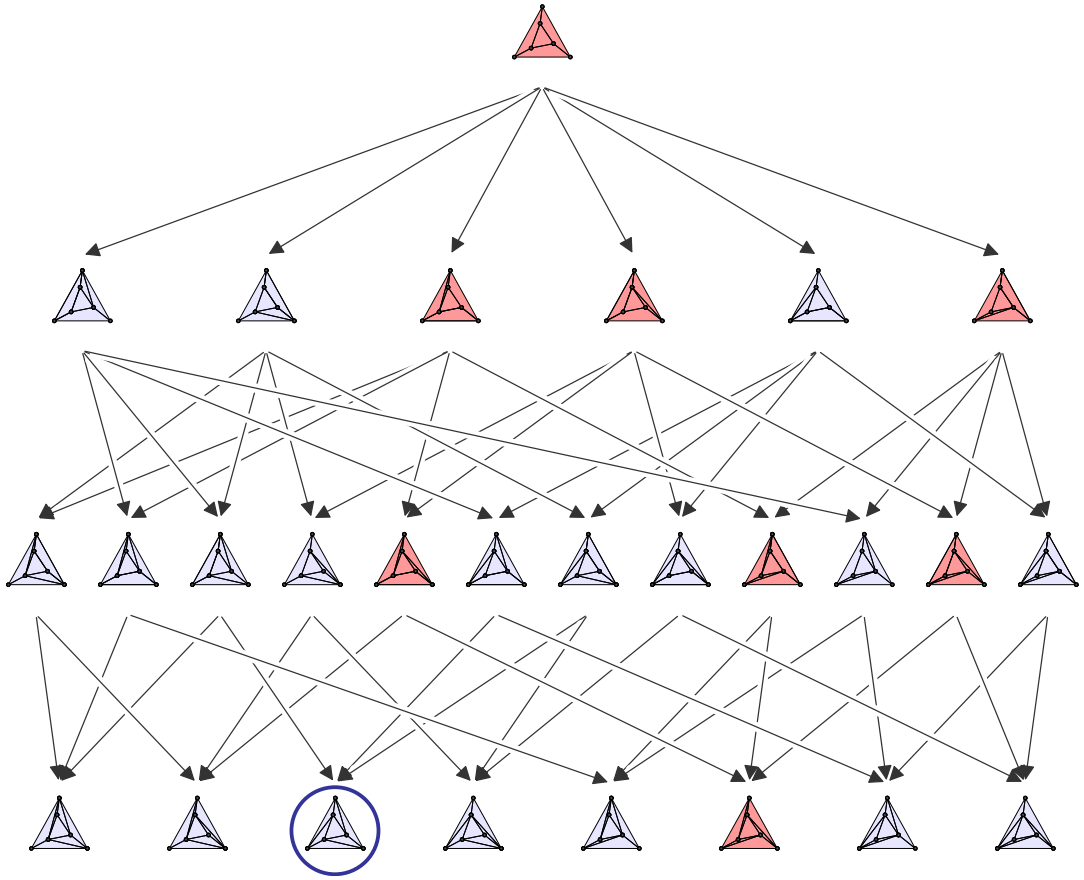


Figure 39: *Constrained refinement poset of the twisted mother of all examples.* The constrained refinement poset for the point-configuration in Figure 35, constrained for the subdivision in Figure 37a. It contains all non-regular subdivisions (shown in red) of the point-configuration in Figure 35.

5 Flip-path Algorithm

So far we were mostly concerned with the structure of the secondary polytope and its properties. The secondary polytope already proofs the connectedness of all regular triangulations of a point-configuration. Maybe more connected subclasses can be found with the help of the (constrained) secondary polytope. But once we know that there exists a flip-path from a start to a target triangulation, the question is how to find such a path.

If one triangulation is reachable via flips from another triangulation, in most cases there will be a multitude of different flip-paths that connect the two triangulations. Most of the time one would probably want to find the shortest path. But there are currently no known fast algorithms that find the shortest path. Of course one can always enumerate all triangulations, construct the complete flip-graph, and perform a graph search algorithm (e.g. breadth-first search). But for longer paths this won't have a good performance.

For many situations one might also be satisfied with a short flip-path, even if it is not the shortest one. We present an algorithm that is based on linear optimization on the secondary polytope, which finds a flip-path between two regular triangulations. We also analyze the algorithm's runtime and the length of the path it finds. For an overview of linear optimization see [16].

5.1 Structure of the algorithm

The algorithm uses the fact that the secondary polytope is a polytope and thus linear optimization can be performed on it. The algorithm finds a flip-path from a regular start triangulation to a regular target triangulation.

Let's assume we have already computed the secondary polytope. Every regular triangulation corresponds to a vertex on the secondary polytope. So we can take the vertex *vertex_start* corresponding to the start triangulation, and the vertex *vertex_target* corresponding to the target triangulation. We now search for a path from *vertex_start* to *vertex_target* on the 1-skeleton of the secondary polytope. To find this we take an optimization vector (*opt_vector*) for which a linear program is maximized in *vertex_target*. This means that *opt_vector* must lie in the outer normal cone of *vertex_target*. The algorithm then performs a linear optimization on the secondary polytope for *opt_vector*. It starts at *vertex_start*, and because of the choice of *opt_vector*, it will halt in *vertex_target*. The sequence of vertices it encounters on the way represents the sought-after flip-path. The algorithm is sketched in Figure 40.

```
1: function find_path(vertex_start, vertex_target)
2:     opt_vector ← vector_from_normal_cone(vertex_target)
3:     vertex_current ← vertex_start
4:     vertex_list ← list()
5:     while vertex_current ≠ vertex_target do
6:         max_scalar_product ← −∞
7:         for vertex_adjacent in vertex_adjacencies(vertex_current) do
8:             scalar_product ← opt_vector · vector(vertex_adjacent)
9:             if scalar_product > max_scalar_product then
10:                max_scalar_product ← scalar_product
11:                vertex_max ← vertex_adjacent
12:            end if
13:        end for
14:        vertex_current ← vertex_max
15:        vertex_list.append(vertex_max)
16:    end while
17:    return vertex_list
18: end function
```

Figure 40: *Optimization algorithm.* Pseudo-code for the linear optimization algorithm.

Description of the Algorithm in Figure 40:

Line 1: The algorithm receives the vertices *vertex_start* and *vertex_target* as input parameters.

Line 2: A vector from the normal cone of *vertex_target* is assigned to *opt_vector*.

Line 3: The currently considered vertex *vertex_current* is at the beginning *vertex_start*.

Line 4: A list *vertex_list* for the vertices of the path is created.

Line 5: The process of the step-wise optimization gets repeated until *vertex_target* is reached.

Line 6: For the optimization, the maximum of all scalar products (computed in Line 8) *max_scalar_product* is saved. At the start it is set to minus infinity.

Line 7: Every vertex *vertex_adjacent* adjacent to the current vertex *vertex_current* is considered separately in this loop.

Line 8: The scalar product of *opt_vector* and the vector representation of the adjacent vertex *vertex_adjacent* is calculated.

Line 9-11: When the scalar product of the present adjacent vertex is greater than the former maximum of the scalar products, *max_scalar_product* is updated and the adjacent vertex is saved as *vertex_max*.

Line 14-15: The vertex with the highest scalar product is the new current vertex *vertex_current* and is appended to the path-list *vertex_list*.

Line 17: The path-list is returned.

Most of the time, the algorithm won't be very attractive if one has to calculate the secondary polytope in advance. But the algorithm can also work without the secondary polytope. Each vertex that corresponds to a regular triangulation has the coordinates of the GKZ-vector of this triangulation. So we can work with the actual triangulations and their GKZ-vectors instead of working with the vertices.

For a triangulation $\mathscr{T}$ and a corresponding vertex v, the adjacent vertices v_i correspond to triangulations $\mathscr{T}_i$ that can be reached from $\mathscr{T}$ via a regular flip. Whether a flip is possible or not can be tested with a simple determinant test in the point-configuration. When the point-configuration is 2-dimensional and in convex position, every possible flip is regular. Otherwise, one has to make sure that only regular flips are considered. This assures that the algorithm only uses regular triangulations, since a regular flip always flips to a regular triangulation. If non-regular triangulations would be used, the algorithm could get stuck.

5.2 Quality of the algorithm

When we analyze the flip-path that is generated from the algorithm, the biggest quality criterion is most likely the length of the flip-path. A path as short as possible is probably desired. There always exists a shortest flip-path between two triangulations. So it is a good criterion to compare the length of the flip-path generated by the algorithm to the length of the shortest possible flip-path.

Since the secondary polytope deforms when we move the points of the point-configuration, the scalar products in the algorithm change, and therefore the flip-path can change too. That means, that if we move the points of a point-configuration, the generated flip-path can be different from the generated flip-path of the original point-configuration, even if the set of triangulations doesn't change. This is somehow undesirable because the shortest flip-paths of point-configurations that have the same set of triangulations are the same. But the flip-path generated by the algorithm could get longer and longer when we move the points.

For a point-configuration $\mathcal{A}$, we can always use the algorithm on another point-configuration $\mathcal{A}'$ that has the same set of triangulations as $\mathcal{A}$. The algorithm will generate a valid flip-path for $\mathcal{A}$ that might be shorter as if we had used the algorithm on $\mathcal{A}$. Especially for 2-dimensional point-configurations in convex position this is helpful. As long as n points are in convex position, they have the same set of triangulations, independent of the position of the points. Therefore, we can use the algorithm on n points on a circle, and so the secondary polytope will have a high amount of symmetry and its edges will have relative evenly distributed lengths.

For different point-configurations and all pairs of triangulations on them, the following statistics compare the length of the algorithm's flip-path and that of the shortest possible flip-path between a pair of triangulations. The shortest flip-path between two triangulations is computed with a breadth-first search on the 1-skeleton of the secondary polytope (which equals the flip-graph). For the target vertex $vertex\_target$, the optimization vector is taken as the arithmetic mean of all normalized normal vectors of faces that are adjacent to $vertex\_target$.

The table in Figure 41 and the diagrams in Figure 42 show the evaluation for 6 different point-configurations. Exact descriptions of the provided numbers and graphics can be found in the figure captions.

The first 4 examples are all 2-dimensional point-configurations with points evenly distributed on a circle. For *6 points on a circle* the flip-paths from the algorithm all have the same lengths as the shortest flip-paths. At *7 points on a circle* the first deviations occur. 9 percent of the triangulation-pairs have a shortest flip-path with length 5, but the algorithm yields a flip-path with length 6. Also on *8 points on a circle* and *9 points on a circle* deviations occur and one can assume that on n points on a circle for $n \geq 7$ the percentage of deviations will increase. But the higher the deviation of the lengths is, the lower is the percentage of flip-paths that have this deviation. So for some applications the algorithm might be a valuable compromise between the amount of deviations and its time-complexity (which will be shown later).

In this evaluation, *9 points on a circle* is the example with the most points. There is no example with more points, because for a convex point-configuration with n points in 2 dimensions the number of triangulations (all of them are regular) is the $(n-2)$th catalan-number $C_{n-2} = \frac{1}{n-1}\binom{2(n-2)}{n-2}$, and the number of triangulation-pairs is $\binom{C_{n-2}}{2}$. For $n = 9$ this means that already 91.806, and for $n = 10$, 1.021.735 triangulation pairs exist. So the size of the evaluated examples is limited by computing power, time, and time-complexity.

There are also two examples of point-configurations where the points do not lie on a circle. The first is *6 points in convex position*. The comparison between its flip-path evaluation and those of *6 points on a circle* is interesting. When the points are not evenly distributed on a circle, the flip-path lengths have deviations, and one can see that the algorithm produces much better results for points on a circle. The second example is the *mother of all examples* which also has deviations in the flip-path lengths.

length of shortest path	0		1		2		3		4		5		6		7		8		9	
6 points on a circle																				
count	14		42		72		60		8											
length +0	14	100%	42	100%	72	100%	60	100%	8	100%										
7 points on a circle																				
count	42		168		392		560		448		154									
length +0	42	100%	168	100%	392	100%	560	100%	448	100%	140	90%								
length +1											14	9%								
8 points on a circle																				
count	132		660		1920		3696		4784		4056		1936		240					
length +0	132	100%	660	100%	1920	100%	3696	100%	4784	100%	3944	97%	1672	86%	216	90%				
length +1											112	2%	248	12%	16	6%				
length +2													16	0%	8	3%				
9 points on a circle																				
count	429		2574		8910		21120		36060		45090		40122		22890		6480		366	
length +0	429	100%	2574	100%	8910	100%	21120	100%	35996	99%	44350	98%	37212	92%	18918	82%	5022	77%	252	68%
length +1									64	0%	740	1%	2784	6%	3636	15%	1272	19%	114	31%
length +2													126	0%	318	1%	132	2%		
length +3															18	0%	36	0%		
length +4																	18	0%		
6 points in convex position (not on a circle)																				
count	14		42		72		60		8											
length +0	14	100%	42	100%	69	95%	46	76%	7	87%										
length +1					3	4%	10	16%	1	12%										
length +2							4	6%												
mother of all examples																				
count	16		48		84		78		30											
length +0	16	100%	48	100%	78	92%	60	76%	24	80%										
length +1					6	7%	12	15%	6	20%										
length +2							6	7%												

Figure 41: *Optimization algorithm - path length table.* Table of the path length evaluation for the algorithm in Figure 40. In the first row is a separation into classes of pairs of regular triangulations with equal length l of their shortest flip-paths. Then follow various examples of point-configurations. In a row named "count" and column l is written the number of triangulation-pairs where the shortest flip-path has length l. In a row named "length + i" and column l is written the number of triangulation-pairs where the algorithm's flip-path has length $l+i$, and the percentage of those triangulation-pairs.

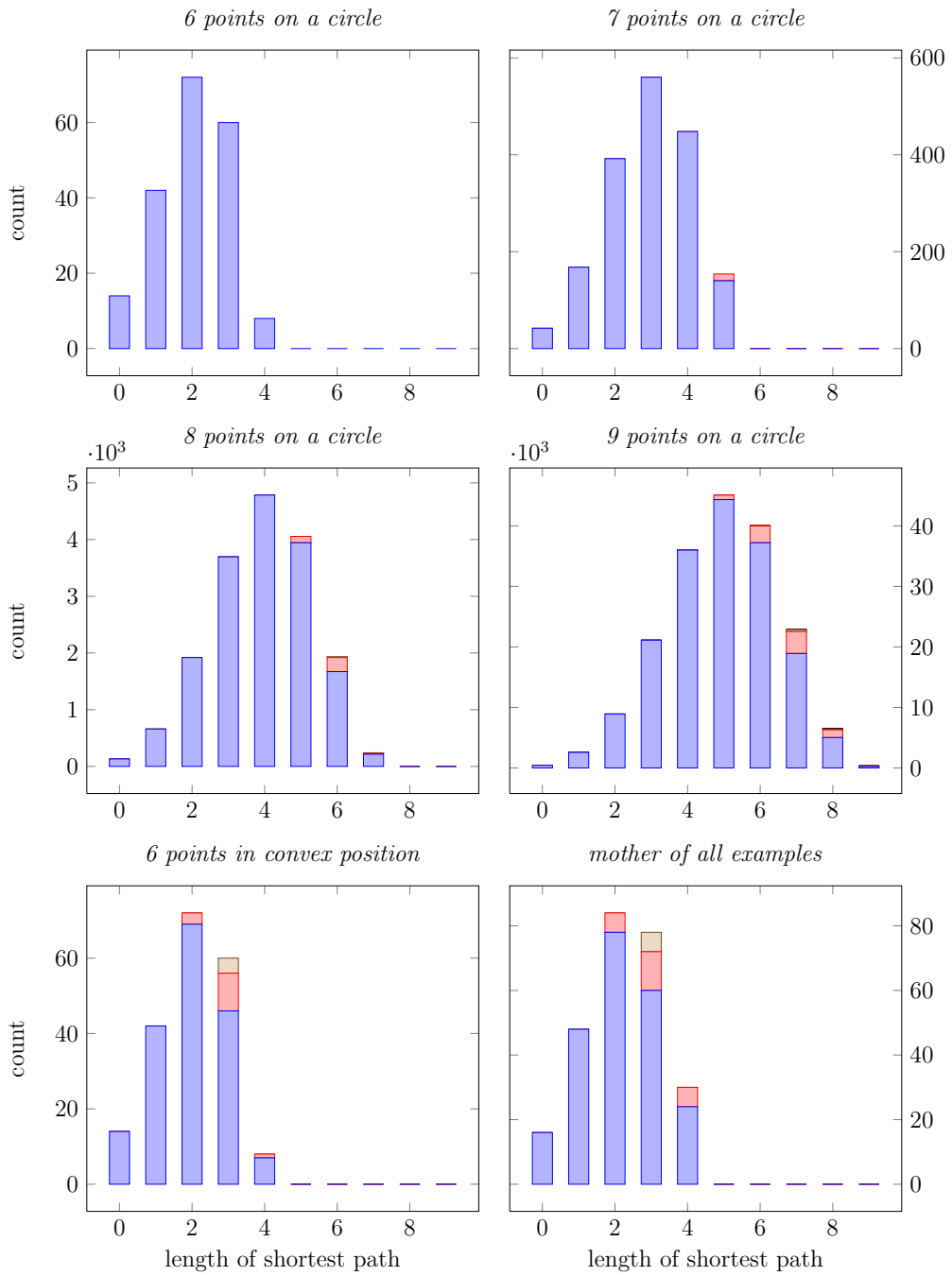


Figure 42: *Optimization algorithm - path length diagrams.* Diagrams of the path length evaluation for the algorithm in Figure 40. The different examples of point-configurations are shown in their own diagrams. Each diagram shows one stack of bars for each class of triangulation-pairs where the shortest flip-path has length l. The bars represent the number of triangulation-pairs where the algorithm yields a flip-path with length $l + i$, and they are stacked on each other for increasing i.

5.3 Runtime of the algorithm

After the length of the algorithm's flip-paths, the runtime is the second important criterion. Since the algorithm is especially useful for 2-dimensional point-configurations in convex position, we only analyze this case. This case has also some properties that make the runtime analysis easier, and we will describe some ways to optimize the algorithm in order to achieve the runtimes that are presented.

We consider a 2-dimensional point-configuration with n points in convex position. The secondary polytope has dimension $(n-3)$ but is embedded in $\mathbb{R}^n$, every triangulation is regular, and $(n-3)$ flips can be performed on every triangulation. The neighborhood of every vertex of the secondary polytope looks like a $(n-3)$-dimensional cone with $(n-3)$ edges. This means that there are $\binom{n-3}{n-2} = n-3$ facets in the cone.

As already mentioned, the algorithm doesn't need to compute the actual secondary polytope. It only needs the vertices of the secondary polytope, so it is sufficient to work with the GKZ-vectors of the triangulations. Every GKZ-vector has n components. For GKZ-vectors v and v' of triangulations that are flip-neighbors we call the vector $v - v'$ the GKZ-vector of the directed edge from v to v'. When a flip is performed in a triangulation, exactly 4 components of the GKZ-vector of the triangulation change (because the triangulation is 2-dimensional and for 4 points of the point-configuration the incidence relations to triangles change). This means, that the GKZ-vector of the corresponding directed edge has only 4 components which are not zero.

We will now evaluate the runtime for different stages:

Optimization vector:

The algorithm must first compute the optimization vector *opt_vector*. Let v be the GKZ-vector of the target triangulation and v_i the GKZ-vectors of the $(n-3)$ flip-neighbors of the target triangulation. We compute $v_i' = v - v_i$ which are the GKZ-vectors of each directed edge from v to one v_i. Only 4 components of a v_i' are not zero and so it can be calculated in $O(1)$ time. All v_i' can be calculated in $O(n)$ time. Since the secondary polytope has dimension $(n-3)$, we can compute 3 linear independent normal vectors w_j for the affine hull of the secondary polytope. This can be done by solving the linear equation $A \cdot \vec{x} = \vec{0}$ where A consists of the v_i' as row-vectors. Solving the linear equation takes $O(n^3)$ time. For the optimization vector, we want to compute the normal vectors of the facets of the normal-cone of v. For a facet, we can calculate the linear equation $B \cdot \vec{y} = \vec{0}$ where

B consists of the row-vectors w_j and the $(n-3)$ vectors v_i' where the corresponding v_i lie in the facet. A vector in the resulting space is a normal vector to the facet. The equation-solving takes again $O(n^3)$ time. For all facets, this takes $O(n^4)$ time. At last, the normal vectors are normalized, summed up, and divided by $(n-3)$. This takes $O(n^2)$ time and gives a vector in the normal cone of v that will be taken as *opt_vector*. In summary, the computation of the optimization vector takes $O(n) + O(n^3) + O(n^4) + O(n^2) = O(n^4)$ time. Solving the linear-equations is the most time-consuming operation, but most row-vectors of the linear-equation matrices consist of only zeros, except for 4 components. So we can expect that the runtime is better in practice.

Note that, when flip-paths from different start triangulations to the same target triangulation have to be computed, the optimization vector only needs to be computed once.

Preparations for the start triangulation:

After the optimization vector has been calculated, the algorithm starts in the GKZ-vector of the start triangulation. Let now v be the GKZ-vector of the start triangulation and v_i the GKZ-vectors of the $(n-3)$ flip-neighbors of the start triangulation. The calculation of all $v_i' = v - v_i$ takes again $O(n)$ time. Instead of calculating the scalar products $\mathit{opt\_vector} \cdot v_i$ for the optimization, better performance can be achieved by calculating the scalar products $s_i = \mathit{opt\_vector} \cdot v_i'$. Since every v_i has only 4 components that are not zero, computation of one scalar product takes $O(1)$ time, and computation of all scalar products takes $O(n)$ time. In summary, the preparations for the start triangulation take $O(n)+O(n) = O(n)$ time.

Optimization step:

The last part of the evaluation is concerned with the optimization step. Assume that the algorithm is currently in a triangulation $\mathscr{T}$ and has a list of scalar products. Taking the maximum of the list takes $O(n)$ time. The algorithm can now move on to the next triangulation $\mathscr{T}'$ that corresponds to the maximum of the scalar products. The list of scalar products now needs to be updated. The step from $\mathscr{T}$ to $\mathscr{T}'$ corresponds to an edge exchanging flip (because the points are in convex position) located inside a quadrangle. All flips, except for 5, that are possible in $\mathscr{T}$, are also possible in $\mathscr{T}'$. These 5 flips are the already performed flip from $\mathscr{T}$ to $\mathscr{T}'$ and the 4 flips that exchange edges that are part of the quadrangle. Instead of them, the reverse-flip of the already performed flip and 4 other flips on

the quadrangle are possible in $\mathscr{T}'$. The scalar products corresponding to these 5 flips must be updated, but all other flips stay the same and so their scalar products also stay the same. Calculation of one scalar product takes $O(1)$ time, but finding the value in the list takes $O(n)$ time. So the update time is $O(5n) = O(n)$. Now the optimization step can be repeated until the target triangulation is reached. In summary, the optimization step takes $O(n) + O(n) = O(n)$ time. For k optimization steps that are needed to reach the target triangulation (which means that the generated flip-path has length k), the optimization steps take $O(kn)$ time.

This means that the preparation time $O(n)$ from the last stage is irrelevant. The whole process could probably be accelerated, when the list of scalar products is sorted (or stored in a heap). But the theoretical upper bound of the time-complexity is hard to decrease because of the update task.

Summary:

In summary, for a 2-dimensional point-configuration with n points in convex position and a given target triangulation the algorithm needs a preprocessing time of $O(n^4)$ (that is, as described, expected to be faster in practice) and a computation time of $O(kn)$ for a flip-path with length k from any start triangulation to the target triangulation.

6 Conclusion

The problem, whether the flip-graph of triangulations of a point-configuration is connected or not, exists for a long time. A lot of research was put into solving this problem, and results for various classes of flip-graphs have been achieved. To gain further insight into this topic, advanced structures might be needed. One important structure for researching the flip-graph is the secondary polytope. We have presented an overview of most of the important definitions and concepts necessary to build and analyze the secondary polytope. The definitions start bottom up and hardly any previous knowledge is required. Additionally, many figures are provided in order to facilitate understanding and intuition. The precision and the completeness of the definitions reduce ambiguities.

We have also established the definition of the constrained secondary polytope. This polytope is related to a subset of the flip-graph. In contrast to the secondary polytope, from where only information about regular triangulations can be gained, in the constrained secondary polytope non-regular triangulations can also be analyzed. The concept of a constraint is already applied to former definitions (e.g. constrained subdivision function, constrained secondary fan) that were necessary for the secondary polytope.

Various examples of point-configurations are given, and their (constrained) secondary polytopes along with their subdivision posets are analyzed. The examples show again numerous figures for the structures.

Unfortunately, no direct results about the connectedness of flip-graphs were discovered. But various properties of constrained secondary polytopes were figured out. It remains to be seen if this properties lead to new results for flip-graphs of triangulations. Still many more properties remain to be discovered. Especially results about secondary polytopes can probably be carried over to constrained secondary polytopes. See Section 3.4 for possible future work on constrained secondary polytopes.

At last, an algorithm for finding flip-paths between regular triangulations is given. The algorithm works with linear optimization on the secondary polytope. The quality of the generated flip-paths is evaluated by comparing them to the shortest possible flip-paths. It turns out that the algorithm performs best for 2-dimensional point-configurations in convex position. Additionally, a runtime analysis for the case of 2-dimensional point-configurations in convex position is given.

List of Figures

References

[1] F. Aurenhammer. Voronoi diagrams — a survey of a fundamental geometric data structure. *ACM Comput. Surv.*, 23:345–405, September 1991.

[2] F. Aurenhammer and R. Klein. Voronoi diagrams. *Handbook Computational Geometry*, pages 201–290.

[3] M. Bern and D. Eppstein. Mesh generation and optimal triangulation. *Computing in Euclidean geometry*, 1(1):23–90, 1992.

[4] L. J. Billera, P. Filliman, and B. Sturmfels. *Constructions and complexity of secondary polytopes*. Technical report (Cornell University. Mathematical Sciences Institute). Mathematical Sciences Institute, Cornell University, 1990.

[5] B. Bollobas. *Graph Theory, An Introductory course*. Springer-Verlag, New York, Heidelberg, Berlin, 1 edition, 1979.

[6] P. Bose and F. Hurtado. Flips in planar graphs. *Comput. Geom. Theory Appl.*, 42:60–80, January 2009.

[7] J. A. De Loera, J. Rambau, and F. Santos. *Triangulations - Structures for Algorithms and Applications*. Springer, Wiesbaden, 1st edition. edition, 2010.

[8] H. Edelsbrunner. An acyclicity theorem for cell complexes in d dimensions. In *Proceedings of the fifth annual symposium on Computational geometry*, SCG '89, pages 145–151, New York, NY, USA, 1989. ACM.

[9] H. Edelsbrunner, M. J. Ablowitz, S. H. Davis, E. J. Hinch, A. Iserles, J. Ockendon, and P. J. Olver. *Geometry and Topology for Mesh Generation (Cambridge Monographs on Applied and Computational Mathematics)*. Cambridge University Press, New York, NY, USA, 2006.

[10] S. Fortune. Voronoi diagrams and delaunay triangulations. In Jacob E. Goodman and Joseph O'Rourke, editors, *Handbook of discrete and computational geometry*, pages 377–388. CRC Press, Inc., Boca Raton, FL, USA, 1997.

[11] I. M. Gelfand, M. M. Kapranov, and A. V. Zelevinsky. *Discriminants, resultants and multidimensional determinants*. Birkhäuser, Boston, MA, 1994.

[12] C. Lee. The associahedron and triangulations of the n-gon. *European J. Combin.*, 10(6):551–560, 1989.

[13] L. Pournin and Th. M. Liebling. Constrained paths in the flip-graph of regular triangulations. *Comput. Geom. Theory Appl.*, 37:134–140, July 2007.

[14] F. Santos. A point set whose space of triangulations is disconnected. *J. Amer. Math. Soc.*, 13:611–637, 1999.

[15] F. Santos. Non-connected toric hilbert schemes. *Math. Annalen*, 332:645–665, 2005.

[16] A. Schrijver. *Theory of linear and integer programming.* Wiley-Interscience series in discrete mathematics and optimization. Wiley, 1998.

[17] D. D. Sleator, R. E. Tarjan, and W. P. Thurston. Rotation distance, triangulations, and hyperbolic geometry. *J. Amer. Math. Soc*, 1(3):647–681, 1988.

[18] R. P. Stanley. *Enumerative combinatorics.* Wadsworth Publ. Co., Belmont, CA, USA, 1986.

[19] C. Weibel. *Minkowski sums of polytopes: combinatorics and computation.* PhD thesis, école polytechnique fédérale de lausanne, 2007.

[20] G. M. Ziegler. *Lectures on polytopes.* Graduate texts in mathematics. Springer-Verlag, 1995.

Index