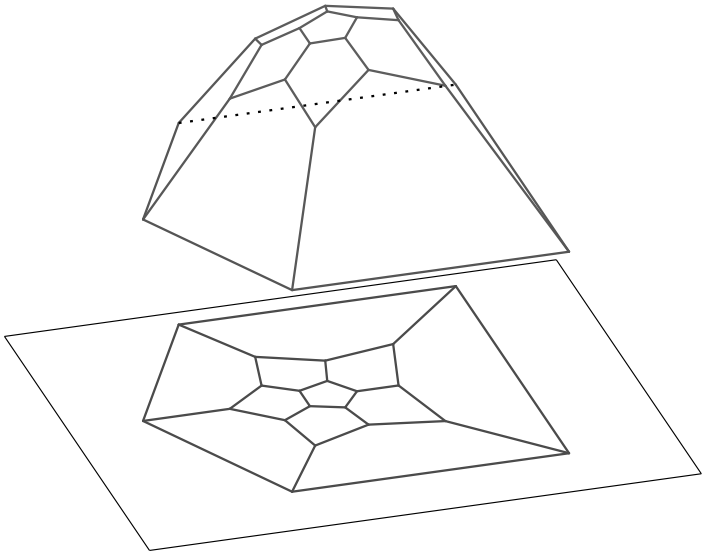
Lifting Planar Graphs to Realize Integral 3-Polytopes and Topics in Pseudo-Triangulations



Dissertation zur Erlangung des Doktorgrades
vorgelegt am 29. Februar 2008

Fachbereich Mathematik und Informatik
der Freien Universität Berlin

von
André Schulz

Author: Andé Schulz
Freie Universität Berlin
Institut für Informatik
Takustraße 9
14195 Berlin, GERMANY

Email:
schulza@mi.fu-berlin.de
Phone: +49 30-838-75150
Fax : +49 30-838-75192

Supervisor: Günter Rote
Freie Universität Berlin
Institut für Informatik
Takustraße 9
14195 Berlin, GERMANY

Email: rote@mi.fu-berlin.de
Phone: +49 30-838-75150
Fax : +49 30-838-75192

Referees: Günter Rote
Freie Universität Berlin
Institut für Informatik
Takustraße 9
14195 Berlin, GERMANY

Email: rote@mi.fu-berlin.de
Phone: +49 30-838-75150
Fax : +49 30-838-75192

Ileana Streinu
Smith College
Department of Computer Science
McConnell Hall 210
Northampton, MA 01063, USA

Email:streinu@cs.smith.edu
Phone: +01 413 585-3827
Fax : +01 413 585-3786

Defense on June 5, 2008.

Dedicated to my son, Peer.

Abstract

Lifting planar embeddings with equilibrium stress is a well known method that dates back to the 19th century. We discuss the known theory about liftings and develop a framework that allows us to apply the lifting technique easily. In this thesis we apply the lifting method to different geometric problems.

As a first application we show how to embed 3-polytopes with small integer coordinates. Our method improves the upper bound for the size of the largest coordinate from $O(2^{18n^2})$ to $O(2^{7.55n})$. A new generalized version of Tutte's *spring embedding* assures that a planar 2d-embedding contains an equilibrium stress and is therefore liftable. We point out connections between the size of the integral embedding and the number of maximal spanning forests a planar graph can have.

The second field of applications for the lifting technique are topics about pseudo-triangulations. Our main observation shows how to model regular triangulations as linear programs over the polytope of pointed pseudo-triangulations. We introduce an equivalent of the Delaunay triangulation for pointed pseudo-triangulations of simple polygons. Our approach is motivated by the paraboloid lifting of the Delaunay triangulation and the generalization of linear programs that compute the Delaunay triangulation in special cases. We also investigate the so-called canonical pointed pseudo-triangulation and study some of its geometric properties. Our observations lead to a new characterization of pointed pseudo-triangulations as embeddings of minimal rigid graphs that can balance a given load with positive edge weights.

The thesis contains also results on pseudo-triangulation problems that were not obtained with help of liftings. We show that a sequence of superpolynomial many convexifying flips exists that transform a lifted pseudo-triangulation into a maximal locally convex surface. This is obtained by constructing a simple polygon that realizes an improving flip sequence of length $n^{\Theta(\log n)}$ between two of its pointed pseudo-triangulation. Furthermore we show that (1) it is NP-hard to decide if a graph contains a pseudo-triangulation and (2) it is NP-hard to decide if a graph can be extended to a pseudo-triangulation with small vertex degree. Both decision problems are studied in different incarnations. We obtain a new and easier NP-completeness proof of the *triangulation existence problem*, one of the classic NP-complete triangulation problems.

Zusammenfassung

Das Heben von planaren Einbettungen mit Gleichgewichtsstress ist eine Methode, die bereits im 19. Jahrhundert Anwendung fand. Wir stellen diese Methode vor und erstellen Werkzeuge, welche es erlauben, das Heben von planaren Einbettungen effektiv auszuführen. In der Dissertation werden wir die Technik zum Heben planarer Graphen auf geometrische Problemstellungen anwenden.

Als erste Anwendung zeigen wir, wie man 3-Polytope mit kleinen ganzzahligen Koordinaten einbetten kann. Die vorgestellte Methode verbessert die bisherige obere Schranke für die größte Koordinate von $O(2^{18n^2})$ auf $O(2^{7.55n})$. Eine neue, verallgemeinerte Variante der Tutte-Einbettung stellt sicher, dass die planare 2d-Einbettung einen Gleichgewichtsstress besitzt und deshalb hochhebbar ist. Wir zeigen Zusammenhänge zwischen der Größe der Einbettung und der maximalen Anzahl von Spannbäumen auf planaren Graphen auf.

Das zweite Anwendungsgebiet für das Heben planarer Graphen sind Fragestellungen über Pseudo-Triangulierungen. Unser Hauptresultat zeigt, wie man reguläre Triangulierungen als lineares Programm über dem Polytop der gespitzten Pseudo-Triangulierungen modellieren kann. Wir führen eine Entsprechung der Delaunay Triangulierung für gespitzte Pseudo-Triangulierungen von einfachen Polygonen ein. Unser Ansatz basiert auf der Parabolid-Abbildung, sowie auf Verallgemeinerungen von linearen Programmen, welche die Delaunay-Triangulierung für spezielle Szenarien berechnen. Des Weiteren untersuchen wir die kanonische Pseudo-Triangulierung und diskutieren einige ihrer geometrischen Eigenschaften. Unsere Erkenntnisse führen zur Charakterisierung von gespitzten Pseudo-Triangulierungen als minimal starre Graphen, welche eine gegebene Last mit positiven inneren Kantengewichten ausgleichen können.

Die Dissertation enthält außerdem Resultate über Pseudo-Triangulierungen, welche nicht von der Hebe-Technik Gebrauch machen. Wir zeigen, dass eine Sequenz von super-polynomiell viele Konvexitäts-erzeugende Flips existiert, die eine polyedrische Pseudo-Triangulierung in eine lokal konvexe Oberfläche zu überführt. Dies wird erreicht, in dem wir ein einfaches Polygon konstruieren, welches zwei gespitzte Pseudo-Triangulierungen besitzt, zwischen denen eine verbessernde Flipsequenz der Länge $n^{\Theta(\log n)}$ existiert. Des Weiteren zeigen wir, dass es erstens NP-schwer ist zu entscheiden, ob ein geometrischer Graph eine Pseudo-Triangulierung enthält und zweitens, dass es ebenfalls NP-schwer ist zu entscheiden, ob sich ein geometrischer Graph zu einer Pseudo-Triangulierung mit kleinem Knotengrad erweitern lässt. Beide Probleme werden in verschiedenen Varianten untersucht. In diesem Zusammenhang erhalten wir auch einen neuen, vereinfachten Beweis der NP-Vollständigkeit des Triangulierungs-Existenz Problems.

Acknowledgments

I would like to thank all the people that supported me in writing this thesis. First of all I want to thank my advisor Günter Rote. I am very thankful that he introduced me to the topics I studied in this thesis. Also I would like to thank him for the support, the valuable comments, loads of interesting ideas, and for proofreading this thesis. My second thanks go to Ileana Streinu for agreeing to co-referee this thesis and for the inspiring discussions during the last five years.

I thank all members of the work group *Theoretical Computer Science* at Free University Berlin. Especially, I would like to thank my room mate Kevin. It was always great fun to work with him. I furthermore thank Frank for his advices and for proofreading, Klaus for showing Sandra and me Buch, Tamara for helping me out with red-tape issues and Christian for all the cake. Moreover, I thank David Orden for inviting me to Alcalá de Henares and for his hospitality. I thank Oswin Aichholzer and Francisco Santos for helping me with my future academic career. I thank the people of the pseudo-triangulation community for their collaboration and for inviting me to the pseudo-triangulation workshops.

I am grateful to my parents, my family, and my friends for their support during my years of studying.

Most of all I thank Sandra. For everything.

Contents

Chapter 1

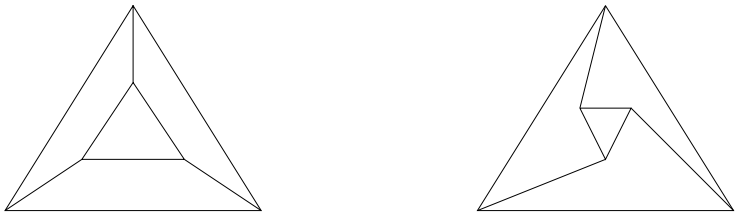
Introduction

The concept of *lifting* is an established technique to study geometric objects. Generally, a lifting takes a geometric object that is realized in d dimensions and assigns to it an additional coordinate by a lifting map. The new, enhanced object lives now in $(d+1)$-dimensional space. Such an object can be for example a polytope or an embedding of a graph. The orthogonal projection of the enhanced object back into the lower dimensional space gives the original object. Liftings are motivated by the fact that higher dimensional objects can store more "information" than the non-lifted counterpart. The goal is to apply existing techniques in the higher dimensional setting and trace back the results to the lower dimensional object. In this thesis we study liftings of planar graphs from $\mathbb{R}^2$ to $\mathbb{R}^3$.

Suppose we have a plane straight-line graph embedded in the xy-plane. We specify a lifting by giving every vertex of the graph a certain height. All vertices of a face have to lie on a common plane. The height of every point on a face that is not a vertex is interpolated. It makes only sense to speak about liftings, if we have a clear idea what we understand as *faces* of the plane geometric graph. Therefore, we restrict ourselves to planar 3-connected graphs, which guarantee a unique facial structure.

However, not all embeddings of planar graphs can be lifted in this way. The graph shown in Figure 1.1(a) depicts a liftable graph. The lifting corresponds to a tetrahedron with chopped off tip. It can be realized by giving the inner vertices the height 1, while leaving the outer vertices in the xy-plane. On the other hand, the same graph with a different plane realization is not liftable (Figure 1.1(b)).

Interestingly, there exists a powerful characterization which embeddings support a lifting. It was James Clerk Maxwell who observed in 1864 that the plane realization of a graph has to fulfill an equilibrium criterion to become liftable [58]. Assume that the edges of the plane graph correspond to springs. More precisely, we have springs that try to push the vertices away and springs that try to pull vertices together. The strength might differ from spring to spring, but the induced force is proportional to the spring length (Hook's law). If there exists a set of springs for the edges of the geometric graph, such that the forces meeting in every point cancel, we say the geometric graph supports



(a) (b)

Figure 1.1: Realization of a combinatorial graph in the plane such that a lifting exists (a) or not (b).

an equilibrium stress. The support of an equilibrium stress (not all stresses zero) is equivalent to the "lifability" of the plane embedding. Moreover, the computation of the lifting is easy if the configuration of springs is given.

Maxwell has not formally proved all of his observations. Nevertheless the principles were applied in mechanical engineering all along. From the mathematical point of view the ideas of Maxwell and others about liftings became forgotten. In the early 1980s Walter Whiteley and Henri Crapo started to rediscover Maxwell's work [30, 31, 91]. They also completed Maxwell's proofs. Other people (e.g. Hopcroft and Kahn [49], Richter-Gebert [72]) used the lifting method as powerful tool to solve and investigate geometric problems.

This thesis addresses geometric problems that are studied with help of the lifting method. We start our investigation by explaining how to compute a lifting. To obtain a clear formalism for the work with liftings we introduce concepts like stresses in graphs and reciprocal diagrams. This preliminary part is presented in Chapter 2. We recommend to (at least) browse through this chapter if the reader is not familiar with the lifting technique.

In the following chapters we study different geometric problems, which we group into two blocks. One block (Chapter 3 and 4) studies integer realizations of convex bodies in the 3-dimensional space. The other block (Chapter 5) describes how to characterize regular triangulations with help of linear programs over the polytope of pointed pseudo-triangulations (short PPT-polytope). In Chapter 6 we add some complexity theoretic observations, which are related to Chapter 5 but are not related to liftings. In the final Chapter 7 we give a short conclusion and discuss open problems. We continue with a more detailed introduction of the studied geometric problems.

1.1 Convex 3-Polytopes with Integer Coordinates

Due to Steinitz' seminal theorem [83] we know that any 3-connected planar graph is the graph of a 3-polytope. The graph describes the combinatorial structure of the polytope. Even though the original proof of Steinitz' theorem did not use liftings of planar graphs, the theorem can be proved with help of the lifting method discussed above (see Richter-Gebert [72]). The lifting method allows to compute the coordinates of the 3d realization easily. One can observe that all coordinates are rational. Therefore, integer coordinates suffice to embed

convex 3-polytopes.

What is known about the size of such an integral embedding? Richter-Gebert analyzed the common divisor of the rational coordinates and observed that a polytope with n vertices can be realized on a grid of size $O(2^{18n^2})$. For simplicial 3-polytopes Richter-Gebert was able to prove a better bound. The upper bound for these polytopes was only exponential in the size of n. Ribó [70] showed that 3-polytopes with at least one quadrilateral face can be embedded on an exponential sized grid as well. The question if an exponential bound holds in general has been still open. The unsolved case boils down to find an efficient embedding algorithm for polytopes whose smallest face is a pentagon (due to Euler's formula every planar 3-connected graph has to contain a face with at most five vertices).

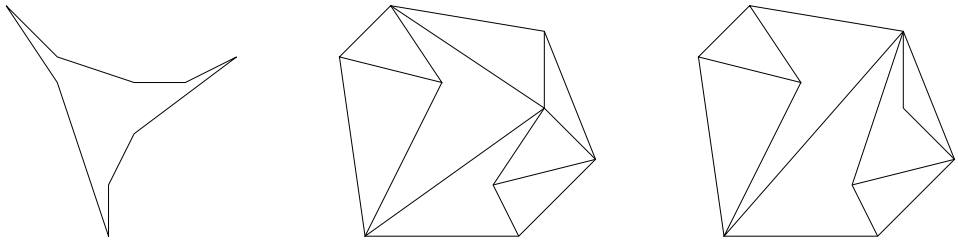
We solve this remaining open case in Chapter 3. The crucial part is to find a liftable plane embedding. Tutte's *spring embedding* method is a powerful tool to construct plane embeddings in equilibrium. Unfortunately the equilibrium is not "complete" for our purpose. The forces induced by the springs do not sum up to zero at the boundary. We show how to apply Tutte's method to obtain an embedding where all vertices are in equilibrium. For a careful analysis we need furthermore bounds on the number of spanning trees of a planar graph.

Related to the problem of embedding 3-polytopes on the grid is the realization of order types in $\mathbb{Z}^2$. An (abstract) order type of a set of n points specifies the orientation of any triple of points. Not all order types are realizable in $\mathbb{R}^2$ (e.g. the famous non-Pappus configuration), but any realizable order type can be realized with integer coordinates. As for the realization of 3-polytopes we have a combinatorial object (a realizable order type instead of a planar 3-connected graph) that supports a realization with integral coordinates. But in contrast to our results it is known that there exist order types that require integer coordinates doubly exponential in n [40].

1.2 Topics in Pseudo-Triangulations

In the second part of this thesis we address problems concerning pseudo-triangulation. A pseudo-triangle is a convex polygon with three internal angles smaller than π. We call the tiling of a domain into pseudo-triangles a *pseudo-triangulation* (see Figure 1.2). Of special interest are pseudo-triangulations where every vertex is incident to an angle greater than π – the so called *pointed* pseudo-triangulations. The most prominent variant of pseudo-triangulations are triangulations. Here every pseudo-triangle of the tiling is a triangle. Triangulations are a well studied object with uncountable applications. The younger and more general concept of pseudo-triangulations has many different applications, too. We list some of the most important applications and redirect the interested reader to the more comprehensive survey article by Rote, Santos and Streinu [76].

In 1993 Pocchiola and Vegter introduced pseudo-triangulations to investigate the visibility complex of convex objects in the plane [67, 68, 69]. Their work was inspired by pseudo-line arrangements [66]. Before 1993 special pseudo-



(a) (b) (c)

Figure 1.2: Example of a pseudo-triangle (a), a pseudo-triangulation (b) and a pointed pseudo-triangulation (c).

triangulations of simple (vertex empty) domains were studied to perform efficient ray shooting operations in simple polygons [23, 41]. These so called *geodesic* triangulations were built by tiling a polygon using non-crossing geodesic paths between its vertices. Another interesting field of applications for pseudo-triangulations are kinetic data structures [3, 15, 52]. Here pseudo-triangulations have been successfully used to quickly detect collisions in a set of moving objects. Also art gallery type problems can be solved with help of pseudo-triangulations. The task here is to guard a polygonal domain with guards whose "vision" is limited to an angle π. Speckmann and Tóth [82] showed that in this case for a simple polygon with n vertices (k of them convex) $\lfloor (2n-k)/3 \rfloor$ guards suffice to guard the polygon.

Pointed planar drawings on the sphere can be considered as spherical pseudo-triangulations. Gaiane Panina used these structures to disprove A.D. Alexandrov's uniqueness conjecture for convex bodies by constructing a saddle surface that is based on a lifted pseudo-triangulation [64]. A spherical pseudo-triangulation describes the normal fan of a virtual 3-polytope. If the spherical pseudo-triangulation is furthermore pointed the induced fan belongs to a hyperbolic virtual 3-polytope [65]. Thus spherical pseudo-triangulations can be used to visualize virtual 3-polytopes.

We conclude the list of applications with the popular *carpenter-rule problem*. The problem asks if there is an unfolding for any simple planar polygonal chain. A beautiful solution of this problem was given by Ileana Streinu [84]. It is based on the fact that a pointed pseudo-triangulation with one convex hull edge removed induces an expansive unfolding motion. A sequence of these motions can be assembled to a global unfolding of the polygonal linkage. Interestingly, a different solution for this problem by Connelly, Demaine and Rote [28] uses liftings of planar graphs. Another (energy-driven) solution for this fascinating problem is presented in [22].

The research on expansive mechanisms induced by pointed pseudo-triangulations led to the description of the *polytope of pointed pseudo-triangulations* by Rote, Santos, and Streinu [75]. The vertices of this high dimensional polytope represent the pointed pseudo-triangulations that can be realized on a given point set. Orden and Santos generalized this approach and found a polytope that represents all pseudo-triangulations of a point set (not only the pointed

ones) as its vertices [63].

Having a polytopal representation of all (pointed) pseudo-triangulations of a given point set is useful, because it allows to perform linear optimization on this set. We are interested in defining regular pseudo-triangulations as solutions of linear programs. This question is addressed in Chapter 5. We show how solutions of certain linear programs can be expressed as liftings of planar graphs.

The definition of a reasonable canonical pointed pseudo-triangulation for a point set, that fulfills a certain optimality criterion, turns out to be a challenging question. It is desirable to have a local geometric criterion that tells if a local modification brings us closer to the optimum or not. With help of this information one can construct a canonical pseudo-triangulation by performing a sequence of local modifications. The existence of a canonical optimality criterion makes the enumeration technique *reverse search* [13] applicable.

For non-pointed pseudo-triangulations minimal locally convex functions can be used to define such a optimal pseudo-triangulation (see Aurenhammer et al. [5, 6]). This approach is based on lifting planar graphs (in this case pseudo-triangulations) into $\mathbb{R}^3$. A convex polyhedral surface can be constructed by swapping all concave edges of the lifted surface. In the triangulation setting the famous (weighted) Delaunay triangulation acts as canonical optimal triangulation from this point of view. Unfortunately, the concept of minimal locally convex functions cannot be used for pointed pseudo-triangulations, because they only yield flat, degenerate convex liftings.

A promising idea to define a canonical pointed pseudo-triangulation was proposed in [75]. Here the solution of a canonical linear program was suggested as candidate for a canonical pointed pseudo-triangulation. We study this special pointed pseudo-triangulation in this thesis and give a geometric characterizations of it for convex point sets. This is achieved by transforming the canonical linear program into a problem of constructing a convex polyhedral surface. Again, we use the lifting technique to express the solution of the linear program as a polyhedral surface. This makes it possible to apply the framework developed by Aurenhammer et al. [5, 6].

The fact that the domain of the pseudo-triangulations is empty makes it possible to use liftings. In this case (convex) liftings are not degenerate. However, for the general case we were not able to come up with a solution using liftings.

Notice that Bereg [16] defined a canonical pointed pseudo-triangulation to study the flip distance of $O(n \log n)$ for pointed pseudo-triangulations. However, it is desirable to define a (different) canonical pointed pseudo-triangulation that is based on the PPT-polytope. This might lead to an enumeration scheme similar to the one of Bereg [17] and faster compared to the scheme of Brönnimann et al. [21].

The investigation of linear programs over the polytope of pointed pseudo-triangulations leads to new insights in pointed pseudo-triangulations. We were able to characterize pointed pseudo-triangulations as minimal rigid frameworks that can resolve external forces with positive interior stresses. Interestingly for any generic set of forces exists a unique pointed pseudo-triangulation that resolves it with positive stresses in the interior.

Most recently different generalizations of pseudo-triangulations have been studied. Maximal locally convex functions have been used to define an equivalent in higher dimension [12] and for domains with holes [8]. We consider the (classical) two-dimensional version of pseudo-triangulations in this thesis only.

In the last part of the thesis we study several complexity questions that arise in the context of pseudo-triangulations. In particular, we present several new NP-completeness results. We show that it is NP-hard to decide if a certain geometric graph contain a (pointed) pseudo-triangulation and that it is NP hard to decide if a geometric graph can be completed to a (pointed) pseudo-triangulation with bounded vertex degree. Our observations lead to a simplified proof of the NP-hardness of the *triangulation existence problem*, which asks if a given geometric graph contains a triangulation.

Chapter 2

Stresses and Liftings of Planar Graphs

The technique of lifting geometric graphs to $\mathbb{R}^3$ is used frequently in the thesis. For this reason we discuss stresses on planar graphs and their induced liftings in this preliminary chapter.

Throughout the thesis let $G = (V, E)$ be a connected graph with vertex set $V = \{v_1, \ldots, v_n\}$ and edge set $E \subseteq V \times V$. The number of vertices is denoted by n, the number of edges by m. We identify an edge by the indices of its vertices and write (i, j) instead of (v_i, v_j).

An embedding of a graph G is a realization of G in $\mathbb{R}^n$, where vertices are mapped to distinct points in the plane and edges are mapped to Jordan arcs connecting the points. A realization of G in the plane is called *plane* embedding. Unless otherwise specified an embedding of a graph means for the scope of this thesis a plane embedding. A graph G is called *planar* if it can be embedded into the plane without crossings of the edges. We call a crossing free embedding *planar embedding*. Furthermore, when all Jordan arcs are straight line segments the planar embedding is named *planar straight-line embedding*. Throughout the thesis we consider embeddings with straight lines only. For this reason we omit the term "straight-line" in the following when we speak about planar embeddings.

The order of the emanating edges for every vertex defines a *combinatorial embedding* of the graph G. Combinatorial embeddings that can be realized by a planar embedding are called *planar maps*. The cycles of edges induced by a combinatorial embedding define the set of faces $F_G = \{\ldots, f_i, \ldots\}$. A combinatorial embedding makes it possible to assign every directed edge (i, j) a unique face f_l, which lies left to it, and a unique face f_r, which lies right to it. We denote this situation by the ordered quadruple $(i, j|l, r)$ called *oriented patch*. Clearly, if $(i, j|l, r)$ is an oriented patch then $(j, i|r, l)$ is an oriented patch as well.

A graph is *3-connected* if one has to delete at least three vertices to split the graph into two disconnected components. Notice that every planar 3-connected graph has only two planar maps, where one planar map reverses the orientations

of the other planar map (see Whitney [92]). The faces defined by these two maps differ only in the orientation. The set of vertices that define a face is the same. In this thesis we consider mostly planar graphs that are also 3-connected. Since we are not interested in the orientation of the faces we can use any of the planar maps without restriction. In other words, the unoriented description of the faces depends in our setting only on the graph G.

We describe a straight-line embedding by an injective map $\mathbf{p} : V \to \mathbb{R}^2$. The map $\mathbf{p}$ assigns to every vertex v_i a two-dimensional vector $\mathbf{p}(v_i) = (x_i, y_i)$. We use the notation $\mathbf{p}_i$ as abbreviation for $\mathbf{p}(v_i)$. An embedding of G given by $\mathbf{p}$ is denoted as $G(\mathbf{p})$.

2.1 Equilibrium Stresses

Let $G = (V, E)$ be a graph with not necessarily planar embedding $\mathbf{p}$. A *stress* on G is an assignment ω of scalars to the edges E. The stress $\omega(e)$ of an edge $e = (i, j)$ is denoted as ω_{ij}. The stresses have to respect symmetry and therefore we have $\omega_{ij} = \omega_{ji}$. For convenience we set $\omega_{ij} = 0$ if (i, j) is not in E.

Definition 2.1. *Let $G = (V, E)$ be a graph with stress ω and let $G(\mathbf{p})$ be an embedding of G. We say:*

1. *A point $\mathbf{p}_i$ is in equilibrium, if*

$$\sum_{\mathbf{p}_j} \omega_{ij}(\boldsymbol{p}_i - \boldsymbol{p}_j) \;=\; \mathbf{0}. \tag{2.1}$$

2. *The embedding $G(\mathbf{p})$ is in equilibrium, if all of its vertices are in equilibrium.*

If at least one of the stresses ω_{ij} is nonzero we call the stress *non-trivial*. We say an embedding *supports* an equilibrium stress, if there exists some non-trivial stress for which $G(\mathbf{p})$ is in equilibrium.

Stresses on graphs are often interpreted as physical forces. We use this point of view frequently in this thesis. The force induced by an edge (i, j) at $\mathbf{p}_i$ equals the vector $\omega_{ij}(\mathbf{p}_i - \mathbf{p}_j)$. In equilibrium all forces sum up to zero – we say they *resolve*. If the forces do not sum up to zero (at a vertex) we call the forces *non-resolving*.

Notice that the existence of a non-trivial equilibrium stress is not a property of an (abstract) graph, but of its embedding in the plane. Figure 1.1 shows two embeddings of the same graph. The embedding depicted in Figure 1.1(a) supports an equilibrium stress, whereas the embedding in Figure 1.1(b) does not. In the following section we give criteria for the existence of non-trivial equilibrium stresses in embeddings of planar graphs.

2.1.1 Maxwell's Reciprocal Diagrams

A tool for a better understanding of embeddings of planar graphs with equilibrium stress is the concept of *reciprocal diagrams* introduced by James Clerk

Maxwell in 1864 [58]. A reciprocal diagram of an embedding $G(\mathbf{p})$ is the embedding of the dual graph of G. The definition of a dual graph makes only sense if G is planar. Therefore, reciprocal diagrams are only well defined for planar graphs. The embedding, however, does not have to be necessarily planar.

Let G^* denote the dual graph of G. Throughout the thesis we consider dual graphs of 3-connected planar graphs only. In this case the dual graph is unique. The vertex set of the reciprocal diagram is given by the set of faces F_G. The edges of the reciprocal correspond to adjacent faces of G.

Definition 2.2. *An embedding $G^*(\mathbf{r})$ is called* reciprocal diagram *of $G(\mathbf{p})$ if for every oriented patch $(i, j|l, r)$ of G the vector $\mathbf{p}_i - \mathbf{p}_j$ is orthogonal to the vector $\mathbf{r}_l - \mathbf{r}_r$.*

We have not forbidden that two vertices in the reciprocal diagram have the same coordinates. However, such a setting is degenerated in the same sense that a stress has the value zero. For this reason we call a reciprocal diagram where every vertex has a distinct position *non-trivial*.

We can identify all embeddings of a planar graph G that support an equilibrium stress by the following theorem.

Theorem 2.3 (Maxwell). *An embedding $G(\mathbf{p})$ of a planar graph G supports a non-trivial equilibrium stress, if and only if a non-trivial reciprocal diagram exists.*

Proof. (sketch): Assume $G(\mathbf{p})$ supports an equilibrium stress ω. As mentioned before the vector $\omega_{ij}(\mathbf{p}_i - \mathbf{p}_j)$ can be seen as a force that affects $\mathbf{p}_i$ and $\mathbf{p}_j$. In equilibrium the forces meeting in every vertex sum up to zero. An example is shown in Figure 2.1. Since the forces in every point sum up to zero, one can rearrange the vectors of the forces such that they form a (face) cycle – a so called *polygon of forces*[1]. See Figure 2.2(a) for an illustration of the polygons of forces produced by the example shown in Figure 2.1. We observe that the

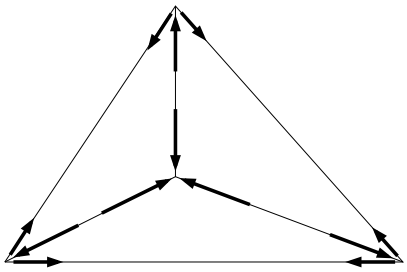
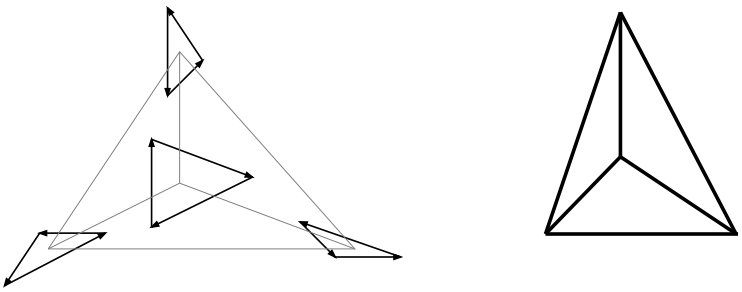


Figure 2.1: An example of an embedding with forces induced by the equilibrium stress.

force induced by an edge (i, j) at $\mathbf{p}_i$ is opposite to the force induced by the same edge at $\mathbf{p}_j$. This implies that polygons of forces can be assembled to an

[1]The fact that the resolving forces form a polygon is a well known fact and was remarked long before Maxwell [47, page 334].

embedding $G^*(\mathbf{r})$ (see Figure 2.2(b)). The edges of such an embedding $G^*(\mathbf{r})$ are parallel to the edges of $G(\mathbf{p})$. Rotating by $\pi/2$ yields a Maxwell reciprocal.



(a) (b)

Figure 2.2: Constructing the Maxwell reciprocal.

On the other hand, if there exists a Maxwell reciprocal we can construct resolving forces for $G(\mathbf{p})$ by reversing the construction from above. This defines an equilibrium stress. □

A supported stress of $G(\mathbf{p})$ can be directly read off Maxwell's reciprocal. Let $G^*(\mathbf{r})$ be a reciprocal diagram of $G(\mathbf{p})$ and $(i, j|l, r)$ an oriented patch of G. As a consequence of the proof of Theorem 2.3 the stress ω_{ij} can be expressed as

$$\omega_{ij} = \frac{y_l - y_r}{x_i - x_j} = \frac{x_r - x_l}{y_i - y_j}.$$

If $x_i = x_j$ or $y_i = y_j$ one of the above fractions is undefined but the other can still be used to express ω_{ij}. We notice that $G(\mathbf{p})$ is a reciprocal diagram of every reciprocal diagram of $G(\mathbf{p})$. Therefore, any reciprocal is in equilibrium too. An equilibrium stress ω' in the reciprocal diagram can be obtained by

$$\omega'_{rl} := 1/\omega_{ij},$$

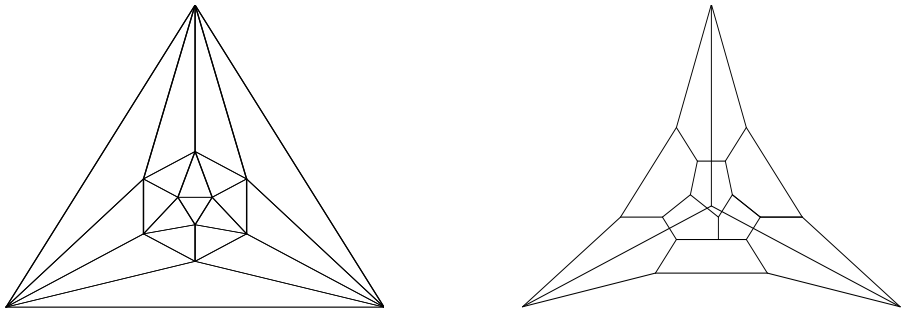
if $(i, j|r, l)$ is an oriented patch of G.

Other criteria for the existence of equilibrium stresses for special types of graphs can be found in [91]. We refer to [31] for a survey on Maxwell reciprocals.

Notice that a reciprocal diagram of a non-crossing embedding does not have to be necessarily crossing free. Moreover, embeddings with convex faces might result in a non-convex boundary of the reciprocal. An example is shown in Figure 2.3. The dual of the icosahedron is the dodecahedron. One can observe that the boundary vertices produce non-convex faces. Furthermore, the embedding is not crossing free. Crossing free embeddings that have a crossing free reciprocal can be characterized by the sign pattern of the stress and the shape of the faces [62].

2.1.2 Stresses on K_4

Any embedding of the complete graph with four vertices K_4 supports a non-trivial equilibrium stress. The equilibrium stress is unique up to scaling with a



(a) (b)

Figure 2.3: An embedding of the graph of the icosahedron and a reciprocal diagram with crossings of it.

constant. We show how to compute this stress in terms of the coordinates of the embedding $K_4(\mathbf{p})$[2].

Let $[i, j, k]$ denote the signed area of the triangle spanned by $\mathbf{p}_i, \mathbf{p}_j$ and $\mathbf{p}_k$. If the three points are in clockwise order the area will be negative, otherwise positive. We can compute $[i, j, k]$ as

$$\frac{1}{2} \det \begin{pmatrix} x_i & x_j & x_k \\ y_i & y_j & y_k \\ 1 & 1 & 1 \end{pmatrix}.$$

Proposition 2.4. *Let $K_4(\mathbf{p})$ be a plane embedding of the K_4 with no three points on a line. We define the stress for any edge $(i, j) \in E$ as*

$$\omega_{ij} := \frac{1}{[i, j, k][i, j, l]},$$

where $\mathbf{p}_k, \mathbf{p}_l$ are the remaining two vertices. This stress is an equilibrium stress on $K_4(\mathbf{p})$.

Proof. By symmetry it suffices to show that $\mathbf{p}_1$ is in equilibrium. This is true if

$$\frac{\mathbf{p}_1 - \mathbf{p}_2}{[1, 2, 3][1, 2, 4]} + \frac{\mathbf{p}_1 - \mathbf{p}_3}{[1, 3, 2][1, 3, 4]} + \frac{\mathbf{p}_1 - \mathbf{p}_4}{[1, 4, 2][1, 4, 3]} = \mathbf{0}.$$

We multiply with $[1, 2, 3][1, 2, 4][1, 3, 4]$ and obtain as equivalent statement

$$(\mathbf{p}_1 - \mathbf{p}_2)[1, 3, 4] + (\mathbf{p}_3 - \mathbf{p}_1)[1, 2, 4] + (\mathbf{p}_1 - \mathbf{p}_4)[1, 2, 3] = \mathbf{0}.$$

We notice that $[1, 3, 4] - [1, 2, 4] + [1, 2, 3]$ equals the signed area $[2, 3, 4]$. Thus we can rephrase the last expression as

$$\mathbf{p}_1[2, 3, 4] - \mathbf{p}_2[1, 3, 4] + \mathbf{p}_3[1, 2, 4] - \mathbf{p}_4[1, 2, 3] = \mathbf{0}.$$

[2]The equilibrium stress on the K_4 (in the form we introduced it) is also mentioned in [75]. The proof presented there is different from ours.

Keep in mind that the last line includes two conditions, one for the x and one for the y-coordinates of $K_4(\mathbf{p})$. We can express each of in form of a determinant, namely

$$\det \begin{pmatrix} x_1 & x_2 & x_3 & x_4 \\ x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ 1 & 1 & 1 & 1 \end{pmatrix} = 0, \quad \det \begin{pmatrix} y_1 & y_2 & y_3 & y_4 \\ x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ 1 & 1 & 1 & 1 \end{pmatrix} = 0.$$

The expansion by the first rows yields that both determinants are zero. □

2.2 Lifting of Equilibrium Stressed Graphs

The technique of lifting equilibrium stressed embeddings is applied in several constructions in this thesis. This section explains the basic procedure of the lifting process. The construction of liftings with equilibrium stresses goes back to Crapo and Whiteley [30, 91] and was also addressed by Hopcroft and Kahn [49] and Richter-Gebert [72].

Maxwell observed that there is a correspondence between embeddings with equilibrium stress and projections of 3d polyhedra. We call an embedding of a planar 3-connected graph to $\mathbb{R}^3$ *spatial* if the vertices of every face of the graph lie on a common plane. There exist different versions of Maxwell's theorem. For the scope of the thesis the following formulation is the most suitable.

Theorem 2.5 (Maxwell, Whiteley). *Let G be a planar 3-connected graph with embedding $G(\mathbf{p})$ and designated face f_1. There exists a correspondence between*

A.) equilibrium stresses ω on $G(\mathbf{p})$,

B.) spatial embeddings of G in $\mathbb{R}^3$, where face f_1 lies in the xy-plane and the orthogonal projection of the spatial embedding in the xy-plane along the z-axis gives $G(\mathbf{p})$.

The proof that A induces B is due to Walter Whiteley [91]. For this reason we call the Theorem 2.5 Maxwell-Whiteley Theorem. The Italian mathematician Luigi Cremona set Maxwell's work in a clearer form [32]. This is why Theorem 2.5 is also known as the *Maxwell-Cremona* correspondence. Cremona also introduced slightly different reciprocal diagrams. The difference is that in the Maxwell reciprocals corresponding edges are perpendicular, whereas in the Cremona reciprocals they are parallel. We refer to [90] for a survey of Cremona's work.

There exists stronger versions Theorem 2.5, which include also a connection between the sign of the ω_{ij} stresses and curvature of the lifted edge (i, j). This important condition is studied by us separately later in this section (see Observation 2.8).

In the thesis we apply the direction of Theorem 2.5 proved by Whiteley frequently. We therefore give a brief constructive proof of this direction. In

particular, we show that every plane embedding of a planar 3-connected graph with an equilibrium stress can be lifted to a 3d-polyhedron. Furthermore we show how to compute this polyhedron.

Other proofs of the reverse direction of the Maxwell-Whiteley Theorem were obtained by Hopcroft and Kahn [49] and Richter-Gebert [72]. The following proof is a modification of Richter-Gebert's proof. It emphasizes the connection between reciprocal diagrams and the lifting.

The lifting of an embedded graph $G(\mathbf{p})$ is characterized by a set of planes in $\mathbb{R}^3$. Every face $f_i \in F_G$ is associated with a plane H_i. The lifted face f_i is the orthogonal projection onto H_i along the z-axis. We express a plane H_i by the equation

$$H_i : \ \mathbf{p} \mapsto z_i(\mathbf{p}) = \langle \mathbf{p}, \mathbf{a}_i \rangle + d_i.$$

We use the reciprocal diagrams to define the gradients $\mathbf{a}_i$ of the planes H_i. Notice that even for a fixed equilibrium stress there is no unique reciprocal diagram. Any translation and scaling of a reciprocal diagram produces another valid reciprocal diagram. To obtain a unique definition of the lifted planes for a fixed equilibrium stress we fix one reciprocal diagram as *the* reciprocal diagram $G^*(\mathbf{r})$ for a given equilibrium stress ω. The rotation of a vector $\mathbf{p} = (x, y)^T$ by $\pi/2$ is denoted by $\mathbf{p}^\perp := (-y, x)^T$. We set

$$\mathbf{r}_1 := (0, 0)^T$$

and for any oriented patch $(i, j|l, r)$ we assume

$$\mathbf{r}_l - \mathbf{r}_r = \omega_{ij}(\mathbf{p}_i - \mathbf{p}_j)^\perp. \tag{2.2}$$

Notice that the condition (2.2) holds for every edge in the reciprocal diagram constructed in the proof for Theorem 2.3. Thus there exists a reciprocal diagram $G^*(\mathbf{r})$ that fulfills these conditions.

Let the planes H_i be defined by the following equations:

$$\begin{aligned} \mathbf{a}_k &:= \mathbf{r}_k, \\ d_1 &:= 0, \\ d_l &:= \omega_{ij}\langle \mathbf{p}_i, \mathbf{p}_j^\perp \rangle + d_r \quad (i, j|l, r) \text{ is oriented patch of } G. \end{aligned} \tag{2.3}$$

Lemma 2.6. *The set of parameters $\mathbf{a}_i$ and d_i is well defined.*

Proof. The gradients $\mathbf{a}_i$ are uniquely defined since we fixed a reciprocal diagram $G^*(\mathbf{r})$. The inductive definition of the scalars d_i needs deeper investigation.

The value of d_i can be computed by selecting a sequence of face transitions. Let $F_k := (r, l)$ be a face transition in G from face f_r to face f_l. A face transition (r, l) is valid if there exists an oriented patch $(i, j|l, r)$ for some edge (i, j). For a face transition F_k, defined by the oriented patch $(i, j|l, r)$, we denote the associated oriented edge (i, j) by t_k.

Let $F_1, F_2, \ldots F_u$ be sequence of face transition with $F_1 = (1, \cdot)$ and $F_u = (\cdot, i)$. If for any $k < u$ holds that $F_k = (\cdot, \theta)$ induces $F_{k+1} = (\theta, \cdot)$ the sequence can be used to calculate d_i. In general such a sequence is not unique.

Therefore, we have to show that, no matter which sequence one chooses for the computation of d_i, the obtained value for d_i is always the same.

Assume we have two sequences of face transitions starting from f_1 and ending at f_i. We reverse one of the sequences and link both together. This leads to a sequence of faces that induce a cycle in the dual graph. Let this cycle be

$$\mathcal{C} := F_1, \dots F_K.$$

If two face transition sequences lead to different values for d_i then the sum $\sum_{t_k=(i,j):k\leq K} \omega_{ij}\langle \mathbf{p}_i, \mathbf{p}_j^\perp\rangle$ would be nonzero. Therefore, it is sufficient to show that for any face cycle $\mathcal{C}$

$$\sum_{t_k=(i,j):k\leq K} \omega_{ij}\langle \mathbf{p}_i, \mathbf{p}_j^\perp\rangle = 0 \tag{2.4}$$

holds.

The cycle $\mathcal{C}$ separates the graph into two components which we call I and O. First we assume that I contains only one vertex with name $\mathbf{p}_1$ (this situation is depicted in Figure 2.4(a)). We can deduce

$$\begin{aligned}
\sum_{t_k=(i,j):k\leq K} \omega_{ij}\langle \mathbf{p}_i, \mathbf{p}_j^\perp\rangle &= \sum_{(1,j)\in E} \omega_{1j}\langle \mathbf{p}_1, \mathbf{p}_j^\perp\rangle \\
&= \left\langle \mathbf{p}_1, \sum_{(1,j)\in E} \omega_{1j}\mathbf{p}_j^\perp \right\rangle \\
&= -\left(\sum_{(1,j)\in E} \omega_{1j}\right) \langle \mathbf{p}_1, \mathbf{p}_1^\perp\rangle \\
&= 0.
\end{aligned}$$

Thus condition (2.4) holds.

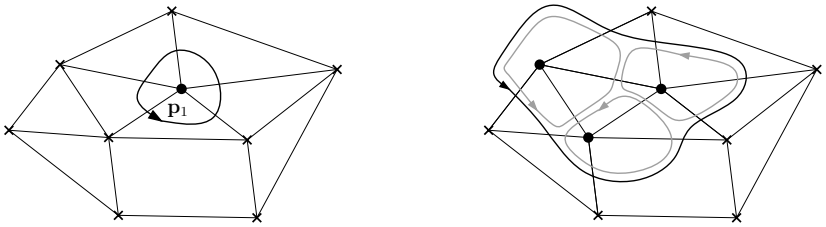
In general I contains more than one vertex. In this case we decompose $\mathcal{C}$ into smaller circles $\mathcal{C}_i$ with $i \in I$. A circle $\mathcal{C}_i$ traverses around $\mathbf{p}_i$ in the same orientation as $\mathcal{C}$. We notice that $\omega_{ij}\langle \mathbf{p}_i, \mathbf{p}_j^\perp\rangle + \omega_{ij}\langle \mathbf{p}_j, \mathbf{p}_i^\perp\rangle = 0$. Thus the edges with both end points in I "cancel" when summing over all small circles. In particular we have

$$\sum_{t_k=(i,j):k\leq K} \omega_{ij}\langle \mathbf{p}_i, \mathbf{p}_j^\perp\rangle = \sum_{i\in I}\sum_{(i,j)\in E} \omega_{ij}\langle \mathbf{p}_i, \mathbf{p}_j^\perp\rangle.$$

See Figure 2.4(b) for an example. Since we know that for any fixed i the sum $\sum_{(i,j)\in E} \omega_{ij}\langle \mathbf{p}_i, \mathbf{p}_j^\perp\rangle$ equals zero, the sum induced by the larger cycle is also zero, which proves the lemma. □

After showing that the definition (2.3) gives a set of well defined planes H_i we still have to prove that the lifting yields a consistent spatial embedding.

Theorem 2.7. *For an embedding $G(\mathbf{p})$ with equilibrium stress ω the lifting specified in* (2.3) *gives a lifting to a polyhedron.*



(a) (b)

Figure 2.4: The sets I (dots) and O (crosses) for two situations discussed in the proof of Lemma 2.6. The arrow indicates the cycle $\mathcal{C}$.

Proof. We have to show that for any oriented patch $(i, j|l, r)$ of G the lifted edge (i, j) is contained in the planes H_l and H_r. It is sufficient to prove this for the points $\mathbf{p}_i$ and $\mathbf{p}_j$.

We know that $d_l - d_r = \omega_{ij}\langle \mathbf{p}_i, \mathbf{p}_j^\perp \rangle$ due to equation (2.3). The rules given in (2.3) induce furthermore that the difference $\mathbf{a}_l - \mathbf{a}_r$ equals the vector $\mathbf{r}_l - \mathbf{r}_r$ in the reciprocal diagram $G^*(\mathbf{r})$. Due to condition (2.2) this vector equals $\omega_{ij}(\mathbf{p}_i - \mathbf{p}_j)^\perp$. We deduce

$$\begin{aligned} z_l(\mathbf{p}_i) - z_r(\mathbf{p}_i) &= \langle \mathbf{a}_l, \mathbf{p}_i \rangle - \langle \mathbf{a}_r, \mathbf{p}_i \rangle + d_l - d_r \\ &= \langle \mathbf{a}_l - \mathbf{a}_r, \mathbf{p}_i \rangle + (d_l - d_r) \\ &= \langle \omega_{ij}(\mathbf{p}_i - \mathbf{p}_j)^\perp, \mathbf{p}_i \rangle + \omega_{ij}\langle \mathbf{p}_i, \mathbf{p}_j^\perp \rangle \\ &= 0 \end{aligned}$$

Evaluating $z_l(\mathbf{p}_j) - z_r(\mathbf{p}_j)$ leads to the same result. □

Throughout the thesis we construct liftings incrementally. As a first step we fix the plane H_1 for a designated face f_1 by determining $\mathbf{a}_1$ and d_1. The iterative step is the following: We select a face f_l that is incident to an already lifted face f_r. Let $(i, j|l, r)$ be an oriented patch of G. As a result of (2.3) we

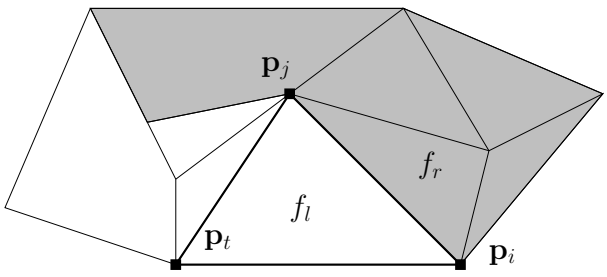


Figure 2.5: Lifting face by face. The shaded area has already been lifted – the next face to lift is f_l.

can calculate

$$\mathbf{a}_l = \omega_{ij}(\mathbf{p}_i - \mathbf{p}_j)^\perp + \mathbf{a}_r, \tag{2.5}$$

$$d_l = \omega_{ij}\langle \mathbf{p}_i, \mathbf{p}_j^\perp \rangle + d_r. \tag{2.6}$$

In this way all planes can be computed incrementally. We end with the complete lifting of $G(\mathbf{p})$.

The signs of the equilibrium stress contain information about the curvature of the edges in the spatial embedding. Let (i, j) be an edge in the lifting. Assume that we have the situation depicted in Figure 2.5. For simplicity we set $\mathbf{p}_i = \mathbf{a}_r = (0, 0)^T$. This assumption implies that $d_l = d_r = 0$. The height of the lifted point $\mathbf{p}_t$ on the plane H_l can be expressed as

$$\begin{aligned} z_l(\mathbf{p}_t) &= \langle \omega_{ij}(\mathbf{p}_i - \mathbf{p}_j)^{\perp}, \mathbf{p}_t \rangle, \\ &= \omega_{ij} \left\langle \begin{pmatrix} y_j \\ -x_j \end{pmatrix}, \begin{pmatrix} x_t \\ y_t \end{pmatrix} \right\rangle, \\ &= 2\, \omega_{ij}[t, j, i]. \end{aligned}$$

The sign of $[t, j, i]$ is negative. Thus a positive stress ω_{ij} produces a negative z-coordinate for $\mathbf{p}_t$. Since f_r is located in the xy-plane we can deduce that f_l lies "below" this plane. We identify all edges with $\omega_{ij} > 0$ as *mountains* and all edges with $\omega_{ij} < 0$ as *valleys*. See Figure 2.6 for an illustration. Knowing

mountain valley

Figure 2.6: The dependence of the sign of ω_{ij} and the curvature of the lifting.

the sign pattern for the stresses on all edges allow us to predict the curvature.

Observation 2.8. *Let $G(\mathbf{p})$ be an embedding of a planar 3-connected graph G with equilibrium stress. If the stresses on the boundary edges are negative and all other stresses positive then the spatial lifting induced by the equilibrium stress results in a convex 3-polytope.*

Stresses and equilibrium stresses in geometric graphs are used intensively in rigidity theory. We leave this important part out of the chapter because it is not relevant for the scope of the thesis. We recommend the survey article of Connelly [27] for the reader interested in this topic.

Chapter 3

An Upper Bound for Realizing 3-Polytopes on the Integer Grid

3.1 Overview

Due to Steinitz' famous theorem [83] we know that the edge graphs of convex 3-polytopes are 3-connected and planar. Conversely, any 3-connected planar graph is the edge graph of some 3-polytope. Hence, 3-connected planar graphs can be seen as combinatorial description of 3-polytopes.

Assume we have given the combinatorial description of a 3-polytope by its graph $G = (V, E)$. Let n be the number of vertices of G. It is known that G can be realized as 3-polytope with integer coordinates only. Furthermore, Richter-Gebert showed that coordinates not larger than $O(2^{18n^2})$ suffice for an integer embedding [72].

In the following we improve the bound of Richter-Gebert and show that G can be realized as 3-polytope with integer coordinates not greater than $O(2^{7.55n})$. This implies that for any 3-polytope a combinatorially equivalent polytope can be stored with $O(n)$ bits per vertex.

Integer realizations with at most exponentially large coordinates in terms of n were previously known for polytopes whose graph contains a triangle [72]. For stacked polytopes a better upper bound exists [95], but it is still exponential. Recently Ribó showed that also polytopes with at least one quadrilateral face can be embedded with exponentially large integer coordinates [70].

Embedding a planar 3-connected graph in the plane can be done very efficiently. For a convex free embedding of a planar graph an $O(n) \times O(n)$ grid is sufficient [33, 79]. This is also true if the embedding has to be convex [19]. A strictly convex drawing can be realized on an $O(n^2) \times O(n^2)$ grid [14]. In higher dimension it is known that there are 4-polytopes that cannot be realized with rational coordinates. Moreover a 4-polytope that can be realized on the grid might require coordinates that are doubly exponential in the number of its vertices [72, 73].

Road map for an algorithm. In the following we present an embedding algorithm to realize G as integral 3-polytope. The algorithm works in three stages. In the first stage we construct a two-dimensional non-crossing embedding of G. The 2d embedding is chosen in such a way that it supports an equilibrium stress. In the second stage of the algorithm we lift the plane embedding to $\mathbb{R}^3$. This is based on the observations made in Chapter 2. Finally, in the last stage, we compute appropriate scaling factors and scale the embedding such that it uses integer coordinates only. A careful analysis of the whole algorithm gives the improved upper bound for the integer coordinates.

3.2 Constructing a Plane Embedding

The objective of the first stage of the algorithm is to embed G in the xy-plane. The embedding should be non-crossing and it should support an equilibrium stress. An embedding that supports no equilibrium stress is useless for any approach that follows the road map of our algorithm. As a consequence of the Maxwell-Whiteley Theorem such a plane embedding cannot be lifted to $\mathbb{R}^3$.

We use the *spring embedding* method of Tutte [86, 87] to construct a non-crossing convex embedding that supports an equilibrium stress. We first describe the spring embedding method as black box. The spring embedding needs four ingredients: the graph G, a specified outer face f_0, the coordinates for the vertices of f_0, and finally a positive stress ω for any edge that is not on the outer face. It only makes sense to place the outer face as convex polygon. The coordinates of f_0 have to respect this fact. Tutte's method processes these inputs and produce the coordinates for all interior vertices. The embedding is non-crossing convex and every interior point is in equilibrium with respect to the stress ω. Therefore, the spring embedding seems to be the right method to embed G in the plane. Notice that the equilibrium stress is only partial. The vertices of the face f_0 are not in equilibrium. The good news is that we have not specified the stresses on the edges of f_0 yet. Our hope is to find appropriate stresses for these edges such that all vertices are in equilibrium. Unfortunately, this is not always possible. It depends on the location we gave the boundary vertices. Only certain positions for the vertices of f_0 result in an embedding that can be extended to an equilibrium embedding.

To obtain a strategy that gives a complete equilibrium embedding we take a closer (and more formal) look at the spring embedding. Let k be the number of vertices in f_0. For simplicity we want k as small as possible. Euler's formula implies that k is at most 5. We assume that the vertices in G are labeled in such a way that the first k vertices belong to f_0 in cyclic order. Let $B := \{1, \ldots, k\}$ be the index set of the boundary points and let $I := \{k+1, \ldots, n\}$ denote the index set of the interior points. The edges of f_0 are called *boundary edges*, all other edges *interior edges*. Tutte's spring embedding works with any positive stress for the interior edges. For simplicity we set all stresses on interior edges to 1.

Let A be the adjacency matrix of G without the boundary edges and let D be the diagonal matrix of row sums of A. We subdivide A and D into block

matrices indexed by the vertex sets I and B. The matrix $D_I - A_{II}$ is called the *reduced Laplacian matrix* $\bar{L}$ of G.

The location of the boundary face is given by the vectors $\mathbf{x}_B = (x_1, \ldots, x_k)^T$ and $\mathbf{y}_B = (y_1, \ldots, y_k)^T$. As a consequence of the spring embedding the position of the of the interior points $\mathbf{x}_I = (x_{k+1}, \ldots, x_n)^T$ and $\mathbf{y}_I = (y_{k+1}, \ldots, y_n)^T$ can be computed with help of the equilibrium condition (2.1). In particular,

$$\begin{aligned} \mathbf{x}_I &= \bar{L}^{-1} A_{IB} \mathbf{x}_B, \\ \mathbf{y}_I &= \bar{L}^{-1} A_{IB} \mathbf{y}_B. \end{aligned} \tag{3.1}$$

So far we can guarantee equilibrium for the inner vertices:

$$\forall i \in I \quad \sum_{\mathbf{p}_j} \omega_{ij}(\mathbf{p}_i - \mathbf{p}_j) \;=\; \mathbf{0}. \tag{3.2}$$

For the boundary vertices we define the vectors $F := \{\mathbf{F}_1, \ldots, \mathbf{F}_k\}$ as the non-resolving forces which arise at the boundary vertices and cannot be canceled by the interior stresses:

$$\forall i \in B \quad \sum_{\mathbf{p}_j} \omega_{ij}(\mathbf{p}_i - \mathbf{p}_j) \;=: \mathbf{F}_i. \tag{3.3}$$

The forces F depend on the location of the boundary points and the combinatorial structure of G. Notice that changing the coordinates of the boundary face changes the forces F. The following lemma helps to express this dependence.

Lemma 3.1 (Substitution Lemma). *There are nonnegative weights $\tilde{\omega}_{ij} = \tilde{\omega}_{ji}$, for $i, j \in B$, independent of $\mathbf{p}$, such that the resulting forces at the boundary vertices $i \in B$ are obtained by*

$$\mathbf{F}_i = \sum_{j \in B : j \neq i} \tilde{\omega}_{ij}(\mathbf{p}_i - \mathbf{p}_j). \tag{3.4}$$

The weights $\tilde{\omega}$ are multiples of $1/\det \bar{L}$.

Proof. Let $\mathbf{F}_x$ denote the vector $(F_1^x, \ldots, F_k^x)^T$, where F_i^x is the x-coordinate of the force $\mathbf{F}_i$. Let A and D be defined as in the definition for the reduced Laplacian matrix. We rephrase (3.3) as $\mathbf{F}_x = D_B \mathbf{x}_B - A_{BI} \mathbf{x}_I$. With help of (3.1) we substitute $\mathbf{x}_I$ and obtain

$$\begin{aligned} \mathbf{F}_x &= D_B \mathbf{x}_B - A_{BI} \bar{L}^{-1} A_{IB} \mathbf{x}_B \\ &= D_B \mathbf{x}_B - A_{BI} (D_I - A_{II})^{-1} A_{IB} \mathbf{x}_B \\ &=: \tilde{A} \mathbf{x}_B. \end{aligned}$$

For the y-coordinates, we obtain a similar formula with the same matrix $\tilde{A}$. We define $\tilde{\omega}_{ij}$ as the entries $\tilde{a}_{ij}$ of $\tilde{A}$. Since $A_{BI} = (A_{IB})^T$, the matrix $\tilde{A}$ is symmetric and therefore $\tilde{\omega}_{ij} = \tilde{\omega}_{ji}$ holds.

To show that the expression $\mathbf{F}_x = \tilde{A} \mathbf{x}_B$ has the form stated in (3.4) we have to check that all row sums in $\tilde{A}$ equal 0. Let $\mathbf{1}$ denote the vector where all entries

are 1. We know that $A_{II}\mathbf{1}+A_{IB}\mathbf{1} = D_I\mathbf{1}$ and therefore $(D_I-A_{II})^{-1}A_{IB}\mathbf{1} = \mathbf{1}$. Plugging this expression into $\tilde{A}\mathbf{1} = D_B\mathbf{1} - A_{BI}(D_I - A_{II})^{-1}A_{IB}\mathbf{1}$ gives us $\tilde{A}\mathbf{1} = D_B\mathbf{1} - A_{BI}\mathbf{1}$, which equals $(0,\ldots,0)^T$.

The matrix $\tilde{A}$ can be written as a rational expression whose denominator is the determinant of $D_I - A_{II} = \bar{L}$, and thus the weights $\tilde{\omega}$ are multiples of $1/\det\bar{L}$. □

Notice that the $\tilde{\omega}$ values are independent of the location of $\mathbf{x}_B$ and $\mathbf{y}_B$, they only depend on the combinatorial structure of G. In other words, the stresses $\tilde{\omega}_{ij}$ store all the necessary information about the combinatorial structure of G we need. Thus, we have a compact (constant) description of the structure of G that is responsible for the forces F. We name the stresses $\tilde{\omega}$ *substitution stresses* to emphasize that they are used as a substitution for the combinatorial structure of G.

For the later analysis of the grid size it is necessary to bound the substitution stresses.

Lemma 3.2. *Any substitution stress $\tilde{\omega}_{ij}$ is bounded by*

$$0 \leq \tilde{\omega}_{ij} < n - k.$$

Proof. The substitution stresses are independent on the location of f_0. Therefore, we can choose the positions for the boundary points freely. We place vertex $\mathbf{p}_i$ at position $(0,0)^T$. Let the x-coordinate of all other boundary vertices be 1. The stress $\tilde{\omega}_{ij}$ is the x-part of $\mathbf{F}_j$ and therefore by (3.3) we have $\tilde{\omega}_{ij} = \sum_{k\in I}\omega_{jk}(x_j - x_k)$. This sum consists of $|I|$ positive summands all less than 1 (remember that all ω's are 1). Thus we have $0 \leq \tilde{\omega}_{ij} < n - k$. □

Next we decide how to locate the boundary face such that the forces F can be canceled out by some stress on the boundary edges. The boundary stresses ω can be expressed as a linear equation system ($\mathbf{F}_i = 0, i = 1,\ldots,k$). This system is over constrained and therefore we regard it as a system where the coordinates $\mathbf{x}_B$ and $\mathbf{y}_B$ appear as variables too. The extended non-linear system consists of the $2k$ equations given in (3.4) plus the following $2k$ equations

$$\forall i \in B : \omega_{i,suc(i)}(\mathbf{p}_i - \mathbf{p}_{suc(i)}) + \omega_{i,pre(i)}(\mathbf{p}_i - \mathbf{p}_{pre(i)}) = -\mathbf{F}_i, \qquad (3.5)$$

where $suc(i)$ denotes the successor of v_i and $pre(i)$ denotes the predecessor of v_i at f_0 in cyclic order. The system is now under constrained. To solve it, we fix as many boundary coordinates as necessary to obtain a unique solution. We continue with a case distinction on k to guarantee the convexity of the outer face.

Case 1: G contains a triangular face

The triangular case is easy: we can position the boundary vertices at any convenient position. We choose:

$$\mathbf{p}_1 = \begin{pmatrix}0\\0\end{pmatrix}, \mathbf{p}_2 = \begin{pmatrix}1\\0\end{pmatrix}, \mathbf{p}_3 = \begin{pmatrix}0\\1\end{pmatrix}. \qquad (3.6)$$

Lemma 3.3. *If the smallest face of G is a triangle and we place the boundary vertices as stated in* (3.6) *then the boundary forces can be resolved.*

Proof. We embed G as spring embedding and calculate the substitution stresses. After setting $\omega_{12} = -\tilde{\omega}_{12}, \omega_{23} = -\tilde{\omega}_{23}, \omega_{13} = -\tilde{\omega}_{13}$ all points are in equilibrium. □

Case 2: G contains a quadrilateral but no triangular face

If f_0 is a quadrilateral we have to fix some coordinates of the boundary vertices such that it is possible to cancel the forces F. A unique solution can be obtained by setting

$$\mathbf{p}_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \mathbf{p}_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \mathbf{p}_3 = \begin{pmatrix} 2 \\ y_3 \end{pmatrix}, \mathbf{p}_4 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \tag{3.7}$$

Under this assumption, we obtain

$$\begin{aligned} \omega_{12} &= -2\tilde{\omega}_{13} - \tilde{\omega}_{12}, \\ \omega_{23} &= \tilde{\omega}_{24} - 2\tilde{\omega}_{13} - \tilde{\omega}_{23}, \\ \omega_{34} &= -\frac{\tilde{\omega}_{24}}{2} - \tilde{\omega}_{34}, \\ \omega_{14} &= \frac{\tilde{\omega}_{24}\tilde{\omega}_{13}}{\tilde{\omega}_{24} - 2\tilde{\omega}_{13}} - \tilde{\omega}_{14}, \\ y_3 &= \frac{\tilde{\omega}_{24}}{2\tilde{\omega}_{13} - \tilde{\omega}_{24}}. \end{aligned} \tag{3.8}$$

We assume that $\tilde{\omega}_{13} \geq \tilde{\omega}_{24}$. (Otherwise we cyclically relabel the vertices of f_0.) Thus $y_3 > 0$ and f_0 forms a *convex* face.

Notice that the substitution stresses $\tilde{\omega}_{ij}$ between adjacent vertices (on the boundary) are irrelevant. The forces resulting by the boundary stresses $\tilde{\omega}_{ij}$ can be directly canceled by the corresponding stresses ω_{ij}. This can also be observed by looking at the solution (3.8) of the corresponding equation system: Boundary stresses do not appear in the solution for y_3 and furthermore the sum $\tilde{\omega}_{ij} + \omega_{ij}$ for boundary edges (i, j) does not depend on any other boundary stress either.

Lemma 3.4. *If the smallest face of G is a quadrilateral and we place the boundary vertices as stated in* (3.7) *and* (3.8)*, then f_0 forms a convex polygon and the boundary stresses in* (3.8) *cancel the forces F.* □

Case 3: G contains no triangular and no quadrilateral face

The case if the smallest face of G is a pentagon is more complicated. We have $\binom{5}{2} = 10$ substitution stresses $\tilde{\omega}_{ij}$, but the adjacent ones do not count (by the same reasons given in the previous case). So we are left with five "diagonal" substitution stresses $\tilde{\omega}_{ij}$.

Like in the previous cases we determine a unique solution of the equation system by fixing some of the coordinates of the boundary face. However, we have to make more effort to guarantee the convexity of f_0. The following lemma helps us here:

Lemma 3.5. *We can relabel the boundary points for any stress* $(\tilde{\omega}_{ij})_{1\leq i,j\leq 5}$ *such that*

$$\tilde{\omega}_{35} \geq \tilde{\omega}_{24} \quad \textit{and} \quad \tilde{\omega}_{25} \geq \tilde{\omega}_{13}.$$

None of the interior stresses $\tilde{\omega}_{ij}$ *is zero.*

Proof. Without loss of generality we assume that the largest stress on an interior edge is $\tilde{\omega}_{35}$. If $\tilde{\omega}_{25} \geq \tilde{\omega}_{13}$ we are done. Otherwise we relabel the vertices by exchanging $\mathbf{p}_3 \leftrightarrow \mathbf{p}_5$ and $\mathbf{p}_1 \leftrightarrow \mathbf{p}_2$. To see that none of the interior $\tilde{\omega}$ values is zero take a look at the proof of Lemma 3.2. As a consequence of the construction there, $\tilde{\omega}_{ij}$ can only be zero if there is no interior edge incident to i in G. In this case (remember G is 3-connected) the triangle spanned by $\mathbf{p}_i$, $\mathbf{p}_{i-1}$ and $\mathbf{p}_{i+1}$ is contained in G. This contradicts the assumption that the smallest face of G is a pentagon. □

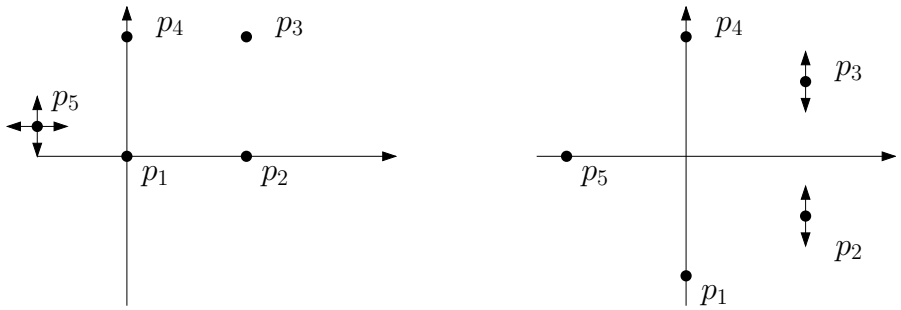
For the rest of this section we label the vertices such that Lemma 3.5 holds. The way we embed the f_0 depends on the substitution stresses $\tilde{\omega}_{ij}$.

Case 3A:
We assume that

$$\tilde{\omega}_{35}\tilde{\omega}_{14} + \tilde{\omega}_{14}\tilde{\omega}_{25} + \tilde{\omega}_{25}\tilde{\omega}_{24} + \tilde{\omega}_{13}\tilde{\omega}_{35} > \tilde{\omega}_{35}\tilde{\omega}_{25}. \tag{3.9}$$

In this case we assign

$$\mathbf{p}_1 = \begin{pmatrix}0\\0\end{pmatrix}, \mathbf{p}_2 = \begin{pmatrix}1\\0\end{pmatrix}, \mathbf{p}_3 = \begin{pmatrix}1\\1\end{pmatrix}, \mathbf{p}_4 = \begin{pmatrix}0\\1\end{pmatrix}, \mathbf{p}_5 = \begin{pmatrix}x_5\\y_5\end{pmatrix}.$$



(a) Case 3A (b) Case 3B

Figure 3.1: Placement of the boundary vertices.

Figure 3.1 (a) illustrates the location of the points. Together with the equations (3.4) and (3.5) we obtain as solution for $\mathbf{p}_5$:

$$x_5 = \frac{(\tilde{\omega}_{13} - \tilde{\omega}_{25} - \tilde{\omega}_{24})(\tilde{\omega}_{35} + \tilde{\omega}_{13} - \tilde{\omega}_{24})}{\tilde{\omega}_{35}\tilde{\omega}_{14} + \tilde{\omega}_{14}\tilde{\omega}_{25} + \tilde{\omega}_{25}\tilde{\omega}_{24} + \tilde{\omega}_{13}\tilde{\omega}_{35} - \tilde{\omega}_{35}\tilde{\omega}_{25}},$$

$$y_5 = \frac{\tilde{\omega}_{35} + \tilde{\omega}_{13} - \tilde{\omega}_{24}}{\tilde{\omega}_{35} + \tilde{\omega}_{25}}.$$

The boundary stresses ω_{ij} are more complicated expressions they are not necessary for further computations. For completeness we have listed them in Appendix A.

We have to check that f_0 forms a convex polygon. Clearly, $y_5 > 0$, since the $\tilde{\omega}_{ij}$'s are greater zero and $\tilde{\omega}_{35} \geq \tilde{\omega}_{24}$. Moreover $y_5 < 1$, because $\tilde{\omega}_{25} \geq \tilde{\omega}_{13}$. The numerator of x_5 is negative and due to (3.9) the denominator of x_5 is positive. Therefore, $x_5 < 0$ and f_0 forms a convex polygon.

Case 3B:
We assume the opposite of (3.9), namely

$$\tilde{\omega}_{35}\tilde{\omega}_{14} + \tilde{\omega}_{14}\tilde{\omega}_{25} + \tilde{\omega}_{25}\tilde{\omega}_{24} + \tilde{\omega}_{13}\tilde{\omega}_{35} \leq \tilde{\omega}_{35}\tilde{\omega}_{25}. \tag{3.10}$$

The coordinates for the boundary vertices are chosen as

$$\mathbf{p}_1 = \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \mathbf{p}_2 = \begin{pmatrix} 1 \\ y_2 \end{pmatrix}, \mathbf{p}_3 = \begin{pmatrix} 1 \\ y_3 \end{pmatrix}, \mathbf{p}_4 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \mathbf{p}_5 = \begin{pmatrix} -1 \\ 0 \end{pmatrix}.$$

See Figure 3.1(b) for an illustration. This leads to the solution

$$\begin{aligned} y_2 &= -2\frac{\tilde{\omega}_{24}\tilde{\omega}_{13} + \tilde{\omega}_{24}\tilde{\omega}_{35} + \tilde{\omega}_{25}\tilde{\omega}_{13} + 2\tilde{\omega}_{25}\tilde{\omega}_{35} - \tilde{\omega}_{13}^2 - 2\tilde{\omega}_{13}\tilde{\omega}_{35} - \tilde{\omega}_{35}\tilde{\omega}_{14}}{\tilde{\omega}_{24}\tilde{\omega}_{35} + \tilde{\omega}_{25}\tilde{\omega}_{13} + 2\tilde{\omega}_{25}\tilde{\omega}_{35}}, \\ y_3 &= 2\frac{\tilde{\omega}_{24}\tilde{\omega}_{13} + \tilde{\omega}_{24}\tilde{\omega}_{35} + \tilde{\omega}_{25}\tilde{\omega}_{13} + 2\tilde{\omega}_{25}\tilde{\omega}_{35} - \tilde{\omega}_{24}^2 - 2\tilde{\omega}_{24}\tilde{\omega}_{25} - \tilde{\omega}_{14}\tilde{\omega}_{25}}{\tilde{\omega}_{24}\tilde{\omega}_{35} + \tilde{\omega}_{25}\tilde{\omega}_{13} + 2\tilde{\omega}_{25}\tilde{\omega}_{35}}. \end{aligned}$$

The boundary stresses ω_{ij} are again complicated expressions and listed only in the Appendix.

The outer face is convex if $-2 < y_2 < y_3 < 2$. The inequalities $-2 < y_2$ and $y_3 < 2$ are equivalent to

$$\begin{aligned} -\tilde{\omega}_{13}^2 - \tilde{\omega}_{13}\tilde{\omega}_{35} - \tilde{\omega}_{35}\tilde{\omega}_{14} + \tilde{\omega}_{13}(\tilde{\omega}_{24} - \tilde{\omega}_{35}) < 0 \quad \text{and} \\ -\tilde{\omega}_{24}^2 - \tilde{\omega}_{24}\tilde{\omega}_{25} - \tilde{\omega}_{14}\tilde{\omega}_{25} + \tilde{\omega}_{24}(\tilde{\omega}_{13} - \tilde{\omega}_{25}) < 0. \end{aligned}$$

Both inequalities hold, because we add only negative summands on the left side. It remains to check if $y_2 - y_3 < 0$. First we get rid of the denominator and bring all negative summands on the right side. This leads to the equivalent inequality

$$\begin{aligned} \tilde{\omega}_{13}^2 + \tilde{\omega}_{24}^2 + 2\tilde{\omega}_{13}\tilde{\omega}_{35} + 2\tilde{\omega}_{24}\tilde{\omega}_{25} + \tilde{\omega}_{25}\tilde{\omega}_{14} + \tilde{\omega}_{35}\tilde{\omega}_{14} < 2\tilde{\omega}_{24}\tilde{\omega}_{35} + 2\tilde{\omega}_{25}\tilde{\omega}_{13} \\ + 4\tilde{\omega}_{25}\tilde{\omega}_{35} + 2\tilde{\omega}_{24}\tilde{\omega}_{13}. \end{aligned} \tag{3.11}$$

We observe that $\tilde{\omega}_{13}^2 \leq \tilde{\omega}_{25}\tilde{\omega}_{13}$ and $\tilde{\omega}_{24}^2 \leq \tilde{\omega}_{24}\tilde{\omega}_{35}$. Because of the assumption for case 3B we have $4\tilde{\omega}_{35}\tilde{\omega}_{25} > 2\tilde{\omega}_{13}\tilde{\omega}_{35} + \tilde{\omega}_{24}\tilde{\omega}_{25} + \tilde{\omega}_{25}\tilde{\omega}_{14} + \tilde{\omega}_{35}\tilde{\omega}_{14}$. Therefore, the right side of (3.11) is greater than its left side, which shows that $y_2 < y_3$ and f_0 forms a convex pentagon. This completes the case distinction and we conclude with:

Lemma 3.6. *If the smallest face of G is a pentagon and we place the boundary vertices as discussed above, then the outer face will be embedded as a convex polygon and the computed boundary stresses cancel the forces F.* □

We defined four different ways to embed G. The selected embedding depends on the combinatorial structure G. If G contains a triangular face we say it is of *type 3*. If it contains a quadrilateral but no triangular face G is of *type 4*. Otherwise the embedding depends on the substitution stresses induced by the combinatorial structure of G. If (3.9) holds (case 3A) G is of *type 5A* otherwise (case 3B) we say G is of type *type 5B*.

3.3 Lifting the Plane Embedding

We continue with the second stage of the algorithm and lift the plane embedding of G to $\mathbb{R}^3$. The technique how to lift an embedding with equilibrium stress is described in Chapter 2. We compute a plane H_i for every face f_i iteratively. The planes H_i are determined by a gradient $\mathbf{a}_i$ and a scalar d_i. We have

$$H_i : \quad z_i(\mathbf{p}) := \langle \mathbf{a}_i, \mathbf{p} \rangle + d_i.$$

Since the embedding of f_0 is convex, the boundary stresses must necessarily be negative, since otherwise the boundary vertices could not be in equilibrium with all interior stresses being positive. Thus we need not explicitly check the sign of the boundary stresses. The sign pattern of the stress implies that the lifting of the plane embedding gives a convex polytope (see Observation 2.8).

We begin the lifting by fixing the plane H_1, which contains an interior face f_1. For simplicity we set $\mathbf{a}_1 = (0, 0)^T$ and $d_1 = 0$. We compute the remaining planes face by face by using equation (2.5) and (2.6). Notice that it is sufficient to compute the height of every vertex only once. Thus, it is not necessary to compute the parameters of H_0 since we can determine the heights of $\mathbf{p}_1, \dots \mathbf{p}_k$ by some plane H_i of an interior face. Therefore, it is sufficient to use stresses on interior edges to compute the lifting. This simplifies the later analysis because all interior stresses are 1, whereas the boundary stresses are more complicated expressions.

3.4 Scaling to Integrality

In the third stage of the algorithm we compute scaling factors such that the embedding will have integer coordinates. We use different scaling factors for the x-direction and y-direction. Let S_x be the scaling factor for the x-coordinates and S_y be the scaling factor for the y-coordinates.

Lemma 3.7. *If the plane embedding consists of integer coordinates only, the z-coordinates of the lifted embedding are integral as well.*

Proof. The computation of the lifting starts with the face f_1. The gradient $\mathbf{a}_1$ and the value d_1 are integers. The parameters $\mathbf{a}_i$, d_i of the other planes H_i can be computed with help of equation (2.5) and (2.6). By an inductive argument it can be observed that (for interior faces) these parameters are integral as well. Evaluating the functions $z_i(\mathbf{p})$ boils down to multiply and add integers. Hence all z-coordinates are integral. □

As a consequence of Lemma 3.7 it is sufficient to scale to integer x and y-coordinates. Furthermore we observe:

Lemma 3.8. *If the boundary points are integral, the spring embedding yields coordinates that are multiples of* $1/\det \bar{L}$.

Proof. The interior plane coordinates are a result of equation (3.1). By Cramer's rule every coordinate can be expressed as

$$x_i = \det \bar{L}^{(i)} / \det \bar{L},$$

where $\det \bar{L}^{(i)}$ is obtained from $\bar{L}$ by replacing the i-th column of $\bar{L}$ by $A_{IB}\mathbf{x}_B$. Since $\det \bar{L}^{(i)}$ is integral, $\det \bar{L}$ is the denominator of x_i. The same holds for y_i. □

It remains to scale the plane embedding such that the boundary vertices get integer coordinates. Let S_x^B be the scaling factor that gives integer *boundary* x-coordinates and S_y^B be the scaling factor that gives integer *boundary* y-coordinates. Due to Lemma 3.8 $S_x := S_x^B \det \bar{L}$ and $S_y := S_y^B \det \bar{L}$ yield valid scaling factors. Since we choose integral scaling factors S_x^B and S_y^B no integer coordinate is scaled to a non-integer.

Let us now observe which factors are necessary to scale to integer boundary coordinates. Clearly the scaling factors depend on the type of G. If G is of type 3 then we need not to scale, since all boundary coordinates are either 0 or 1. If G is of type 4 we have to scale the y-coordinates only. We multiply with $S_y^B := (2\tilde{\omega}_{13} - \tilde{\omega}_{24}) \det \bar{L}$, which cancels the denominator of the only non-integer coordinate y_3. Notice that due to the Substitution Lemma S_y^B is an integer and therefore all integer boundary vertices remain integers.

If G is of type $5A$ we have to scale such that x_5 and y_5 become integral. This is achieved by

$$\begin{aligned} S_x^B &= (\tilde{\omega}_{35}\tilde{\omega}_{14} + \tilde{\omega}_{14}\tilde{\omega}_{25} + \tilde{\omega}_{25}\tilde{\omega}_{24} + \tilde{\omega}_{13}\tilde{\omega}_{35} - \tilde{\omega}_{35}\tilde{\omega}_{25})(\det \bar{L})^2 \\ S_y^B &= (\tilde{\omega}_{35} + \tilde{\omega}_{25}) \det \bar{L}. \end{aligned}$$

It can be easily checked that these factors are integral as well as $S_x^B x_5$ and $S_y^B y_5$.

It remains to introduce scaling factors when G is of type $5B$. Since the only non-integer boundary coordinates are y_2 and y_3, we need to scale in y-direction only. We choose

$$S_y^B = (\tilde{\omega}_{24}\tilde{\omega}_{35} + \tilde{\omega}_{25}\tilde{\omega}_{13} + 2\tilde{\omega}_{25}\tilde{\omega}_{35})(\det \bar{L})^2.$$

Again we observe that due to the Substitution Lemma S_y is integral as well as $S_y y_2$ and $S_y y_3$.

For every type of G there is a pair scaling factors such that the scaled boundary points are integral. Table 3.1 summarizes the discussion and lists the final scaling factors depending on the type of G.

type of G	scaling factors
3	$S_x = \det \bar{L}$ $S_y = \det \bar{L}$
4	$S_x = \det \bar{L}$ $S_y = (2\tilde{\omega}_{13} - \tilde{\omega}_{24})(\det \bar{L})^2$
$5A$	$S_x = (\tilde{\omega}_{35}\tilde{\omega}_{14} + \tilde{\omega}_{14}\tilde{\omega}_{25} + \tilde{\omega}_{25}\tilde{\omega}_{24} + \tilde{\omega}_{13}\tilde{\omega}_{35} - \tilde{\omega}_{35}\tilde{\omega}_{25})(\det \bar{L})^3$ $S_y = (\tilde{\omega}_{35} + \tilde{\omega}_{25})(\det \bar{L})^2$
$5B$	$S_x = \det \bar{L}$ $S_y = (\tilde{\omega}_{24}\tilde{\omega}_{35} + \tilde{\omega}_{25}\tilde{\omega}_{13} + 2\tilde{\omega}_{25}\tilde{\omega}_{35})(\det \bar{L})^3$

Table 3.1: The scaling factors S_x and S_y for the different types of G.

3.5 Analysis of the Grid Size

To bound the size of the coordinates of the integer embedding it is crucial to obtain a good bound for $\det \bar{L}$. There exists a connection between the number of spanning trees in G and $\det \bar{L}$. Let us first define:

Definition 3.9. *Let $\mathcal{B}$ a subset of vertices of G. A subgraph of G is called* spanning $\mathcal{B}$-forest *if*

- *it consists of $|\mathcal{B}|$ vertex disjoint trees covering all vertices of G,*
- *the roots of the trees are the vertices $\mathcal{B}$.*

In the following we use the set of boundary vertices B for the set $\mathcal{B}$. Let $\mathcal{F}_B(G)$ denote the number of spanning B-forests of G and $\mathcal{T}(G)$ the number of spanning trees of G. A generalization of the Matrix-Tree-Theorem is given in [70]. It states that the number of spanning B-forests of G is precisely $\det \bar{L}$. Moreover we can bound $\mathcal{F}_B(G)$ by $\mathcal{T}(G)$.

Lemma 3.10. *Let G be a planar graph with a distinguished face and let B be the set of vertices of this face. The number of spanning B-forests of G is bounded from above by*

$$\mathcal{F}_B(G) \leq \mathcal{T}(G).$$

Proof. Every spanning B-forest can be turned into a spanning tree by adding all boundary edges except $(1, 2)$. No two distinct spanning B-forests are associated with the same spanning tree. Therefore, the number of spanning trees exceeds the number of spanning B-forests. □

Notice that the bound of Lemma 3.10 holds only if the vertices B are the vertices of a single face, which is true in our setting.

It is easy to give an exponential upper bound for $\mathcal{T}(G)$:

Proposition 3.11 (Ribó Mor [70]). *1. The number of spanning trees in a graph is bounded by the product of all vertex degrees:*

$$\mathcal{T}(G) \leq \prod_i \deg(v_i).$$

2. For a planar graph, we have $\prod_i \deg(v_i) < 6^n$.

Proof. 1. Pick an arbitrary vertex v_1. Consider all directed graphs that are obtained by choosing an outgoing edge in G out of every vertex except v_1. The number of these directed graphs is given by $\prod_{i\neq 1} \deg(v_i)$. By ignoring the edge orientations, one obtains all spanning trees (and many graphs that are not spanning trees).

2. This follows from the arithmetic-geometric-mean inequality and the fact that $\sum_i \deg(v_i) < 6n$, which is a consequence of Euler's formula. □

Sharper bounds for $\mathcal{T}(G)$ have been given by Ribó Mor [70], see also Rote [74]. These bounds take into account whether G contains triangular or quadrilateral faces:

$$\begin{aligned} \text{if } G \text{ is of type 3:} \quad \mathcal{F}_B(G) &\leq 5.\bar{3}^n, \\ \text{if } G \text{ is of type 4:} \quad \mathcal{F}_B(G) &\leq 3.529988^n, \\ \text{if } G \text{ is of type 5A/5B:} \quad \mathcal{F}_B(G) &\leq 2.847263^n. \end{aligned}$$

Since we know upper bounds for the $\tilde{\omega}$ values (by Lemma 3.2) and $\det \bar{L}$ (by the previous discussion) we can bound the size of the integer coordinates of the embedding of G. We start with bounding the x and y-coordinates. Let Δx denote an upper bound for the difference between the largest and the smallest x-coordinate. Δy is defined in the same way for the y-coordinates.

Again we have to discuss the 4 cases separately. If G is of type 3 then clearly $\Delta x = \Delta y = \det \bar{L}$. If G is of type 4 the largest x-coordinate is $2S_x$ and the smallest zero. Thus we have $\Delta x = 2 \det \bar{L}$. The largest y coordinate is obtained at y_4 (remember $\tilde{\omega}_{13} \geq \tilde{\omega}_{24}$), therefore $\Delta y = (2\tilde{\omega}_{13} - \tilde{\omega}_{24})(\det \bar{L})^2$. Let us now assume G is of type $5A$. The value of Δx is given by $x_2 - x_5$. Evaluating this expression leads to

$$\Delta x = (\tilde{\omega}_{25}(\tilde{\omega}_{13} + \tilde{\omega}_{14}) + \tilde{\omega}_{35}(\tilde{\omega}_{14} + \tilde{\omega}_{25}) - (\tilde{\omega}_{13} - \tilde{\omega}_{24})^2)(\det \bar{L})^3.$$

Since the smallest y-coordinate is zero we have $\Delta y = y_3$, which equals $(\tilde{\omega}_{35} + \tilde{\omega}_{25})(\det \bar{L})^2$. It remains to discuss the case if G is of type $5B$. *Before* the scaling the coordinates fulfill $-1 \leq x \leq 1$ and $-2 < y < 2$. Combining these inequalities with the scaling factors yields $\Delta x = 2 \det \bar{L}$ and $\Delta y = 4(\tilde{\omega}_{24}\tilde{\omega}_{35} + \tilde{\omega}_{25}\tilde{\omega}_{13} + 2\tilde{\omega}_{25}\tilde{\omega}_{35})(\det \bar{L})^3$. We sum up the results for Δx and Δy in Table 3.2 and Table 3.3.

We finish the analysis of the necessary grid size by calculating the size of the z-coordinates. We observe that all z-coordinates are negative. Thus it suffices to compute the smallest z-coordinate.

type of G	Δx
3	$\det \bar{L}$
4	$2 \det \bar{L}$
$5A$	$(\tilde{\omega}_{25}(\tilde{\omega}_{13}+\tilde{\omega}_{14})+\tilde{\omega}_{35}(\tilde{\omega}_{14}+\tilde{\omega}_{25})-(\tilde{\omega}_{13}-\tilde{\omega}_{24})^2)(\det \bar{L})^3$
$5B$	$2 \det \bar{L}$

Table 3.2: The values Δx depending on the type of G.

type of G	Δy
3	$\det \bar{L}$
4	$(2\tilde{\omega}_{13}-\tilde{\omega}_{24})(\det \bar{L})^2$
$5A$	$(\tilde{\omega}_{35}+\tilde{\omega}_{25})(\det \bar{L})^2$
$5B$	$4(\tilde{\omega}_{24}\tilde{\omega}_{35}+\tilde{\omega}_{25}\tilde{\omega}_{13}+2\tilde{\omega}_{25}\tilde{\omega}_{35})(\det \bar{L})^3$

Table 3.3: The values Δy depending on the type of G.

Lemma 3.12. *Let $G(\mathbf{p})$ be an integral embedding of a graph with n points with equilibrium stress ω and let the stress on all interior edges be 1. The difference between two x-coordinates is less than Δx and the difference between two y-coordinates less than Δy. Then we have an integral lifting with*

$$0 \leq z_i < 2n\Delta x\Delta y$$

for all z-coordinates z_i.

Proof. Due to Lemma 3.7 we know that there exists an integral lifting for the setting described in the lemma. We place an interior face f_1 in the xy-plane and compute the lifting by using the stress on interior edges. We choose as face f_1 a face that shares an edge with the outer face f_0. Furthermore we assume that the boundary point farthest away from the line that contains $f_1 \cap f_0$ is located in the origin (let this point be $\mathbf{p}_1$). This is no restriction since a translation of the embedding does not interfere with the lifting. The lifted polytope lies between the planes H_0 and H_1 and all its z-coordinates are negative. We notice that the smallest z-coordinate is realized at $\mathbf{p}_1$.

Let f_k be an interior face that containes $\mathbf{p}_1$. The z-coordinate of $\mathbf{p}_1$ equals $z_k(\mathbf{p}_1)$, which is given as

$$z_1 = z_k(\mathbf{p}_1) = \langle \mathbf{a_k}, \mathbf{p}_1 \rangle + d_k = d_k.$$

The value d_k can be computed with help of the equation (2.6). Let $\mathcal{C}$ be a set of interior edges that are crossed by "walking" from f_1 to f_k. Due to Euler's formula G has at most $2n-4$ triangles. No face is entered twice and thus every face is providing exactly one edge (the edge where the "walk" leaves the face) to the set $\mathcal{C}$. This implies that $\mathcal{C}$ includes at most $2n-3$ edges. We ignore the orientation of the edges at this place since it does not matter for bounding d_k.

We deduce

$$\begin{aligned} -d_k &\leq \sum_{(i,j)\in\mathcal{C}} |\langle \mathbf{p}_i, \mathbf{p}_j^{\perp}\rangle| \\ &\leq 2n \max\{|\langle \mathbf{p}_i, \mathbf{p}_j^{\perp}\rangle| \; | 1 \leq i,j \leq n\}. \end{aligned}$$

For two points $\mathbf{p}_i$,$\mathbf{p}_j$ we have

$$\langle \mathbf{p}_i, \mathbf{p}_j^{\perp}\rangle = |x_i y_j - x_j y_i| = \det \begin{pmatrix} x_i & x_j & 0 \\ y_i & y_j & 0 \\ 1 & 1 & 1 \end{pmatrix} = 2[i,j,1].$$

Thus $\langle \mathbf{p}_i, \mathbf{p}_j^{\perp}\rangle$ equals two times the negative area of the triangle spanned by $\mathbf{p}_i$,$\mathbf{p}_j$ and $\mathbf{p}_1$. This triangle is contained inside the embedded outer face f_0 and also inside a rectangle with edge lengths Δx and Δy. A rectangle has at least twice the area of an inscribed triangle. To see this, observe that an inscribed triangle with the largest area must have one of the rectangle edges as base and the other as height. Thus $2[i,j,1] > -\Delta x \Delta y$ and the smallest z-coordinate is larger than $-2n\Delta x\Delta y$. Translating the polytope such that all z-coordinates become positive yields the bound of the lemma. □

By applying Lemma 3.12 we compute the bounds for the z-coordinates using the values of Δx and Δy listed in Table 3.2 and 3.3. We conclude with the main theorems of this chapter.

Theorem 3.13. *Every 3-polytope with n vertices whose graph contains at least a triangle can be realized on an integer grid with*

$$\begin{aligned} 0 &\leq x_i, y_i < 5.\bar{3}^n, \\ 0 &\leq z_i < 2n \; 28.\bar{4}^n. \end{aligned}$$

Theorem 3.14. *Every 3-polytope with n vertices whose graph contains at least one quadrilateral face can be realized on an integer grid with*

$$\begin{aligned} 0 &\leq x_i < 2 \; 3.531^n, \\ 0 &\leq y_i < 2n \; 12.462^n, \\ 0 &\leq z_i < 8n^2 \; 46.381^n. \end{aligned}$$

For the most general theorem we have to combine the analysis for the cases $5A$ and $5B$. We rotate the embedding if G is of type $5B$ by exchanging the x and y-coordinates to obtain a better bound. The largest absolute z-coordinate is realized in case $5A$.

Theorem 3.15. *Every 3-polytope with n vertices can be realized on an integer grid with*

$$\begin{aligned} 0 &\leq x_i < 16n^2 \; 23.083^n, \\ 0 &\leq y_i < 2n \; 8.107^n, \\ 0 &\leq z_i < 16n^4 \; 187.128^n. \end{aligned}$$

Notice that we can improve the constant factor of the z-coordinate by a more careful analysis. This can be achieved by placing the face f_0 in the xy-plane and then compute the lifting using the interior edges but also one boundary edge. As mentioned before the structure of the stresses on the boundary edges is more complicated. Since the improvement would only be a constant factor we decided to present the easier analysis with help of Lemma 3.12. The more complicated analysis can be found at [71].

We see that exponentially large coordinates suffice to embed G as 3-polytope. The exponential growth of the size of the coordinates is determined by $(\det \bar{L})^5$.

Corollary 3.16. *Every 3-polytope with n vertices can be realized with integer coordinates of size $O(2^{7.55n})$.*

3.6 An Example: The Dodecahedron

The regular dodecahedron is one of the five Platonic solids. It has 20 vertices, 30 edges and 12 faces that are regular pentagons. Figure 3.2 shows the graph and a 3-dimensional realization of the dodecahedron. This example is motivated by the fact that all faces of the dodecahedron are 5-gons. Thus we have to apply the more involved methods for an integer embedding. Since the do-

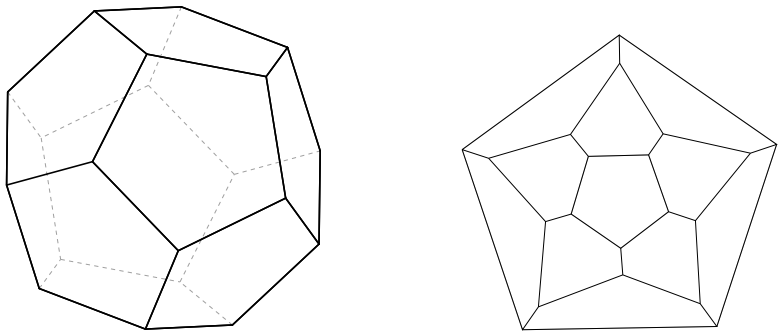


Figure 3.2: The dodecahedron and its graph.

decahedron is symmetric it makes no difference which face we choose as the outer face. We start the computation with calculating the $\tilde{\omega}$ values. We obtain for all the stresses $\tilde{\omega}_{13}, \tilde{\omega}_{14}, \tilde{\omega}_{24}, \tilde{\omega}_{25}$ and $\tilde{\omega}_{35}$ the value $36/449$. The fact that all these stresses have the same value is again due to the symmetry of the dodecahedron. Because the outer face is a 5-gon, we have to check if the graph of the dodecahedron is of type $5A$ or $5B$. Evaluating (3.9), shows that the graph is of type $5A$. With help of the $\tilde{\omega}$ values we compute the coordinates of the boundary vertices. We obtain

$$\mathbf{p}_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \mathbf{p}_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \mathbf{p}_3 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \mathbf{p}_4 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \mathbf{p}_5 = \begin{pmatrix} -1/3 \\ 1/2 \end{pmatrix}.$$

We apply Tutte's method to compute the coordinates of the interior points. The result is depicted in Figure 3.3. Next, we scale the 2d-embedding as described

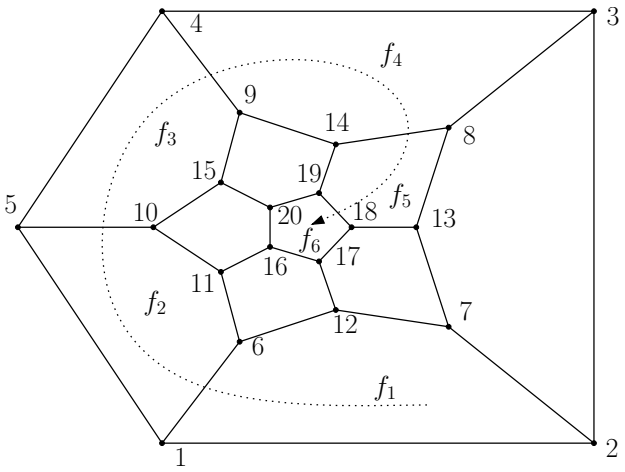


Figure 3.3: The plane embedding of the dodecahedron.

plane H_i	$\mathbf{a}_i/(\det \bar{L})$	$d_i/(\det \bar{L})^2$
$i = 2$	$(15192, -563283072)^T$	0
$i = 3$	$(15192, -1545537024)^T$	31754305760256
$i = 4$	$(0, -2108820096)^T$	68173936063488
$i = 5$	$(-2520, -1289499264)^T$	34719614060544
$i = 6$	$(2592, -1054410048)^T$	20087475047424

Table 3.4: The planes H_i defining the lifting of the dodecahedron.

in Section 3.4. We obtain $\det \bar{L} = 403202$. This imples the scaling factors

$$\begin{aligned} \bar{S}_x &= 1\,264\,158\,727\,403\,904, \\ \bar{S}_y &= 26\,069\,428\,512. \end{aligned}$$

We continue with the lifting of the plane embedding to $\mathbb{R}^3$. The faces are lifted incrementally as described in Section 3.3. It is sufficient to lift the five faces $f_2, \ldots, f_6$ marked in Figure 3.3. All vertices are incident to one of these faces. Lifting the faces in increasing order yields the planes described in Table 3.4. Finally, we compute the coordinates of the polytope by plugging the 2d-coordinates into the equation for the corresponding plane. The result is shown in Figure 3.4. We have scaled the z-coordinates down to obtain an illustrative picture. We observe that the highest absolute coordinate is

$$z_3 = -11\,083\,163\,098\,782\,678\,334\,820\,352.$$

Notice that $|z_3|$ is approximately $2^{83.19}$, which is indeed quite large, but smaller than the bound 2^{151} of Corollary 3.16.

The computed embedding allows a smaller integer realization. Due to the fact that the greatest common divisor of the x-coordinates is 938499426432 and

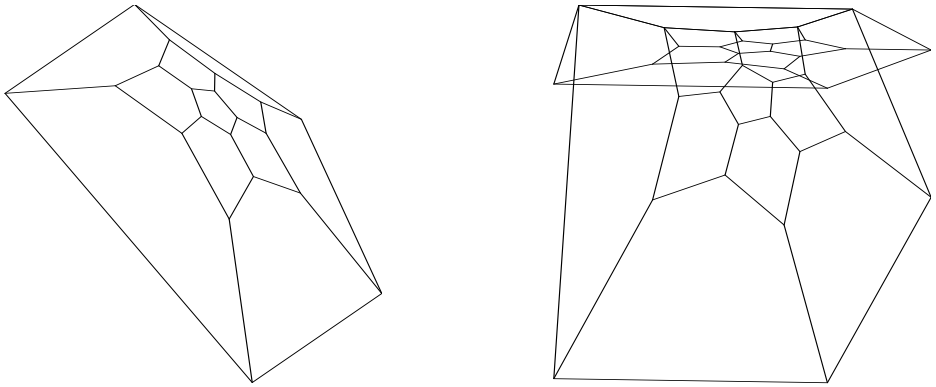


Figure 3.4: Two pictures of the dodecahedron embedded with our algorithm with scaled z-axis. The right picture includes also the equilibrium stressed plane embedding.

the greatest common divisor of the y-coordinates equals 29030544, scaling down by these factors gives a smaller integer embedding. We obtain an integral plane embedding on the $[-27, 1347] \times [0, 898]$ grid. The corresponding z-coordinates range between zero and -406497.

Chapter 4

Towards a Lower Bound for Grid Embeddings of 3-Polytopes

4.1 Lower Bound for Integer Grid Embeddings

In the previous chapter we showed how to embed a planar 3-connected graph G with n vertices as 3-polytope with integer coordinates. Our algorithm produces coordinates exponential in n. Are exponential coordinates really necessary or can we do better and realize the polytope with coordinates polynomial in the size of n? We discuss the question in the following. Based on experiments we provide several arguments for the conjecture:

Conjecture 4.1. *There is a family of planar 3-connected graphs that cannot be realized as 3-polytopes with integer coordinates polynomial in the size of the number of vertices of the graph.*

Before we talk about the reasons which let us believe that Conjecture 4.1 is true, we study provable lower bounds. Actually the best known lower bound is based on a 2d problem. The orthogonal projection of the polytope into any plane has to be a *strictly* convex embedding of G. Due to Acketa and Žunić [1, 2] we know that embedding a simple polygon strictly convex on the grid requires coordinates of size $\Omega(n^{3/2})$ (see also Thiele [85]). It is also possible to have one coordinate in $O(n)$ at the expense of the other (e.g. the n-gon can be also realized on a $O(n) \times O(n^2)$ grid).

Consider the graph depicted in Figure 4.1. The graph contains two convex $(n/2)$-gons. Since the two large faces touch each other they have to lie on planes with different gradient in the 3d embedding. Only one face of the large faces can be aligned orthogonally to the xy-plane. The orthogonal projection of the other large face into the xy-plane has to be a 2d integral realization. Thus, the xy-coordinates are bounded by $\Omega(n^{3/2})$. On the other hand, the same argument holds for the yz-plane and therefore also the yz-coordinates are bounded by $\Omega(n^{3/2})$. This implies that at least one coordinate has to be in $\Omega(n^{3/2})$. Furthermore, we can deduce that if one coordinate is bounded by $O(n^{3/2})$ the other coordinates must be bounded by $\Omega(n^{3/2})$. As in the 2d setting one can improve the size of one coordinate at the expense of another.

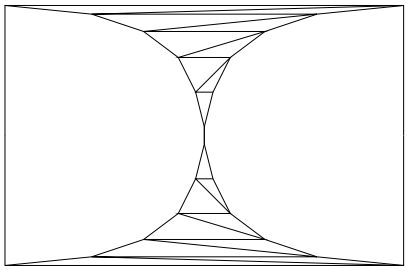


Figure 4.1: Lower bound example that requires a coordinate of size $\Omega(n^{3/2})$ when embedded as 3-polytope on $\mathbb{Z}^3$.

4.2 Experimental Results

To justify Conjecture 4.1 we introduce a family of graphs with large z-coordinates for some of their natural embeddings. We denote the graphs of the family by $\Gamma_k = (\{v_1, \ldots, v_k\}, E_k)$, where the parameter k indicates the number of vertices of the graph. We define the graphs Γ_k inductively. Γ_4 is the complete graph on four vertices K_4. For any $k > 4$ we define Γ_k by its set of edges

$$E_k := \begin{cases} E_{k-1} \setminus \{(1, k-1)\} \cup \{(1,k), (2,k), (k, k-1)\} & k \text{ is odd,} \\ E_{k-1} \setminus \{(2, k-1)\} \cup \{(1,k), (2,k), (k, k-1)\} & k \text{ is even.} \end{cases}$$

See the graph Γ_{10} as example shown in Figure 4.2.

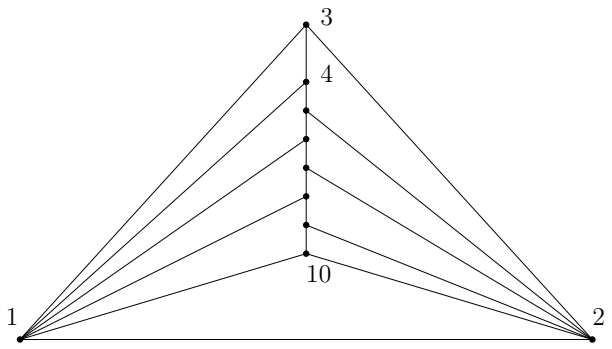


Figure 4.2: The graph Γ_{10}.

Our goal is to find small grid embeddings of the graphs Γ_k for small values of k. We proceed as follows: We choose two natural plane embeddings of Γ_k by giving every vertex v_i a coordinate x_i and y_i. Then we compute the smallest integral lifting of the plane embedding by solving a linear integer program (IP).

Before we show how to set up an IP-formulation for the lifting we introduce

two natural embeddings for Γ_k. We give the first three vertices the coordinates:

$$\begin{aligned} x_1 &= -2k \quad , \qquad y_1 = -k, \\ x_2 &= 2k \quad , \qquad y_2 = -k, \\ x_3 &= 0 \quad , \qquad y_3 = 0. \end{aligned}$$

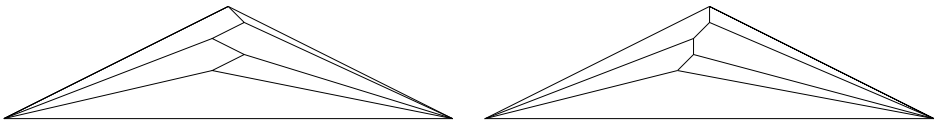
The coordinates of the first embedding are chosen as

$$x_k = (-1)^k, y_k = -k.$$

The coordinates of the second embedding are chosen as

$$x_k = -\lfloor (i-3)/2 \rfloor, y_k = -k.$$

Convince yourself that these embeddings are strictly convex by looking at the examples in Figure 4.3.



embedding 1 embedding 2

Figure 4.3: The two natural embeddings for Γ_7.

The computation of the minimal integer lifting is done by the following IP. The variables in the program are the z-coordinates of the vertices and a special variable called z_{max}. Our objective is to minimize z_{max} under the constraint that all other z-coordinates are not greater than z_{max} and positive. Furthermore, we have to model the fact that all edges of the lifted surface are convex. An edge (i, j) results in a convex edge, if the oriented tetrahedron spanned by the oriented patch $(i, j|l, r)$ has positive volume. The oriented volume can be expressed as follows (we make use of the square bracket notation introduced in Chapter 2):

$$z_i[j,k,l] - z_j[i,k,l] + z_k[i,j,l] - z_l[i,j,k]. \tag{4.1}$$

It can be observed that (4.1) is integral. Therefore it is at least 1/6 if it is greater zero. For any of the quadrilateral faces the volume spanned by the corresponding tetrahedron has to be zero. Plugging everything together gives the following IP-formulation:

$$\begin{aligned} z_{max} &\rightarrow \min, \\ z_i &\leq z_{max} && 3 < i \leq n, \\ 0 &\leq z_i && 3 < i \leq n, \\ z_i[j,k,l] - z_j[i,k,l] + z_k[i,j,l] - z_l[i,j,k] &\geq 1/6 && (i,j|k,l) \text{ oriented patch}, \\ z_i[j,k,l] - z_j[i,k,l] + z_k[i,j,l] - z_l[i,j,k] &= 0 && (i,k,j,l) \text{ quadri. face}, \\ z_1, z_2, z_3 &= 0. \end{aligned}$$

Γ_k	height of the lifting (z_{max})	
	first embedding	second embedding
$k = 6$	161	417
$k = 7$	15.525	949
$k = 8$	40.455	43.252
$k = 9$	346.155	17.330
$k = 10$	99.888.789	657.882
$k = 11$	26.288.093	40.596.825
$k = 12$	199.280.705	8.981.680

Table 4.1: Height of the minimal integer lifting of Γ_k.

The IP-formulation allows us to compute optimal liftings for small values of k by computer software. We used CPLEX 10.200 as IP-solver is based on a branch and bound technique. Solving the IP for Γ_{13} lasted approximately 32 hours on a dual P4 Xeon 2,6 GHz processor system with 2 GB main memory. We list the results in Table 4.1. One can observe that even though the 2d-coordinates are small, the necessary z-coordinates are very large. Notice that the growth is not monotonic.

Figure 4.4 shows the logarithmic plot of the values listed in Table 4.1. We

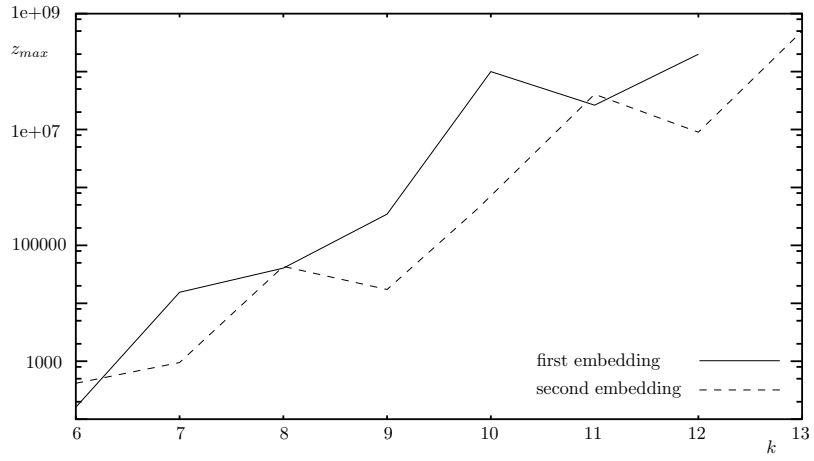
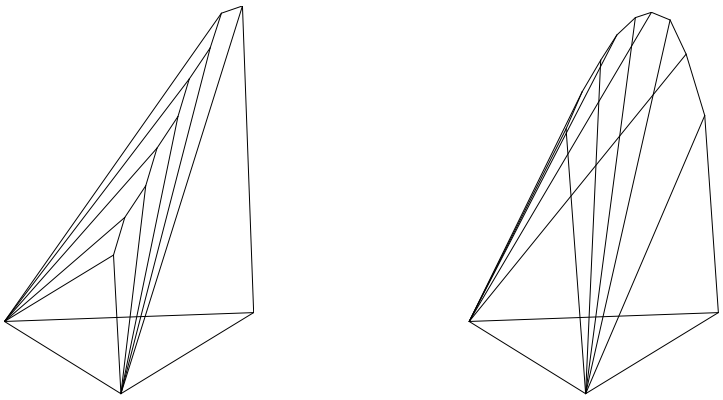


Figure 4.4: Plot of the experimental results.

depict the optimal solutions for Γ_{10} in Figure 4.5. Notice that the pictures are scaled down to get a reasonable illustration. Interestingly, the obtained embeddings look different. In the first embedding the highest vertex is v_4, whereas in the second embedding vertex v_7 defines the peak value.

Since we have only looked at two special plane embeddings, our results are only a weak evidence for Conjecture 4.1. We believe that the graphs Γ_k are candidates for graphs that do not have a polynomial sized integer embedding as 3-polytope.



embedding 1 embedding 2

Figure 4.5: The two lifted optimal embeddings for Γ_{10}.

Chapter 5

Optimizing over the Polytope of Pointed Pseudo-Triangulations

5.1 Introduction to Pseudo-Triangulations

5.1.1 Classification of Pseudo-Triangulations

Pseudo-triangulations are geometric and combinatorial objects. They have connections and applications in many areas like rigidity theory, motion planning and visibility. They are widely used as data structures for algorithms in computational geometry and topology. We refer to the introduction chapter for a motivation for the investigation of pseudo-triangulations.

Most generally, a pseudo-triangulation is defined for an *augmented polygon*, which is roughly speaking a polygon that might contain interior points. More formally:

Definition 5.1 (Augmented polygon[1], Domain). *Let P be a set of n points in the plane and let D denote a simple polygon whose vertices are a subset of P and no vertex of P lies outside D. We call the pair (D, P)* augmented polygon. *The set D is called the* domain.

A point $\mathbf{p}_i$ in P has the coordinates x_i and y_i. If D is not specified, we use the convex hull of P as domain.

Definition 5.2 (Pseudo-triangle). *A* pseudo-triangle *is a polygon with exactly three vertices whose interior angle is less than π.*

See Figure 5.1 for examples of pseudo-triangles. We call a vertex that realizes an angle smaller than π in the interior of a polygon a *corner* of the polygon. Every other vertex is named *non-corner*. The definition of a pseudo-triangle can be naturally extended for any pseudo-k-gon. Thus, a pseudo-quadrilateral denotes a polygon with exactly four interior angles smaller than π.

[1]The term *pointgon* is also used for augmented polygon.

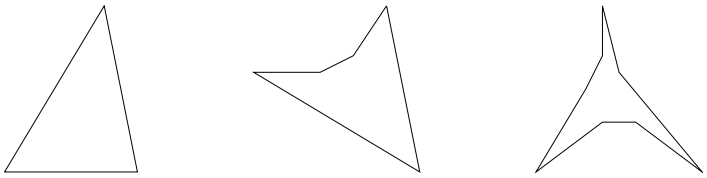
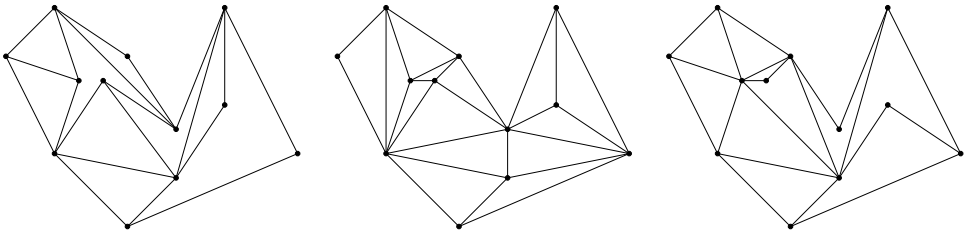


Figure 5.1: Three examples of pseudo-triangles.

Definition 5.3 (Pseudo-triangulation). *A* pseudo-triangulation *of an augmented polygon (D, P) is a tiling of D into pseudo-triangles, such that every point in P is a vertex in at least one pseudo-triangle.*

If all pseudo-triangles in a pseudo-triangulations are triangles, we obtain a *triangulation*. Triangulations are a subclass of pseudo-triangulations.



(a) (b) (c)

Figure 5.2: The same augmented polygon with a pointed pseudo-triangulation (a) a triangulation (b) and a pseudo-triangulation that is neither pointed nor a triangulation (c).

There exists another interesting subclass within the set of all pseudo-triangulations. These are the so-called *pointed pseudo-triangulations*, which have the property that every vertex of the planar region is incident to an angle greater than π. If a vertex contains such a big angle it is called *pointed*. One can see triangulations and *pointed* pseudo-triangulations as anti-poles of the set of all pseudo-triangulation. In the following we abbreviate the term *pointed pseudo-triangulations* with ppt. Figure 5.2 shows different examples of pseudo-triangulations.

If the domain D of the augmented polygon (D, P) equals the convex hull of P, we consider the pseudo-triangulation of (D, P) as the pseudo-triangulation of the point set P. On the other hand, if the vertices of D equal P we say that the pseudo-triangulation of (D, P) is the pseudo-triangulation of the simple polygon D. In this thesis we study either pseudo-triangulations of point sets or of simple polygons.

5.1.2 Properties of Pseudo-Triangulations

We list some properties of pseudo-triangulations for point sets briefly. For a comprehensive discussion we redirect the interested reader to the survey written by Rote, Santos and Streinu [76].

Due to Euler's formula we know that a triangulation with n vertices has $3n - 3 - k$ edges if the outer face is a k-gon. By a counting argument one can show that a ppt has $2n - 3$ edges [84]. Furthermore, a ppt is an embedding of a minimal rigid graph (also [84]). Conversely any planar minimal rigid graph can be embedded as ppt [45].

Pseudo-triangulations can be transformed by local operations called (edge exchanging) *flips*. A flip exchanges an edge of the pseudo-triangulation. Removing an edge of a pseudo-triangulation creates a pseudo-quadrilateral. There is exactly one new position for an edge to decompose the pseudo-quadrilateral into two new (different) pseudo-triangles. See Figure 5.3 for an example of an edge-exchanging flip. Notice that there exist other types of flips (see [6, 5]). Throughout the thesis we consider the *edge-exchanging* flip only and omit the term edge-exchanging in the following.

Any two pointed pseudo-triangulations can be transformed into each other by using at most $O(n \log n)$ flips [16]. Triangulations can also be transformed into each other by flips, but in contrast to pointed pseudo-triangulations $O(n^2)$ flips might be necessary. The number of flips that is needed is known as the *flip distance*.

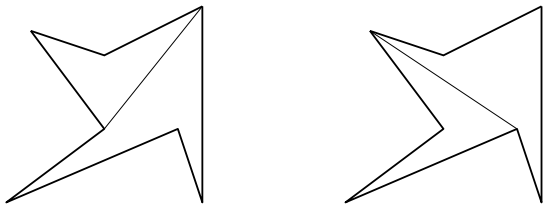


Figure 5.3: The edge exchanging flip.

5.1.3 Spatial Embedding of Pseudo-Triangulations

A pseudo-triangulation can represent two-dimensional polyhedral terrains in $\mathbb{R}^3$. We study this phenomenon in the following section. We restrict ourselves to pseudo-triangulations of simple polygons. The tools discussed here work also for the general setting with points in the interior of the domain. However, for our purposes the simple polygon setting is sufficient.

Assume that we have given some height assignment of the points in P. Let $h : P \to \mathbb{R}$ be a function that describes this assignment. We write h_i for $h(\mathbf{p}_i)$.

We say that a function $f : D \to \mathbb{R}$ is locally convex, if f is convex on every line segment interior in D. For every h we define the function f_h^* as follows:

Definition 5.4. *For a given $h : P \to \mathbb{R}$ let the function $f_h^* : D \to \mathbb{R}$ denote the*

maximal locally convex function over D that fulfills the condition

$$f_h^*(\mathbf{p}_i) \leq h_i, \tag{5.1}$$

for every $\mathbf{p}_i$ of P.

Observe that f_h^* is unique, because it is the pointwise maximum over all locally convex functions. It was observed in [6] that f_h^* defines a polyhedral terrain, which we name $\mathcal{F}_h^*$. In the same paper it is shown that the projection of the non-linearities of $\mathcal{F}_h^*$ onto D produces a (subset) of a pseudo-triangulation. We call this graph $\mathcal{PT}_h$.

A pseudo-triangulation that can be represented as $\mathcal{PT}_h$ is called *regular*. Notice that not all pseudo-triangulations are regular. For example two triangles sharing an edge, where one triangle is contained inside the other, give a pointed pseudo-triangulation that cannot be represented by a surface $\mathcal{F}_h^*$. Also within the class of triangulations we have regular and non-regular triangulations. If we want to emphasize that $\mathcal{PT}_h$ corresponds to a triangulation we write $\mathcal{T}_h$ instead of $\mathcal{PT}_h$.

An example of a regular pseudo-triangulation is shown in Figure 5.4. If the restrictions defined by (5.1) hold with equality (i.e. $f_h^*(\mathbf{p}_i) = h_i$), we call the vertex $\mathbf{p}_i$ *complete* vertex, otherwise, if $f_h^*(\mathbf{p}_i) < h_i$, we call $\mathbf{p}_i$ *incomplete* vertex. Because $\mathcal{F}_h^*$ is piecewise linear, the height $f_h^*(\mathbf{p}_i)$ of an incomplete vertex

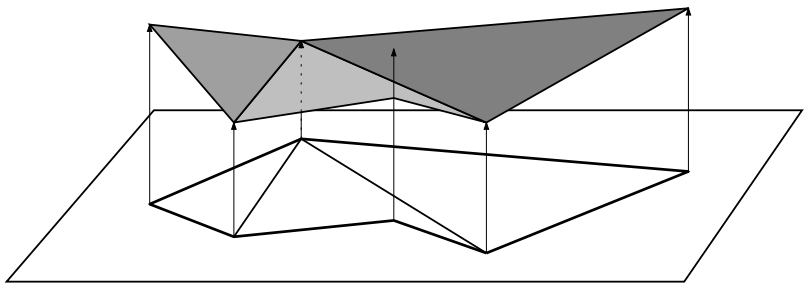


Figure 5.4: A regular pseudo-triangulation.

is given by a plane that is spanned by three other vertices $\mathbf{p}_a, \mathbf{p}_b$ and $\mathbf{p}_c$ (under a generic assumption). It was observed in [6] that the incomplete vertex has to be on a concave chain on the pseudo-triangle with the three corners $\mathbf{p}_a, \mathbf{p}_b$ and $\mathbf{p}_c$.

If all vertices of P are complete the pseudo-triangulation induced by $\mathcal{F}_h^*$ becomes a triangulation. This can be obtained by a set of heights h_i whose lower convex hull in $\mathbb{R}^3$ contains the lifted point set P completely. Regular triangulations are also known as the dual of power diagrams, or weighted Delaunay triangulations [11].

5.1.4 Overview of the Results

In this chapter we present a series of smaller independent results. In Section 5.2 we show that a greedy strategy to construct $\mathcal{F}_h^*$ by flips needs superpolynomial

many flips, even when every flips brings us "closer" to $\mathcal{F}_h^*$. The results there are obtained for pseudo-triangulations of simple polygons. In Section 5.3 we introduce two polytopes that can be used to study pointed pseudo-triangulations of point sets. We use these polytopes in the remaining part of the chapter. In Section 5.4 we apply the lifting technique to obtain an relation between pointed pseudo-triangulations in the two polytopes. The observation in this section leads to further results which we present in Section 5.5–Section 5.7. Section 5.5 and Section 5.6 study pointed pseudo-triangulations of point sets, the results in Section 5.7 are obtained for pointed pseudo-triangulations of simple polygons.

5.2 Long Improving Flip Sequences in Simple Polygons

In the previous section we have studied *convex* terrains that corresponds to pseudo-triangulations. We sketch now how to construct not necessary convex polyhedral terrain as spatial embeddings of pseudo-triangulations. Let (D, P) define a simple polygon and $\mathcal{PT}$ a fixed pseudo-triangulation of it. Let $\bar{h}$ be a height assignment for the corners of D. For each choice of $\bar{h}$ exists a unique polyhedral terrain $\bar{\mathcal{F}}_{\bar{h}}$ above the domain D that respects $\bar{h}$ and whose edges project vertically to (a subset of) the edges of $\mathcal{PT}$ [6, Surface Theorem]. Let h be the extension of $\bar{h}$ that fulfills

$$h_i := \begin{cases} \bar{h}_i & \text{if } \mathbf{p}_i \text{ is a corner in } D \\ \max_j\{\bar{h}_j\} & \text{else} \end{cases} .$$

Due to construction the maximal locally convex function given by (D, P) and h defines all non-corners of D as incomplete vertices. Therefore, in $\mathcal{F}_h^*$, every vertex is incident to an angle greater π and hence $\mathcal{PT}_h$ is pointed.

The terrain $\bar{\mathcal{F}}_{\bar{h}}$ is not necessarily convex, but it can be transformed to $\mathcal{F}_h^*$ by a sequence of flips. Flipping an edge in $\mathcal{PT}$ that is concave at $\bar{\mathcal{F}}_{\bar{h}}$ creates a pseudo-triangulation $\mathcal{PT}'$. The newly added edge in $\mathcal{F}'_{\bar{h}}$ is convex, but other convex edges might become concave after the flip. Flipping away all concave edges result in a sequence of pseudo-triangulations. This process terminates and the last generated pseudo-triangulation coincides with $\mathcal{PT}_h$ [6, Optimality Theorem]. A flip sequence that constantly removes concave edges is called *improving* flip sequence. In general flipping to optimality requires more flip types than the edge-exchanging flip. But if we start with a pointed pseudo-triangulation $\mathcal{PT}$ edge exchanging flips suffice to flip to $\mathcal{PT}_h$. For triangulations the improving flip sequence can be built out of edge exchanging flips as well.

Any improving flip sequence between regular triangulation has length at most $O(n^2)$. This is due to the fact that no edge reappears in the flip sequence. We know that this bound is sharp, even for convex polygons (see Edelsbrunner [34]).

In the pseudo-triangulation setting an edge might reappear in an improving flip sequence towards $\mathcal{PT}_h$ [5]. We show that this phenomenon results in fact in long sequences. As a consequence, flipping away *any* concave edge greedily is not a good strategy to construct $\mathcal{PT}_h$.

Theorem 5.5. *There exists a simple polygon that provides an improving flip sequence of length* $n^{\Theta(\log n)}$.

The proof of the theorem is given by the following construction. We introduce a polygon as building block. This polygon is called *elevator* since it adjusts the height of a distinguished vertex $\mathbf{p}_t$. See Figure 5.5 for an illustration. The elevator contains m vertices. The vertices on the bottom ($\mathbf{p}_1$ and $\mathbf{p}_2$) are in opposition to the points on the convex chain $(\mathbf{p}_3, \ldots, \mathbf{p}_{m-1})$. Furthermore, $(\mathbf{p}_1, \mathbf{p}_2)$ should be nearly parallel to $(\mathbf{p}_3, \mathbf{p}_{m-1})$ and the bending of the convex chain should be small. By construction all vertices will become complete ver-

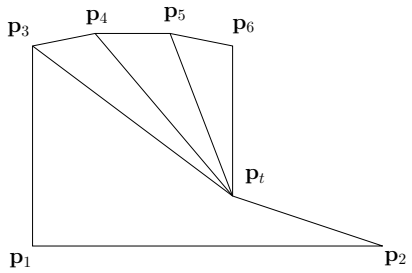


Figure 5.5: The elevator used for the height adjustment of $\mathbf{p}_t$.

tices, except $\mathbf{p}_t$. This can be achieved by setting $h_t = \max_{j \leq m}\{h_i\}$. A flipping sequence for the elevator is shown in Figure 5.6. We name the sequence from (a) to (d) *flipping to the right* and the mirrored sequence from (d) to (a) *flipping to the left*. Throughout the construction we will only consider flipping to the right and left. It depends on the height assignment of the vertices $\mathbf{p}_1, \ldots, \mathbf{p}_m$, if flipping to the right or to the left is an improving sequence.

Lemma 5.6. *For an elevator polygon let* h_i *be always* 0*, except for* $h_1 < 0$ *and* $h_2 < 0$*. Assume that the elevator polygon is pseudo-triangulated, and contains the pseudo-triangle spanned by* $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_i, \mathbf{p}_t$*, the triangle* $\mathbf{p}_{i-1}, \mathbf{p}_i, \mathbf{p}_1$ *and the triangle* $\mathbf{p}_i, \mathbf{p}_{i+1}, \mathbf{p}_t$ *(for some* i*). For any* $\delta > 1$ *we can embed the elevator polygon such that:*

A.) If $h_2 < \delta h_1$*, then the edge* $(1, i)$ *is concave.*

B.) If $h_1 < \delta h_2$*, then the edge* (t, i) *is concave.*

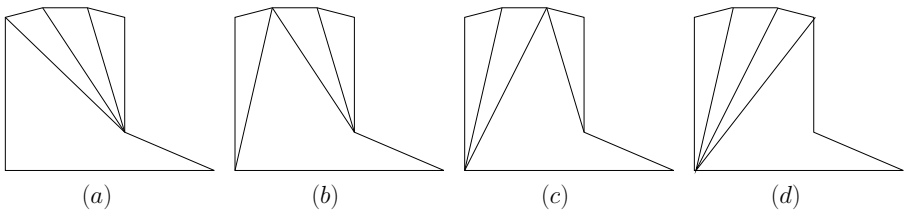


Figure 5.6: Flipping in elevator from Figure 5.5.

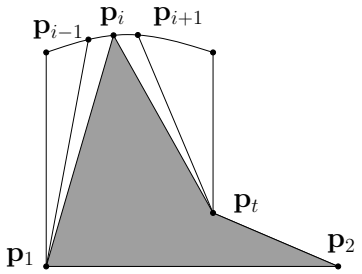


Figure 5.7: An illustration of the conditions of Lemma 5.6.

Proof. Let us first work out the conditions of the lemma. We have the elevator polygon with a pseudo-triangulation that appears in the flipping sequence left to right, or right to left. The particular situation is shown in Figure 5.7. We show that if condition A holds, then $(1, i)$ is concave. On the other hand condition B implies that (t, i) is concave.

Let us prove the Lemma for *some* value $\delta > 1$. We discuss later how the value of δ can be made arbitrarily close to 1. We start with the proof of statement A. The edge $(1, i)$ is concave if $\mathbf{p}_2$ lies below the plane spanned by $\mathbf{p}_i, \mathbf{p}_{i-1}$ and $\mathbf{p}_1$. This is true, if the signed volume of the tetrahedron[2] spanned by $\mathbf{p}_i, \mathbf{p}_{i-1}, \mathbf{p}_2$ and $\mathbf{p}_1$ is positive. We use the square bracket notation to denote the signed area of a triangle. The signed volume of a tetrahedron equals (we ignore the factor 1/6):

$$h_i[i-1, 2, 1] - h_{i-1}[i, 2, 1] + h_2[i, 1, i-1] - h_1[i, 2, i-1].$$

Since $h_i = h_{i-1} = 0$, the signed volume of the tetrahedron can be simplified to

$$-h_1[i, 2, i-1] + h_2[i, 1, i-1].$$

Both triangle areas are negative, because the vertices are in counter clockwise orientation. Thus, $(1, i)$ is concave if

$$h_2 < h_1 \frac{[i, 2, i-1]}{[i, 1, i-1]}. \tag{5.2}$$

We choose

$$\delta := \max_{3<j<m} \left\{ \frac{[j-1, j, 2]}{[j-1, j, 1]}, \frac{[j, j-1, 1]}{[j, j-1, 2]} \right\}.$$

(After the discussion of the second part of the lemma it will be clear why we select δ in this particular way.) The assumption of statement A leads to

$$h_2 < \delta h_1 \leq h_1 \frac{[i, 2, i-1]}{[i, 1, i-1]}.$$

Thus (5.2) holds and therefore the edge $(1, i)$ is concave.

[2] see page 50 for a definition.

We continue with the proof of statement B. The edge (t, i) is concave, if the signed volume of the tetrahedron spanned by $\mathbf{p}_i, \mathbf{p}_2, \mathbf{p}_1, \mathbf{p}_{i+1}$ is positive. The signed volume equals

$$-h_2[i, 1, i+1] + h_1[i, 2, i+1].$$

Both triangle areas used in the last equation are negative. Thus (t, i) is concave if

$$h_1 < h_2 \frac{[i, i+1, 1]}{[i, i+1, 2]}. \tag{5.3}$$

Assuming $h_1 < \delta h_2$ implies (5.3) and therefore we have that in this case the edge (t, i) is concave.

The value δ depends on the embedding of the elevator polygon. δ is the maximum over a set of quotients of triangle areas. These quotients have the form $[j-1, j, 2]/[j-1, j, 1]$ and $[j, j-1, 1]/[j, j-1, 2]$ for $3 < j < m$. We assumed that the embedding of the elevator polygon realizes the convex chain between $\mathbf{p}_3$ and $\mathbf{p}_{m-1}$ such that every segment on the chain is *nearly parallel* to the segment $(\mathbf{p}_1, \mathbf{p}_2)$. This assures that the triangle areas $[j, j-1, 2]$ and $[j, j-1, 1]$ are nearly the same (the base of both triangles is $(\mathbf{p}_j, \mathbf{p}_{j-1})$ and the heights are approximately the same due the assumption of parallelism). Thus all quotients (and therefore δ) are close to 1. □

For the later construction the following observation is helpful. An affine transformation $A\mathbf{p} + t$ changes triangles areas by a multiplicative constant of $\det(A)$. In the definition of δ this factor cancels.

Observation 5.7. *We consider an embedding of the elevator polygon that guarantees statement A and B in Lemma 5.6 for some value δ'. Any affine transformation of the polygon results in an embedding of the elevator polygon for which statement A and B in Lemma 5.6 hold with $\delta = \delta'$.*

Now we are able to construct an improving sequence for the elevator. Let us assume that we set all heights to zero except $h_1 = -1$ and $h_2 = -2 - \delta$. In this case flipping to the left yields an improving sequence. The value of δ is a constant, which depends on the embedding of the elevator only. We observe further that the height of the incomplete vertex $\mathbf{p}_t$ decreases after every flip.

Flipping to the left or right takes $m - 3$ flips if the elevator has m vertices. If we lower the height of the vertices $\mathbf{p}_1$ and $\mathbf{p}_2$ alternately k times, it is possible to flip from left to right repeatedly. Let $h_1(0), h_1(1), h_1(2), \ldots, h_1(k)$ be the decreasing sequence of heights assigned to $\mathbf{p}_1$. Equivalently we name $h_2(0), h_2(1), h_2(2), \ldots, h_2(k)$ the sequence for $\mathbf{p}_2$. We use sequences for h_1 and h_2 that guarantee for any i

$$h_1(i) > h_2(i) > h_1(i+1) > h_2(i+1).$$

Assume that the elevator polygon is embedded such that its value δ used in Lemma 5.6 is less that $h_2(i)/h_1(i)$ and $h_1(i+1)/h_2(i)$ (for all $i \leq k$). In the beginning the height of $h_2(0)$ is smaller than $h_1(0)$. This enables an improving sequence to the left. Now we reduce the height of $\mathbf{p}_1$ to $h_1(1)$. This activates

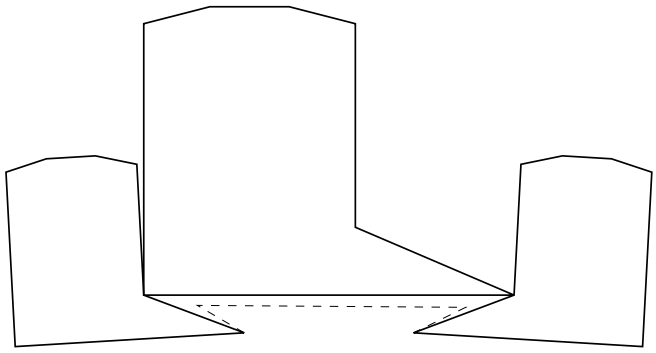


Figure 5.8: Assembling three elevators to obtain a quadratics improving flip sequence.

an improving sequence from left to right. We can repeat this procedure $2k+1$ times without violating the conditions of Lemma 5.6. In the end we have an improving sequence of $(m-3)(2k+1)$ flips.

The adjustment of the heights for $\mathbf{p}_1$ and $\mathbf{p}_2$ can be realized by two new elevators (see Figure 5.8). Notice that we have to add additional edges (for example the dashed edges in the Figure) to make the resulting polygon connected. The construction yields an improving flip sequence of length $2m^2+3m$. We choose $m = \lfloor n/3 \rfloor$ and obtain an improving sequence of length $\Theta(n^2)$. It is necessary to scale the heights of the outer elevators, such that the sequence induced by $\mathbf{p}_t$ constructs a sequence of lowering heights with an appropriate "jump" size.

We observe that inner-most elevator lowers the height of its vertex $\mathbf{p}_t$ $\Theta(m^2)$ times. Therefore, we can use two copies of this enhanced elevator and connect them to a new elevator. In the new elevator we are able to flip $\Theta(m^2)$ times form left to right and back. Choosing $m = \lfloor n/7 \rfloor$ results in a cubic improving sequence. Notice that it might be necessary to scale and shift the heights of the outer-most elevators further.

Most generally, we can connect r elevators sequentially in a "row". The numbers of necessary elevators grows exponentially. Therefore, we have only $m = \lfloor n/(2^r-1) \rfloor$ vertices for each elevator. The resulting flip sequence consists of $\Theta(n^r/(2^r-1)^r)$ flips. We choose $r = \lfloor \log \sqrt{n} \rfloor$ to obtain an improving sequence of length $n^{\Theta(\log n)}$. We omit the details how to embed the elevator polygons. Observation 5.7 enables us to shrink and turn the elevators. Thus, we are able to assemble them as described in the previous discussion. The appropriate scaling of the elevator heights works as follows: First we consider the inner-most elevator. We have to adjust the heights of the elevator of the second row, such that the step size of their vertices $\mathbf{p}_t$ enables repeatedly the flipping sequences for the inner-most elevator. In this way (from inside to the outside) we continue the scaling. In particular the next scaling process adjusts the heights of the elevators in the third row, then the fourth row, and so on. It can be observed that there is no cyclic dependence between these heights.

In contrast to our result we know that there exists (for a vertex empty

domain) also a short improving flip sequence of length $O(n^2)$ [5]. This sequenze can be found efficiently.

5.3 Representing Pseudo-Triangulations as Vertices of Polytopes

Representing geometric or combinatorial objects as vertices of high dimensional polytopes is a common technique to investigate these objects. We are interested in finding optimal pseudo-triangulations with the help of linear programs. In this section a polytope that represents pointed pseudo-triangulations of point sets and a polytope that represents regular triangulations. In the remaining part of the chapter we use both polytopes frequently to investigate pointed pseudo-triangulations.

5.3.1 The PPT-Polytope

The PPT-polytope, which was introduced in [75], represents all pointed pseudo-triangulations of a point set as its vertices. The polytope is a natural extension of the well-known associahedron [54]. The construction of the polytope is based on motions induced by pointed pseudo-triangulations. A one degree of freedom motion occurs if a convex hull edge is removed from a ppt. Up to linear transformations this motion is unique. During the motion all edge lengths are preserved, whereas all other pointwise distances increase. Motions with such a property are called *expansive*. Expansive motions induced by pointed pseudo-triangulations can be used to straighten polygonal arcs in 2d [84]. The set of expansive motions defined by a point set defines a high dimensional polyhedral cone. Applying involved perturbation techniques deforms this cone to a new polyhedron. The vertices of the polyhedron correspond to the pointed pseudo-triangulations that can be realized on the point set. The rays of the polyhedron represent the expansive motions. The polyhedron can be restricted to a face that contains all the vertices of the polyhedron but none of its rays. The constructed polytope is called *PPT-polytope*.

We continue with the formal description of the PPT-polytope. We assume that P is in general position (no three points on a line). Let $conv(P)$ denote the convex hull of the point set P. The PPT-polytope represents all pointed pseudo-triangulations of the domain $conv(P)$ as its vertices. We use the notation $[i, j, k]$ introduced in Chapter 2 as abbreviation for the signed area of the triangle spanned by $\mathbf{p}_i, \mathbf{p}_j$ and $\mathbf{p}_k$. If a zero appears in the square brackets, we refer to the point $(0, 0)^T := \mathbf{0}$, for example the expression $[0, i, j]$ denotes the signed area of the triangle given be $\mathbf{0}, \mathbf{p}_i$ and $\mathbf{p}_j$.

For every pair of points in P let f_{ij} be the *perturbation parameter* of this pair. Throughout the thesis we use as perturbation parameters

$$f_{ij} := [i, j, 0]^2.$$

Notice that there are other valid choices for the f_{ij} values that give the same (combinatorial) polytope. For example for any point $\mathbf{q} \notin P$ the parameter set

$f'_{ij} = [\mathbf{q}, \mathbf{p}_i, \mathbf{p}_j]^2$ is valid as well. Moreover, it is possible to obtain a formulation of the PPT-polytope without perturbation parameters [75].

We define the PPT-polytope by the following set of inequalities in the unknowns $\mathbf{v}_1, \ldots, \mathbf{v}_n$:

$$\begin{array}{lcll} \langle \mathbf{v}_i - \mathbf{v}_j, \mathbf{p}_i - \mathbf{p}_j \rangle & \geq & f_{ij} & \forall i, j \leq n, \\ \langle \mathbf{v}_i - \mathbf{v}_j, \mathbf{p}_i - \mathbf{p}_j \rangle & = & f_{ij} & \text{if } (i,j) \text{ convex hull edge}, \end{array} \tag{5.4}$$

plus three necessary normalizing conditions, which can be formulated as

$$\sum_{i=1}^{n} \mathbf{v}_i = \mathbf{0}, \tag{5.5}$$

$$\sum_{i=1}^{n} \langle \mathbf{p}_i^{\perp}, \mathbf{v}_i \rangle = 0. \tag{5.6}$$

The two dimensional vectors $\mathbf{v}_i$ correspond to (infinitesimal) velocities induced by the expansive motion. The normalizing conditions pin down the (trivial) rigid motions of the expansion. Other variants to "normalize" the PPT-polytope are possible as long as they factor out the translations and rotations of $(\mathbf{v}_i)_{i \leq n}$. The PPT-polytope is defined in the space $\mathbb{R}^{2n}$. As observed in [75] it has dimension $2n - 3$. Hence, a vertex of the polytope is given by a set of $2n - 3$ tight inequalities. The tight inequalities define the set of edges of the ppt that is associated with the vertex of the polytope. If two vertices of the PPT-polytope are adjacent, their induced pointed pseudo-triangulations differ only by an edge. Therefore, an edge in the PPT-polytope represents a flip.

Related to the PPT-polytope is the *polytope of non-crossing graphs* [63], which extends the PPT-polytope. The polytope represents all pseudo-triangulations of a point set as its vertices. Furthermore, it can deal with points in non-general position, whereas the PPT-polytope assumes that no three points of P are on a line.

5.3.2 The Secondary Polytope

The secondary polytope represents all regular triangulations of a point set as vertices. It can be defined as follows (see also [18, 39]): Let Δ be a triangulation of P. We denote by $vol_{\Delta,i}$ the sum of the areas of all triangles in Δ containing $\mathbf{p}_i$. The vector ϕ_Δ is defined as $(vol_{\Delta,1}, vol_{\Delta,2}, \ldots, vol_{\Delta,n})^T$. We define as secondary polytope $\mathcal{SP}$

$$\mathcal{SP} := conv\{\phi_\Delta \mid \Delta \text{ is a triangulation of } P\}.$$

The vertices of $\mathcal{SP}$ correspond to the regular triangulations of P.

If P is in convex position (the vertices numbered in cyclic order) $\mathcal{SP}$ is clearly equivalent to the PPT-polytope. This equivalence is not only combinatorial, but also affine, which is pointed out in [75]. In particular a map between the secondary polytope and the PPT-polytope is known. Let $d_{ij} := f_{ij} - \langle \mathbf{v}_i - \mathbf{v}_j, \mathbf{p}_i - \mathbf{p}_j \rangle$ be the slack variables of the constraints (5.4). As described

in [75] we can express a point $\mathbf{a} = (a_1, \ldots, a_n)$ in the secondary polytope as

$$a_i = -\frac{d_{i-1,i+1}}{[i-1, i, i+1]} + [i-1, i, i+1]. \tag{5.7}$$

Notice that the slack variables of type $d_{i-1,i+1}$ are sufficient to define a point in the secondary polytope for convex point sets.

We finally remark that regular pseudo-triangulations can be also represented as vertices of a polytope similar to the secondary polytope, the so-called polytope of regular pseudo-triangulations [5].

5.4 Regular Triangulations in the PPT-Polytope

For point sets in convex positions all triangulations are regular, and moreover pointed pseudo-triangulations. Regular triangulations are defined by their height assignment h. On the other hand we can identify a pointed pseudo-triangulation by its coordinates in the PPT-polytope. We study the problem how the height assignment h and the coordinates of the PPT-polytope are related. In particular, we address the problem how to identify a regular triangulation as solution of a linear program over the PPT-polytope. Our intention for an expression of $\mathcal{T}_h$ as LP formulation is primarily the better understanding of the PPT-polytope. The following observations work also as an easy scenario for more complicated settings. Let us remark, that there exist efficient methods to compute $\mathcal{T}_h$ by its height vector (this can be done for convex polygons in linear time [4]). Thus the algorithmic aspect for the computation of $\mathcal{T}_h$ as LP is less interesting. Our studies lead to two (independent) LP formulations for $\mathcal{T}_h$ which we present in Theorem 5.13 and Theorem 5.14.

We restrict ourselves in this section to convex point sets. Assume further that the convex point set $P = \{\mathbf{p}_1, \ldots, \mathbf{p}_n\}$ is labeled in cyclic order and that P is in general position. Furthermore, an index shift from i to $i-1$ or $i+1$ is always considered modulo n. Notice that not all lemmata in this section use the convexity assumption of the point set P. In particular, Lemma 5.9 with Corollary 5.10 hold also for non-convex point sets and are used in later sections in this general setting.

We know that $\mathcal{T}_h$ defines a piecewise linear convex function over the lifted point set. The corresponding concave function results in a different triangulation, which we name $\mathcal{T}_{-h}$. We address also the problem how to compute $\mathcal{T}_{-h}$. Once we have a linear program for $\mathcal{T}_h$ an LP formulation for $\mathcal{T}_{-h}$ needs only slight modifications.

For a convenient formalism we extend the square bracket notation for tetrahedra. We consider a set of 3-dimensional vertices that is given by the two dimensional point set P and some height assignment h. The term $[i, j, k, l]$ denotes the signed volume of the tetrahedron spanned by the four lifted vertices $\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k$, and $\mathbf{p}_l$. It is a well known fact that the signed volume can be

computed by[3]

$$[i,j,k,l] := \frac{1}{6}\det\begin{pmatrix} \mathbf{p}_i & \mathbf{p}_j & \mathbf{p}_k & \mathbf{p}_l \\ h_i & h_j & h_k & h_l \\ 1 & 1 & 1 & 1 \end{pmatrix}.$$

In the following we investigate linear optimization programs over the PPT-polytope. Most generally, the objective function is given by a set of n two-dimensional vectors $(\mathbf{d}_i)_{i\leq n}$:

$$\sum_{i=1}^{n}\langle \mathbf{v}_i, \mathbf{d}_i\rangle. \tag{5.8}$$

We restrict ourselves to objective functions whose parameter set $(\mathbf{d}_i)_{i\leq n}$ fulfills additional properties.

Definition 5.8 (moment-free objective function). *We say the objective function* $\sum_{i=1}^{n}\langle \mathbf{v}_i, \mathbf{d}_i\rangle$ *is* moment-free *(with respect to the set P) if and only if*

(1) $\sum_{i=1}^{n}\mathbf{d}_i = \mathbf{0}$,

(2) $\sum_{i=1}^{n}\langle \mathbf{d}_i, \mathbf{p}_i^{\perp}\rangle = 0$.

The intuition behind a moment-free objective function is the following: If we interpret the vectors $\mathbf{d}_i$ as forces acting on P a moment-free parameter set $(\mathbf{d}_i)_{i\leq n}$, produces neither a linear momentum (1) nor an angular momentum (2).

Lemma 5.9. *Let* $\sum_{i=1}^{n}\langle \mathbf{v}_i, \mathbf{d}_i\rangle$ *be a moment-free objective function over the PPT-polytope given by* (5.4)–(5.6)*. The solution of the induced linear program corresponds to a pseudo-triangulation of P with (non-equilibrium) stress u_{ij} on its edges. For every vertex $\mathbf{p}_i$ holds*

$$\sum_{i=1}^{n} u_{ij}(\mathbf{p}_i - \mathbf{p}_j) = \mathbf{d}_i.$$

The stress u_{ij} is non-negative on every interior edge.

Proof. We consider the linear program specified in the lemma as primal program. As dual program we obtain:

$$\begin{array}{rcll} \sum\limits_{0<i<j\leq n} f_{ij}u_{ij} & \to & \max, & \\ \sum_{j=1}^{n} u_{ij}(\mathbf{p}_i - \mathbf{p}_j) + \mathbf{t} + r\mathbf{p}_i^{\perp} & = & \mathbf{d}_i & 1\leq i\leq n, \\ u_{ij} & \geq & 0 & i<j,\ (i,j) \text{ interior edge.} \end{array}$$

Due to complementary slackness in linear programming duality we know that the non-tight inequalities refer to a dual variable which is zero. This means in our setting that every non-edge (i,j) in the primal solution induces $u_{ij}=0$ in

[3]Notice that usually a positive sign of the volume reflects the right-hand orientation of the four points. We use the left-hand system.

the solution of the dual program. Hence, the positive values u_{ij} are realized on (a subset of) the edges of the primal solution.

In the dual program appear variables that result from the normalizing conditions of the PPT-polytope. Condition (5.5) produces the two-dimensional variable $\mathbf{t}$ and condition (5.6) the variable r. We show that these variables vanish if the objective function is moment-free.

We have n restrictions of the form

$$\sum_{j=1}^{n} u_{ij}(\mathbf{p}_i - \mathbf{p}_j) + \mathbf{t} + r\mathbf{p}_i^{\perp} = \mathbf{d}_i. \tag{5.9}$$

Adding up all these equations cancels the u_{ij} values and gives

$$n\mathbf{t} + r\left(\sum_{i=1}^{n} \mathbf{p}_i\right)^{\perp} = \sum_{i=1}^{n} \mathbf{d}_i. \tag{5.10}$$

Because the objective function is moment-free the last equation equals zero. We take the scalar product of both sides of equation (5.9) with $\mathbf{p}_i^{\perp}$. This gives n equations of the form

$$-\sum_{j=1}^{n} u_{ij}\langle \mathbf{p}_i^{\perp}, \mathbf{p}_j\rangle + \langle \mathbf{p}_i^{\perp}, \mathbf{t}\rangle + r\|\mathbf{p}_i\|^2 = \langle \mathbf{p}_i^{\perp}, \mathbf{d}_i\rangle. \tag{5.11}$$

We sum up all these equations and obtain

$$\left\langle \left(\sum_{i=1}^{n} \mathbf{p}_i\right)^{\perp}, \mathbf{t} \right\rangle + r\sum_{i=1}^{n} \|\mathbf{p}_i\|^2 = \sum_{i=1}^{n} \langle \mathbf{p}_i^{\perp}, \mathbf{d}_i\rangle. \tag{5.12}$$

Again this equation is zero because the objective function is moment-free.

The variables $\mathbf{t}$ and r can be computed by solving a homogeneous linear equation system with three equations (5.10) and (5.12). It remains to show that $\mathbf{t} = \mathbf{0}$ and $r = 0$ is the only solution of this system. We can rephrase (5.10) to express $\mathbf{t}$ as

$$\mathbf{t} = -\frac{r}{n}\left(\sum_{i=1}^{n} \mathbf{p}_i\right)^{\perp}.$$

If $r = 0$ then $\mathbf{t} = \mathbf{0}$ and we get the trivial solution. Therefore, let us assume that r is nonzero. We plug this result into equation (5.12) and obtain

$$-\frac{r}{n}\left\langle \left(\sum_{i=1}^{n} \mathbf{p}_i\right)^{\perp}, \left(\sum_{i=1}^{n} \mathbf{p}_i\right)^{\perp} \right\rangle + r\sum_{i=1}^{n} \|\mathbf{p}_i\|^2 = 0.$$

Further simplifications give

$$\sum_{i=1}^{n} \|\mathbf{p}_i\|^2 - \frac{1}{n}\left\|\sum_{i=1}^{n} \mathbf{p}_i\right\|^2 = 0. \tag{5.13}$$

Let $\bar{\mathbf{p}} := \frac{1}{n}\sum_{i=1}^{n} \mathbf{p}_i$ denote the center of gravity of P. We deduce

$$\begin{aligned} \|\mathbf{p}_i\|^2 &= \|\bar{\mathbf{p}} + \mathbf{p}_i - \bar{\mathbf{p}}\|^2 \\ &= \|\bar{\mathbf{p}}\|^2 + 2\langle \bar{\mathbf{p}}, \mathbf{p}_i - \bar{\mathbf{p}}\rangle + \|\mathbf{p}_i - \bar{\mathbf{p}}\|^2. \end{aligned}$$

Plugging this equivalence into equation (5.13) leads to

$$\begin{aligned} \sum_{i=1}^{n}(\|\bar{\mathbf{p}}\|^2 + 2\langle \bar{\mathbf{p}}, \mathbf{p}_i - \bar{\mathbf{p}}\rangle + \|\mathbf{p}_i - \bar{\mathbf{p}}\|^2) - n\|\bar{\mathbf{p}}\|^2 &= n\|\bar{\mathbf{p}}\|^2 + \sum_{i=1}^{n} \|\mathbf{p}_i - \bar{\mathbf{p}}\|^2 - n\|\bar{\mathbf{p}}\|^2 \\ &= \sum_{i=1}^{n} \|\mathbf{p}_i - \bar{\mathbf{p}}\|^2 \end{aligned}$$

We observe that the last expression is strictly positive as long as not all $\mathbf{p}_i$ are the same, which is not allowed in our case. Hence, there is no other solution of the homogeneous system and the lemma follows. □

We can also show a more general result of the last lemma. Let us assume we have a set of edges (which we call constraints) that have to appear in the pseudo-triangulation. Any set of non-crossing pointed edges can be completed to a pointed pseudo-triangulation [84]. Based on the PPT-polytope we can define a polytope that contains all pseudo-triangulations with these constraints by making the corresponding inequalities

$$\langle \mathbf{v}_i - \mathbf{v}_j, \mathbf{p}_i - \mathbf{p}_j \rangle \geq f_{ij}$$

of the PPT-polytope tight. In particular, the modified polytope is a face of the PPT-polytope.

Corollary 5.10. *Let $\sum_{i=1}^{n}\langle \mathbf{v}_i, \mathbf{d}_i\rangle$ be a moment-free objective function over a face of the PPT-polytope given by* (5.4)–(5.6) *and some edge constraints. The solution of the induced linear program corresponds to a pseudo-triangulation of P with (non-equilibrium) stress u_{ij} on its edges. For every vertex $\mathbf{p}_i$ holds*

$$\sum_{i=1}^{n} u_{ij}(\mathbf{p}_i - \mathbf{p}_j) = \mathbf{d}_i.$$

The stress u_{ij} is non-negative on every interior edge that is not part of the constraints.

Proof. We mimic the proof of Lemma 5.9 and study the dual of the linear program stated in the lemma. For any equation in the primal program we have no information about the sign of the corresponding dual variable. The constraints behave like boundary edges. All other edges are not affected by restricting the PPT-polytope to a face. Hence, the arguments of the proof of Lemma 5.9 hold and the lemma follows. □

The observations made so far have not used the convexity of P. Thus everything we did so far holds for any point set in general position. We now come back to the computation of $\mathcal{T}_h$ and need the convexity of P from now on.

We introduce a geometric construction that assures that the heights of the regular triangulation are exactly h_i. To determine the heights we "span" all points in P by a pyramid-shaped shell. We assume for now that none of the $\mathbf{p}_i$ lies in the origin. This assumption is valid since translating the point set does not affect the lifting.

Lemma 5.11. *Given a convex polygon (P, E) with height assignment h for its vertices. Let $\mathbf{p}_0 \neq \mathbf{p}_i$ be an additional point in the origin. Consider the geometric graph $G = (P \cup \{\mathbf{p}_0\}, E \cup \{(0,i) | i \leq n\})$. The lifting of G to $\{(x_i, y_i, h_i) | i \leq n\} \cup \{(0,0,0)^T\}$ corresponds to the stresses*

$$\omega_{0i} := \frac{[0, i-1, i, i+1]}{[0, i-1, i][0, i, i+1]}.$$

Proof. Assume that no y-coordinate is zero (otherwise shift the point set). The stress of an edge $(0,i)$ can be computed by the gradients of the two faces that share this edge. The explicit formula is given by (2.5), in our case we have

$$\mathbf{a}_l - \mathbf{a}_r = \omega_{0i} \mathbf{p}_i^{\perp},$$

where $\mathbf{a}_l$ denotes the gradient of the triangle f_l left of $(0,i)$ and $\mathbf{a}_r$ the gradient of the triangle f_r right of $(0,i)$. The situation is depicted in Figure 5.9. We

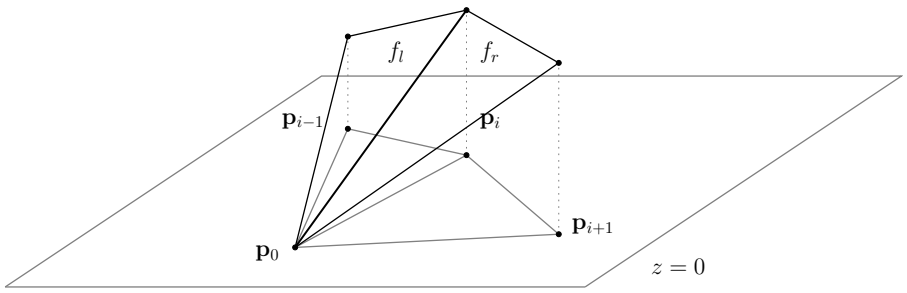


Figure 5.9: The local situation responsible for the stress ω_{0i}.

can determine the gradients $\mathbf{a}_l$ and $\mathbf{a}_r$ by the solution of the following linear equation system

$$\begin{aligned} \langle \mathbf{p}_{i-1}, \mathbf{a}_l \rangle &= h_{i-1}, \\ \langle \mathbf{p}_i, \mathbf{a}_l \rangle &= h_i, \\ \langle \mathbf{p}_i, \mathbf{a}_r \rangle &= h_i, \\ \langle \mathbf{p}_{i+1}, \mathbf{a}_r \rangle &= h_{i+1}. \end{aligned}$$

We observe that it suffices to calculate the x-coordinate a_l^x, a_r^x of the gradients. We obtain

$$\begin{aligned} a_l^x &= \frac{h_{i-1} y_i - h_i y_{i-1}}{[0, i-1, i]}, \\ a_r^x &= \frac{h_i y_{i+1} - h_{i+1} y_i}{[0, i, i+1]}. \end{aligned}$$

The stress ω_{0i} equals $(a_r^x - a_l^x)/y_i$. We substitute a_l^x and a_r^x and obtain

$$\omega_{0i} = \frac{h_i y_{i+1}}{y_i[0,i,i+1]} - \frac{h_{i+1}}{[0,i,i+1]} - \frac{h_{i-1}}{[0,i-1,i]} + \frac{h_i y_{i-1}}{y_i[0,i-1,i]}$$

The following identity helps to simplify the expression for ω_{0i} further:

$$\frac{y_{i-1}}{y_i}[0,i,i+1] + \frac{y_{i+1}}{y_i}[0,i-1,i] = [0,i-1,i+1].$$

We end with the expression for the stress ω_{0i}:

$$\omega_{0i} = \frac{[0,i-1,i,i+1]}{[0,i-1,i][0,i,i+1]}.$$

Executing the same operations with the y-coordinates of the gradients leads to the same stress for ω_{0i}. □

The placement of the point $\mathbf{p}_0$ is not restricted to the origin. One can show that every choice for $\mathbf{p}_0$ is valid. However, assuming $\mathbf{p}_0 = \mathbf{0}$ simplifies the computations. The pyramid-shaped shell gives us an appropriate lifting for set P. The lifting is only in equilibrium at the point $\mathbf{p}_0$. At any point of P the forces are not resolving. In particular, we have the resulting force $\omega_{0i}\mathbf{p}_i$ at each $\mathbf{p}_i$. We use these vectors as parameters $-\mathbf{d}_i$ of our objective function.

Lemma 5.12. *Let ω_{0i} be defined as in Lemma 5.11. The objective function* (5.8) *defined by*

$$\mathbf{d}_i \equiv -\omega_{0i}\mathbf{p}_i$$

is moment-free.

Proof. The stresses ω_{0i} are constructed such that the additional vertex $\mathbf{p}_0$ is in equilibrium. Therefore,

$$\sum_{i=1}^{n} \mathbf{d}_i = -\sum_{i=1}^{n} \omega_{0i}\mathbf{p}_i = \sum_{i=1}^{n} \omega_{0i}(\mathbf{p}_0 - \mathbf{p}_i) = \mathbf{0}.$$

The angular momentum $\sum_{i=1}^{n} \langle \mathbf{p}_i\omega_{0i}, \mathbf{p}_i^{\perp}\rangle$ equals zero as well. □

The main result of this section is stated in the following theorem.

Theorem 5.13. *Given a convex point set P (no point in the origin) and a height assignment h for the vertices of P.*

(1) The minimization of

$$-\sum_{i=1}^{n} \langle \mathbf{v}_i, \mathbf{p}_i\rangle \frac{[0,i-1,i,i+1]}{[0,i-1,i][0,i,i+1]}$$

over the PPT-polytope given by (5.4)– (5.6) *gives the regular triangulation $\mathcal{T}_h$.*

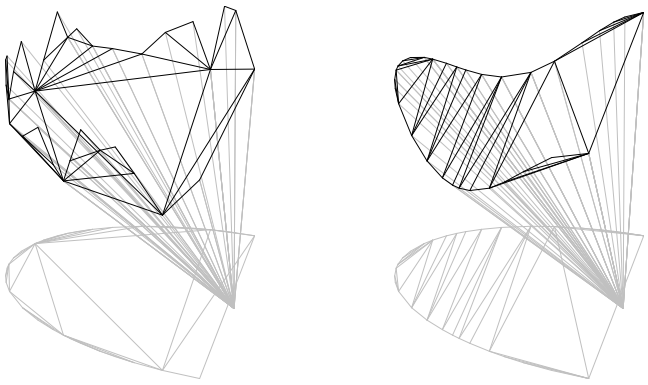
(2) The minimization of

$$\sum_{i=1}^{n} \langle \mathbf{v}_i, \mathbf{p}_i \rangle \frac{[0, i-1, i, i+1]}{[0, i-1, i][0, i, i+1]}$$

over the PPT-polytope given by (5.4)– (5.6) *gives the regular triangulation* $\mathcal{T}_{-h}$.

Proof. We first prove statement (1). Due to Lemma 5.12 the objective function mentioned in statement (1) is moment-free. Therefore (by Lemma 5.9) the solution of the linear program defined in (1) gives a non-negative stress u_{ij} with resulting forces $-[0, i-1, i, i+1]/([0, i-1, i][0, i, i+1])\mathbf{p}_i$ in every point $\mathbf{p}_i$. We can cancel these forces by adding the pyramid-shaped shell defined in Lemma 5.11. Since the graph given by the nonzero u_{ij} is outerplanar, the addition of the edges $(0, i)$ gives a planar graph. This graph is in equilibrium and therefore the induced lifting gives a spatial embedding by the Maxwell-Whiteley Theorem. We know that the "curvature" on the edges of the triangulation is positive and hence the solution of the linear program stated in (1) gives $\mathcal{T}_h$. The objective function mentioned in (2) differs only in the sign from (1). Hence it is moment-free and the solution gives a non-equilibrium stress. We multiply the stress with -1 such that the non-resolving forces match with the forces resulting from the pyramid-shaped shell. As consequence the curvature is now inverted and the solution of the linear program stated in (2) gives $\mathcal{T}_{-h}$. □

We present two examples of the construction of $\mathcal{T}_h$. Both examples use the point set P_{sc} that contains 32 points on the semicircle with center $\mathbf{0}$ and radius 5. In Figure 5.10(a) the heights were chosen uniformly at random on the interval $[15, 20]$. In Figure 5.10(b) they were computed by $h(\mathbf{p}_i) = 10 + 2\cos(y_i/2)$. Notice that the constructed lifted surface might have self-intersections. In particular, the pyramid-shaped shell might intersect the lifted triangulated surface. However, this effect does not violate Theorem 5.13. See Figure 5.11



(a) (b)

Figure 5.10: Two examples for the computation of $\mathcal{T}_h$ by an LP over the PPT-polytope. The pictures are scaled to obtain a nice illustration.

(scaled down) for a self-intersecting surface given by P_{sc} with height assignment $h(\mathbf{p}_i) = 1.5^{y_i}$. Figure 5.12 shows $\mathcal{T}_{-h}$ for the set P_{sc} with the height

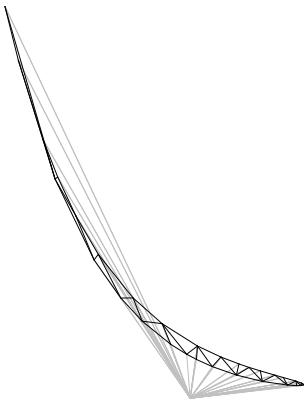


Figure 5.11: A lifted surface that intersects the pyramid-shaped shell.

assignment used for Figure 5.10(b).

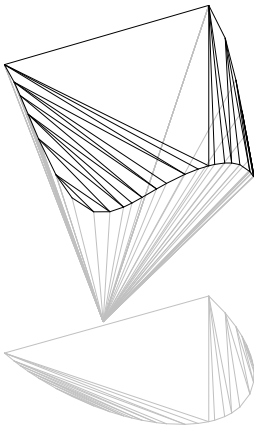


Figure 5.12: An example of the upper convex hull computed as LP over the PPT-polytope.

We continue with a different method for expressing $\mathcal{T}_h$ and $\mathcal{T}_{-h}$ as solutions of linear programs over the PPT-polytope. This alternative approach make use of the affine mapping between the secondary polytope and the PPT-polytope.

Every triangulation covers some volume between the corresponding lifted surface and the plane triangulation. The triangulation $\mathcal{T}_h$ minimizes this (signed) volume. Thus, minimizing $\sum_{i=1}^{n} a_i h_i$ over the secondary polytope gives $\mathcal{T}_h$. The corresponding maximization problem computes $\mathcal{T}_{-h}$. We use the affine map (5.7) to obtain an LP over the PPT-polytope. The objective function

$\sum_{i=1}^{n} a_i h_i$ changes to

$$-\sum_{i=1}^{n} h_i \left(\frac{d_{i-1,i+1}}{[i-1,i,i+1]} + [i-1,i,i+1] \right). \tag{5.14}$$

We can ignore the constant additive term $[i-1,i,i+1]$ for the optimization problem. Remember that the variables d_{ij} are the slack variables in the description of PPT-polytope given by (5.4): $d_{i-1,i+1} = f_{i-1,i+1} - \langle \mathbf{p}_{i-1} - \mathbf{p}_{i+1}, \mathbf{v}_{i-1} - \mathbf{v}_{i+1} \rangle$. Plugging this expression into the objective function (5.14) leads to

$$\sum_{i=1}^{n} h_i \left(-\frac{f_{i-1,i+1}}{[i-1,i,i+1]} + \frac{\langle \mathbf{p}_{i-1} - \mathbf{p}_{i+1}, \mathbf{v}_{i-1} - \mathbf{v}_{i+1} \rangle}{[i-1,i,i+1]} \right).$$

Since the term $h_i f_{i-1,i+1}/[i-1,i,i+1]$ is a constant we simplify the last expression further to

$$\sum_{i=1}^{n} h_i \frac{\langle \mathbf{p}_{i-1} - \mathbf{p}_{i+1}, \mathbf{v}_{i-1} - \mathbf{v}_{i+1} \rangle}{[i-1,i,i+1]}.$$

We now put the last equation in a form similar to (5.8).

Theorem 5.14. *For a point set P in convex position the minimization of the linear function*

$$\sum_{i=1}^{n} \left\langle \mathbf{v}_i, \frac{h_{i+1}(\mathbf{p}_i - \mathbf{p}_{i+2})}{[i,i+1,i+2]} + \frac{h_{i-1}(\mathbf{p}_i - \mathbf{p}_{i-2})}{[i-2,i-1,i]} \right\rangle \tag{5.15}$$

over the PPT-polytope given by (5.4) – (5.6) *leads to a solution that corresponds to the triangulation $\mathcal{T}_h$.*

Corollary 5.15. *Maximizing* (5.15) *over the PPT-polytope given by* (5.4) – (5.6) *leads to a solution that corresponds to the triangulation $\mathcal{T}_{-h}$.*

We interpret the linear program stated in Theorem 5.14 geometrically. We make again use of Lemma 5.9.

Proposition 5.16. *The objective function* (5.15) *is moment-free.*

Proof. We have

$$\mathbf{d}_i = \frac{h_{i+1}(\mathbf{p}_i - \mathbf{p}_{i+2})}{[i,i+1,i+2]} + \frac{h_{i-1}(\mathbf{p}_i - \mathbf{p}_{i-2})}{[i-2,i-1,i]}.$$

We reorder the summands of $\sum_{i=1}^{n} \mathbf{d}_i$ by $\mathbf{p}_i$ and obtain

$$\begin{aligned} \sum_{i=1}^{n} \mathbf{d}_i &= \sum_{i=1}^{n} \mathbf{p}_i \left(\frac{h_{i+1}}{[i,i+1,i+2]} + \frac{h_{i-1}}{[i-2,i-1,i]} - \right. \\ &\quad \left. \frac{h_{i+1}}{[i,i+1,i+2]} - \frac{h_{i-1}}{[i-2,i-1,i]} \right) \\ &= \mathbf{0}. \end{aligned}$$

Now we use the identity $\langle \mathbf{p}_j, \mathbf{p}_i^\perp \rangle = -\langle \mathbf{p}_i, \mathbf{p}_j^\perp \rangle$ to evaluate the angular momentum:

$$\begin{aligned}\sum_{i=1}^{n} \langle \mathbf{d}_i, \mathbf{p}_i^\perp \rangle &= \sum_{i=1}^{n} \left(-\frac{\langle \mathbf{p}_{i+2}, \mathbf{p}_i^\perp \rangle}{[i, i+1, i+2]} - \frac{\langle \mathbf{p}_{i-2}, \mathbf{p}_i^\perp \rangle}{[i-2, i-1, i]} \right) \\ &= \sum_{i=1}^{n} \left(-\frac{\langle \mathbf{p}_{i+2}, \mathbf{p}_i^\perp \rangle}{[i, i+1, i+2]} - \frac{\langle \mathbf{p}_i, \mathbf{p}_{i+2}^\perp \rangle}{[i, i+1, i+2]} \right) \\ &= 0.\end{aligned}$$

□

Since the objective function (5.15) is moment-free we can apply Lemma 5.9 and obtain the following non-equilibrium stress condition for a stress u_{ij}.

$$\sum_{j=1}^{n} u_{ij}(\mathbf{p}_i - \mathbf{p}_j) = \frac{h_{i+1}(\mathbf{p}_i - \mathbf{p}_{i+2})}{[i, i+1, i+2]} + \frac{h_{i-1}(\mathbf{p}_i - \mathbf{p}_{i-2})}{[i-2, i-1, i]} \quad 1 \le i \le n.$$

Setting

$$\omega_{ij} := \begin{cases} u_{ij} - h_{i+1}/[i, i+1, i+2] & \text{if } i+2 = j, \\ u_{ij} & \text{otherwise,} \end{cases}$$

defines an equilibrium stress. If $u_{i,i+2}$ is nonzero we consider the stressed edge $(i, i+2)$ as two edges whose projections in the xy-plane coincide. Nevertheless, we treat both edges separately, one edge gets the stress $u_{i,i+2}$ and the other $h_{i+1}/[i, i+1, i+2]$. We can distinguish both edges by their different corresponding oriented patches.

The graph induced by the nonzero stresses is not planar. However, planarity is a necessary condition for lifting graphs with equilibrium stress. We apply Bow's trick, mentioned in [59] and also applied in [28], to planarize the graph. This is achieved by introducing additional vertices at the crossing of the edges $(i, i+2)$. The remaining edges can be drawn in the outer face without producing any crossing (see Figure 5.13). The old stresses are uniformly distributed on the subdivided edges. This means that adding a point $\mathbf{p}_k$ on an edge (i, j) leads to a stress

$$\omega_{ik} := \omega_{ij} \frac{\|\mathbf{p}_i - \mathbf{p}_k\|}{\|\mathbf{p}_i - \mathbf{p}_j\|},$$

$$\omega_{kj} := \omega_{ij} \frac{\|\mathbf{p}_k - \mathbf{p}_j\|}{\|\mathbf{p}_i - \mathbf{p}_j\|}.$$

It is easy to check that a stress defined in this way is an equilibrium stress on the planarized graph.

After the planarization we can apply the lifting technique of Chapter 2 to obtain a geometric interpretation of the program stated in Theorem 5.14. We construct the lifting by placing the convex polygon in the xy-plane and tilting its corners up. The tilted planes are bent along the diagonals $(i, i+2)$. The bends of different diagonals are superimposed. Since each bend lifts only a single corner the obtained heights are exactly h_i. Observe that we do not need

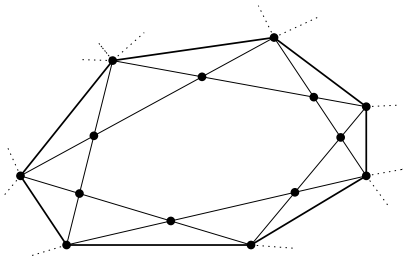


Figure 5.13: The planarization technique for graphs with equilibrium stress. For showing that the graph has a planar embedding the diagonals of the triangulations can be drawn in the outer face.

any stress of an edge of the triangulation for the computation of the heights. The lifted spatial embedding consists of a bowl-shaped shell in which the lifted triangulation is placed. The tilting of one plane is depicted in Figure 5.14. The convexity of the lifted triangulation follows by the (positive) sign of stresses u_{ij}.

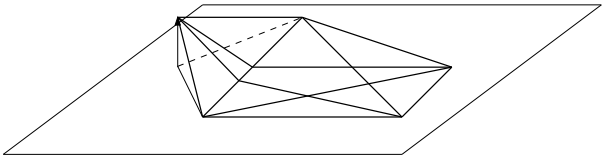


Figure 5.14: Constructing a lifted point set by tilting along diagonals.

Let us remark that it is interesting that two different objective functions in Theorem 5.13 and Theorem 5.14 over the PPT-polytope can be used to express $\mathcal{T}_h$. The two different LP formulations are based on different approaches. One approach relies on the pyramid-shaped shell and the other on a bowl-shaped shell that is used to define the heights used for the lifting. Both objective functions can be transformed into each other by adding a suitable linear combination of the equations used in the definition of the PPT-polytope. We leave out the explicit reformulation.

5.5 Load Resolving

The observations in the previous chapter have consequences for pseudo-triangulations from a rigidity-theoretic point of view. Let us first define:

Definition 5.17 (Load, Moment-free Load). *A set of two-dimensional vectors $(\mathbf{f}_i)_{i\leq n}$ is called a load. A load that fulfills $\sum_{i=1}^{n} \mathbf{f}_i = \mathbf{0}$ and $\sum_{i=1}^{n} \langle \mathbf{p}_i^{\perp}, \mathbf{f}_i \rangle$ for a point set $P = \{\mathbf{p}_1, \ldots, \mathbf{p}_n\}$ is called* moment-free *load for P.*

We know that pointed pseudo-triangulations are (in a combinatorial sense) minimal rigid graphs [84]. Minimal rigid graphs have the property that they

are for any generic point set infinitesimal rigid [27]. This implies that they can absorb any moment-free load $(\mathbf{f}_i)_{i\leq n}$ on P. In other words, there exists a unique set of stresses ω_{ij} on the edges of the minimal rigid graph such that

$$\forall \mathbf{p}_i \in P \quad \sum_j \omega_{ij}(\mathbf{p}_i - \mathbf{p}_j) = \mathbf{f}_i.$$

Notice that as a consequence of Newton's law a load that is not moment free cannot be resolved by any framework.

We have no control over the sign of the stresses that resolve the load. However, there are applications where it is desirable to allow only positive stresses on interior edges. We cannot guarantee that a pointed pseudo-triangulation can resolve a load with positive interior stresses only. But with help of Lemma 5.9 we can compute for a given load a pointed pseudo-triangulation with this property.

Theorem 5.18. *Given a moment-free load $(\mathbf{f}_i)_{i\leq n}$ for a point set P. There exists a pointed pseudo-triangulation on P that resolves the load $(\mathbf{f}_i)_{i\leq n}$ with non-negative interior stresses. The pseudo-triangulation can be computed by minimizing the objective function $\sum_{i=1}^{n}\langle \mathbf{v}_i, \mathbf{f}_i\rangle$ over the PPT-polytope given by* (5.4)–(5.6).

Proof. The existence of a pseudo-triangulation that resolves the load is a direct consequence of Lemma 5.9. □

For most of the loads there is a unique pointed pseudo-triangulation that cancels the load with positive stresses. To see this consider the normal fan of the PPT-polytope (the definition of normal fan can be found in [96]). Since the PPT-polytope is simple, every $2n-3$-dimensional cone in the fan refers to a set of vectors, whose optimization lead to a distinct pseudo-triangulation. Selecting an objective function from the interior of a $2n-3$-dimensional normal cone gives therefore a unique pseudo-triangulation. A random objective function lies with high probability in the interior of a $2n-3$-dimensional normal cone.

We can use appropriate loads to construct pseudo-triangulations such that one of it small angles points to a prescribed direction.

Definition 5.19 (Pointed to a direction). *We say a vertex $\mathbf{p}_i$ of a pointed pseudo-triangulation is pointed to a direction $\mathbf{d}_i \in \mathbb{R}^2$ if the vector $-\mathbf{d}_i$ with point of application $\mathbf{p}_i$ lies inside an angle smaller than π at $\mathbf{p}_i$.*

See Figure 5.15 for examples of points pointed to a direction. If every point $\mathbf{p}_i$ is pointed to a direction $\mathbf{d}_i$ we say the set $(\mathbf{p}_i)_{i\leq n}$ is pointed to the directions $(\mathbf{d}_i)_{i\leq n}$.

Assume we have for every interior vertex a prescribed direction $\mathbf{d}_i$ where it should point to. We can find a pointed pseudo-triangulation that fulfills all these pointedness conditions. This can be deduced by the following argument. Suppose we have a pointed pseudo-triangulation with positive interior stresses ω_{ij}. The force at $\mathbf{p}_i$ equals $\mathbf{f}_i := \sum_{i=1}^{n}\omega_{ij}(\mathbf{p}_i - \mathbf{p}_j)$. Therefore, $\mathbf{f}_i$ lies inside the reflection of the cone defined by the outermost edges incident to $\mathbf{p}_i$. The

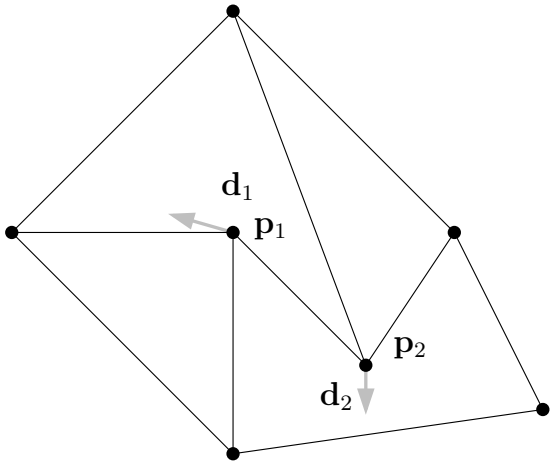


Figure 5.15: Vertex $\mathbf{p}_1$ is not pointed to direction $\mathbf{d}_1$ whereas vertex $\mathbf{p}_2$ is pointed to direction $\mathbf{d}_2$.

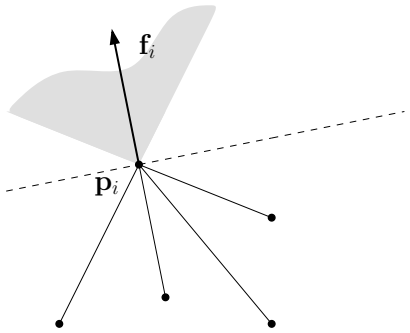


Figure 5.16: Possible force $\mathbf{f}_i$ induced by positive stresses at $\mathbf{p}_i$.

"feasible area" for $\mathbf{f}_i$ is depicted in Figure 5.16 as shaded cone. Clearly, the vector $-\mathbf{f}_i$ with point of application $\mathbf{p}_i$ lies inside a small angle at $\mathbf{p}_i$. Hence, $\mathbf{p}_i$ is pointed to direction $\mathbf{f}_i$. By the same construction it follows that in a pointed pseudo-triangulation that resolves the load $(\mathbf{d}_i)_{i\leq n}$ with positive interior stresses, each $\mathbf{p}_i$ is pointed to direction $\mathbf{d}_i$. We can find such a pointed pseudo-triangulation with help of Theorem 5.18. However we have to guarantee that the load is moment-free. This can be achieved by assigning appropriate vectors $\mathbf{d}_i$ to the boundary vertices. Assume that the convex hull vertices are $\mathbf{p}_1, \ldots, \mathbf{p}_k$. Let the sum of the interior load be $(d_x, d_y)^T$ and let $d_{rot} := \sum_{i=k+1}^{n} \langle \mathbf{d}_i, \mathbf{p}_i^{\perp} \rangle$. One possible way to construct a moment-free load is given by the following assignment (under the assumption that $y_1 \neq y_2$):

$$\mathbf{d}'_i := \begin{cases} ((d_{rot} - d_y x_1 + d_x y_2)/(y_1 - y_2), -d_y)^T & \text{if } i = 1, \\ ((d_{rot} - d_y x_1 + d_x y_1)/(y_2 - y_1), 0)^T & \text{if } i = 2, \\ \mathbf{0} & \text{if } 2 < i \leq k, \\ \mathbf{d}_i & \text{if } k < i \leq n. \end{cases} \tag{5.16}$$

If $y_1 = y_2$ we can find a similar assignment (in this case $x_1 \neq x_2$ holds). We conclude with

Theorem 5.20. *Let P be a point set in general position with convex hull vertices $\mathbf{p}_1, \ldots, \mathbf{p}_k$. We can construct a pointed pseudo-triangulation that is pointed to the directions $(\mathbf{d}_i)_{k<i\leq n}$ by minimizing*

$$\sum_{i=1}^{n} \langle \mathbf{d}'_i, \mathbf{p}_i \rangle$$

over the PPT-polytope given by (5.4)–(5.6)*, where $(\mathbf{d}'_i)_{i\leq n}$ is obtained from $(\mathbf{d}_i)_{k<i\leq n}$ by equation* (5.16)*.*

Let us look at an example for Theorem 5.20. We study the electric field given by two infinitely long line charges perpendicular to the xy-plane with opposite polarities. One line charge intersects the xy-plane at the point $(2, 0)^T$ the other one at $(-2, 0)^T$. The resulting gradient field in the xy plane is depicted in Figure 5.17. Our goal is to compute the pseudo-triangulation that is pointed

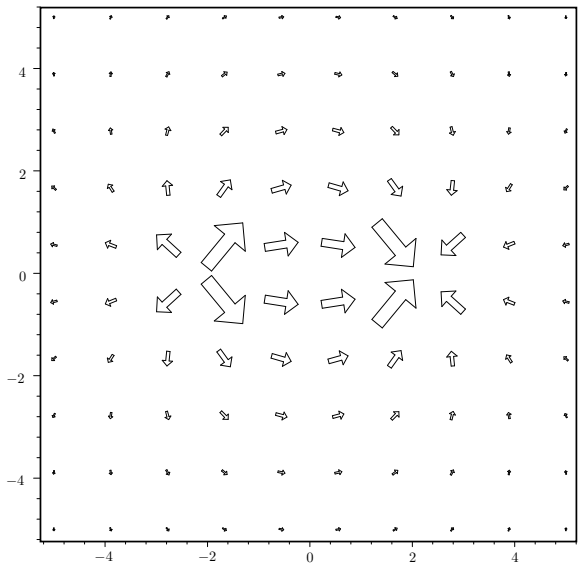


Figure 5.17: The gradient field used is in the example.

to the directions induced by the electric field. The point set P is given by 150 points, randomly picked from the interval $[-5, 5] \times [-5, 5]$. The solution of the linear program described in Theorem 5.20 gives us the desired pseudo-triangulation. It is shown in Figure 5.18. The tiny gray line segments indicate the (normalized) directions $\mathbf{d}_i$.

We finish this section with an application of load resolving. It is well known that not every triangulation is regular. The most prominent example is the projection of Schönhardt's polyhedron. This implies that there exist embeddings whose equilibrium stress is always negative on some edges. However, an

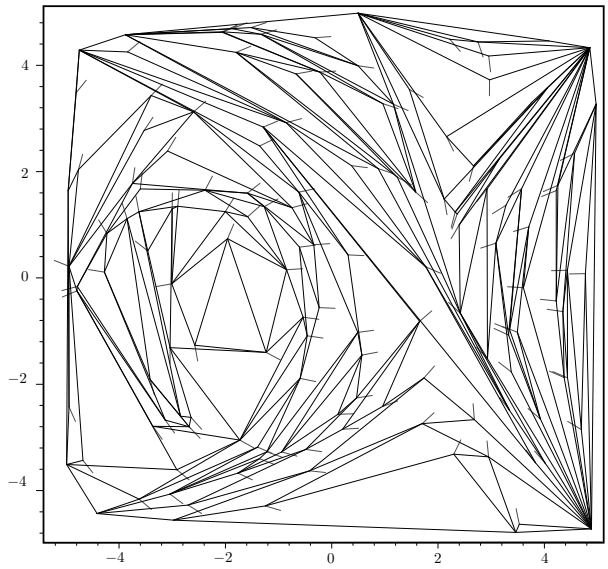


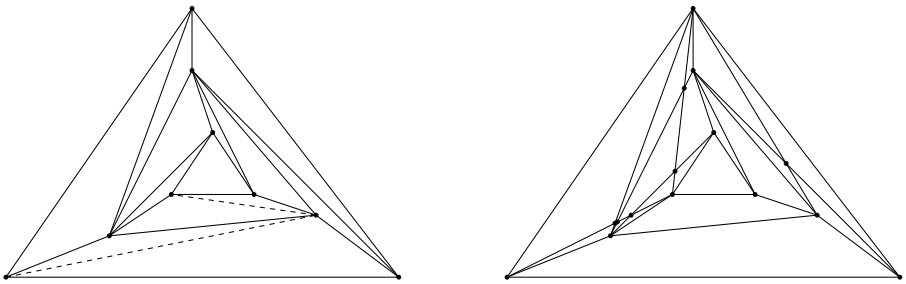
Figure 5.18: A pseudo-triangulation pointed to the directions given by the gradient vector field shown in Figure 5.17.

equilibrium stress that is positive on every interior edge is desirable for many applications.

Laplace operators can be used to model diffusion processes (and other physical dependencies). The discretized version of it (the discrete Laplace operator) is basically a stressed graph in equilibrium. The equilibrium models the fact that no energy/matter is lost when time progresses. Negative stresses are in this setting unnatural. They would model a reverse diffusion process, which has no equivalent in the real world. We refer to [89] for a discussion of desirable properties for discrete Laplace operators.

Imagine that we model a physical process that results in a realization of a non-regular triangulation $\mathcal{T}_{nr}$. Clearly we can take any regular triangulation on the points set of $\mathcal{T}_{nr}$ and use this mesh as model instead. This approach might change $\mathcal{T}_{nr}$ very much. An edge can appear between two points whose graph theoretic distance in $\mathcal{T}_{nr}$ is very large. In other words, the new triangulation behaves well but it does not have much in common with the original model.

We propose an approach to solve this problem that is based on load resolving. We explain it by example. Figure 5.19(a) shows the triangulation $\mathcal{T}_{nr}$. The embedding of $\mathcal{T}_{nr}$ is non-regular – 2 edges with negative stresses are unavoidable. A possible pair of edges with negative stress is drawn dashed in Figure 5.19(a). We begin to construct a new mesh by throwing out the edges with negative stress. The new embedding $\mathcal{T}_{-}$ is not in equilibrium anymore. On the other hand, the original stresses on the remaining edges cancel the load that is given by the removed edges. This load is moment-free and hence we can compute a pointed pseudo-triangulation that resolves it with positive interior stresses (Theorem 5.20). Figure 5.20 depicts this pseudo-triangulation. The



(a) (b)

Figure 5.19: A non regular-triangulation (a) and a regular triangulation on the same point set. Edges with negative edges are draw dashed.

direction of the load is indicated by gray arrows. The union of $\mathcal{T}_-$ with the pointed pseudo-triangulation gives an equilibrium stressed graph with positive interior stresses (Figure 5.19(b)). If we are interested in a planar embedding we have to apply Bow's trick [59] mentioned at page 59.

Our approach presents only a rough idea how to transform a non-regular into a regular triangulation. We have no estimates over the similarity between the regular and the non-regular triangulation. In general the interaction between regular and non-regular triangulation is not well understood and leaves place for further studies. We mention some related problems in Chapter 7.

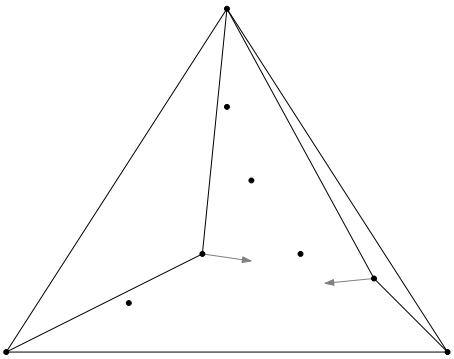


Figure 5.20: The pseudo-triangulation that cancels the load induced by the negative stresses of the triangulation in Figure 5.19(a). The direction of the load is indicated by the tiny gray arrows.

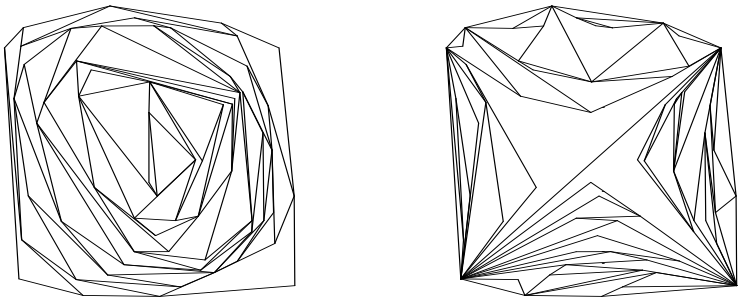
5.6 The Pseudo-Triangulation of the Canonical Objective Function

The canonical pointed pseudo-triangulation (short cppt) was introduced in [75] as the solution of a linear program that minimizes the canonical objective func-

tion

$$\sum_{i=1}^{n} \langle \mathbf{p}_i, \mathbf{v}_i \rangle \tag{5.17}$$

over the PPT-polytope given by the conditions (5.4) – (5.6). The pointed pseudo-triangulation that corresponds to the solution of the analogous maximization problem is called *max-cppt*. See Figure 5.21 for a cppt and max-cppt of a point set with 64 points.



cppt max-cppt

Figure 5.21: Examples of canonical pointed pseudo-triangulations.

Lemma 5.21. *If we remove condition* (5.6) *in the LP formulation of the canonical pointed pseudo-triangulation the solution corresponds still to the cppt. The analogous statement holds for the max-cppt.*

Proof. The condition (5.6) restricts the PPT-polytope to a hyperplane H_{rotate}. This hyperplane is given by the normal vector

$$\mathbf{n}_{rotate} = (\mathbf{p}_1^{\perp}, \ldots, \mathbf{p}_n^{\perp})^T.$$

All other hyper-(half-)planes of the PPT-polytope have a normal vector of the form

$$\mathbf{n}_{ij} = (\ldots, (\mathbf{p}_i - \mathbf{p}_j), \ldots, (\mathbf{p}_j - \mathbf{p}_i), \ldots)^T.$$

Due to the identity $\langle \mathbf{p}_j, \mathbf{p}_i^{\perp} \rangle = -\langle \mathbf{p}_i, \mathbf{p}_j^{\perp} \rangle$ we have

$$\begin{aligned} \langle \mathbf{n}_{rotate}, \mathbf{n}_{ij} \rangle &= \langle (\mathbf{p}_i - \mathbf{p}_j), \mathbf{p}_i^{\perp} \rangle + \langle (\mathbf{p}_j - \mathbf{p}_i), \mathbf{p}_j^{\perp} \rangle \\ &= \langle -\mathbf{p}_j, \mathbf{p}_i^{\perp} \rangle - \langle \mathbf{p}_i, \mathbf{p}_j^{\perp} \rangle \\ &= 0. \end{aligned}$$

Therefore, H_{rotate} is orthogonal to all hyperplanes given by (5.4). The PPT-polytope without condition (5.6) is a cylinder. The vector of the canonical objective function is contained in H_{rotate}. Thus all supporting hyperplanes of the PPT-polytope are orthogonal to it. Figure 5.22 shows a low-dimensional illustration of this situation. Relaxing normalizing condition (5.6) creates no new optima. We observe that the polytope becomes unbounded but the ray

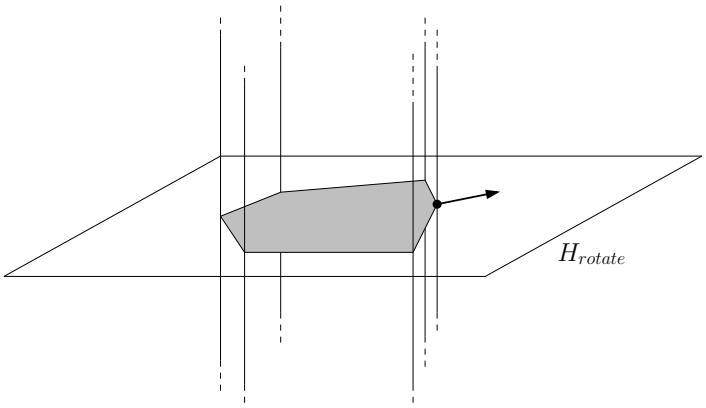


Figure 5.22: Location of the vector of the generic objective function relative to the PPT-polytope.

that passes through the optimum provides no improvement for the objective function. Notice that normalizing condition (5.5) is not orthogonal to H_{rotate}, but since it is an equation it does not matter. □

In the rest of this section we ignore the normalizing condition (5.6).

Theorem 5.22. *The canonical pointed pseudo-triangulation is invariant under affine transformations. This holds also for the max-cppt.*

Proof. Let $a : \mathbb{R}^2 \to \mathbb{R}^2$ be an affine transformation of the form $a(\mathbf{p}) := A\mathbf{p} + \mathbf{t}$ ($\det(A) \neq 0$). Notice that we do not have to modify the parameters f_{ij}. These parameters stay valid for any affine transformation (a consequence of the observations made in [75]). This can also be observed by noticing that the normal fan of the PPT-polytope remains unchanged.

We first study the influence of the translation by $\mathbf{t}$. Adding a translation vector to every point in P has no effect for the conditions (5.4) defining the PPT-polytope. Because the normalizing condition (5.5) holds we have

$$\sum_{i=1}^{n}\langle \mathbf{p}_i + \mathbf{t}, \mathbf{v}_i\rangle = \sum_{i=1}^{n}\langle \mathbf{p}_i, \mathbf{v}_i\rangle + \langle \mathbf{t}, \sum_{i=1}^{n}\mathbf{v}_i\rangle = \sum_{i=1}^{n}\langle \mathbf{p}_i, \mathbf{v}_i\rangle.$$

and therefore the objective function is unchanged.

Let us now study how the transformation $a(\mathbf{p}) := A\mathbf{p}$ changes the linear program. We know that

$$\langle A\mathbf{p}_i, \mathbf{v}_i\rangle = (A\mathbf{p}_i)^T\mathbf{v}_i = \mathbf{p}_i^T(A^T\mathbf{v}_i) = \langle \mathbf{p}_i, A^T\mathbf{v}_i\rangle.$$

As a consequence we can rephrase the constraints given by (5.4):

$$\langle a(\mathbf{p}_i) - a(\mathbf{p}_j), \mathbf{v}_i - \mathbf{v}_j\rangle \leq [0, a(\mathbf{p}_i), a(\mathbf{p}_j)]^2$$

as

$$\langle \mathbf{p}_i - \mathbf{p}_j, A^T\mathbf{v}_i - A^T\mathbf{v}_j\rangle \leq \det(A)^2 f_{ij}.$$

Let us substitute the variables of the PPT-polytope by

$$\tilde{\mathbf{v}}_i := \frac{1}{\det(A)^2} A^T \mathbf{v}_i.$$

Since $\sum_i \langle \mathbf{v}_i, a(\mathbf{p}_i)\rangle = \sum_i \langle A^t \mathbf{v}_i, \mathbf{p}_i\rangle$ we obtain the following linear program for the transformed point set:

$$\begin{array}{rcll} \det(A)^2 \sum_i \langle \tilde{\mathbf{v}}_i, \mathbf{p}_i\rangle & \rightarrow & min. & \text{s.t.} \\ \langle \tilde{\mathbf{v}}_i - \tilde{\mathbf{v}}_j, \mathbf{p}_i - \mathbf{p}_j\rangle & \geq & f_{ij} & \forall i,j \leq n, \\ \langle \tilde{\mathbf{v}}_i - \tilde{\mathbf{v}}_j, \mathbf{p}_i - \mathbf{p}_j\rangle & = & f_{ij} & \text{if } (i,j) \text{ convex hull edge}, \\ \sum \mathbf{v}_i & = & 0 & \end{array} \tag{5.18}$$

This program has the same solution as the canonical program before the affine transformation because of two reasons: (1) The objective functions differs only by a constant multiplicative factor. (2) The normalizing condition $\sum_i \tilde{\mathbf{v}}_i$ is equivalent to (5.5), which means that if A is nonsingular:

$$\sum_i \tilde{\mathbf{v}}_i = \frac{1}{\det(A)^2} A^T \sum_i \mathbf{v}_i = \mathbf{0} \Leftrightarrow \sum_i \mathbf{v}_i = \mathbf{0}.$$

□

Since the cppt is invariant under translation we assume for the rest of this section that the center of gravity of P lies in the origin. Under this assumption the canonical objective function is obviously moment-free. This makes the techniques of Section 5.4 applicable. We can deduce by Theorem 5.20:

Observation 5.23. *Let $\bar{\mathbf{p}}$ be the center of gravity of P. In the canonical pointed pseudo-triangulation every point $\mathbf{p}_i \in P$ is pointed to the direction $\mathbf{p}_i - \bar{\mathbf{p}}$.*

Observation 5.24. *Let $\bar{\mathbf{p}}$ be the center of gravity of P. In the maximal canonical pointed pseudo-triangulation every point $\mathbf{p}_i \in P$ is pointed to the direction $\bar{\mathbf{p}} - \mathbf{p}_i$.*

5.6.1 Convex Point Sets

We continue with the analysis of the cppt for convex point sets. Our goal is to specify the heights h^c, for which $\mathcal{T}_{h^c}$ gives the cppt. As mentioned before, the canonical objective function (5.17) is moment-free. By Lemma 5.9 we know that the cppt supports a non-equilibrium stress of the following type:

$$\begin{array}{rcll} \sum_{j=1}^n u_{ij}(\mathbf{p}_i - \mathbf{p}_j) & = & \mathbf{p}_i & 1 \leq i \leq n, \\ u_{ij} & \geq & 0 & i<j,\ (i,j) \text{ interior edge}. \end{array}$$

We can transform this stress to an equilibrium stress by adding a point $\mathbf{p}_0 = \mathbf{0}$ to P. Furthermore, we connect every point $\mathbf{p}_i$ with an edge of stress -1 with $\mathbf{p}_0$. By construction all points $\mathbf{p}_i$ are in equilibrium. Also the new point $\mathbf{p}_0$ is in equilibrium since the center of gravity lies in the origin. Notice that

the construction is similar to the one used in Lemma 5.11. For convenience we multiply all stresses with -1.

We compute the heights h^c in clockwise order. The plane spanned by the three points $\mathbf{p}_i, \mathbf{p}_{i+1}$ and $\mathbf{p}_0$ is named H_i. Since all planes contain $\mathbf{p}_0$ (the origin) we can express H_i by the equation

$$H_i : z_i(\mathbf{p}) := \langle \mathbf{a}_i, \mathbf{p} \rangle.$$

We place plane H_1, spanned by the points $\mathbf{p}_0, \mathbf{p}_1$ and $\mathbf{p}_n$, in the xy-plane. In the next step we compute the plane H_2 with help of the equation (2.5). Since $\mathbf{p}_0$ lies in the origin and the stress ω_{01} equals $+1$ we obtain

$$\mathbf{a}_2 = \mathbf{p}_1^{\perp}.$$

In this manner we determine the gradients of all planes H_i in clockwise order. We have for any $k \leq n$:

$$\mathbf{a}_k = \sum_{1 \leq i < k} \mathbf{p}_i^{\perp}.$$

The heights of the lifted vertices can be obtained by evaluating $z_i(\mathbf{p}_i) = \langle \mathbf{a}_i, \mathbf{p}_i \rangle$. Figure 5.23 shows an example of the lifting discussed above. Due to the stress signs, the cppt corresponds to the lower convex hull of the lifted point set. The formulation for max-cppt results in different stress signs of the diagonals. Therefore, it corresponds to the upper convex hull given by P and h^c.

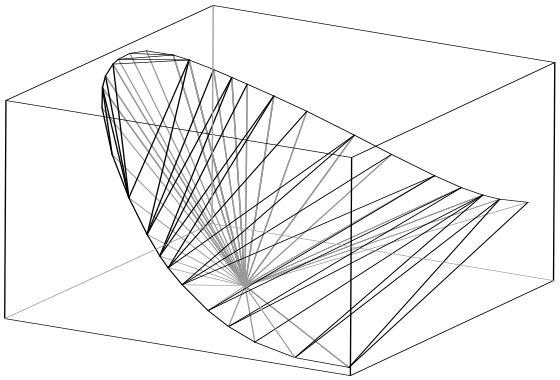


Figure 5.23: A lifting of points in convex position whose lower convex hull realizes the cppt.

The computation of the lifting made an arbitrary choice by placing H_1 in the xy-plane. We can obtain a more symmetric and more canonical lifting by the following construction. Assume we place the plane H_l in the xy-plane and compute the remaining planes starting form H_l in clockwise order. Let $\mathbf{a}_k^l$ be the gradient of the plane H_k when H_l is placed in the xy-plane. Clearly,

$$\mathbf{a}_k^l = \sum_{l < i \leq k} \mathbf{p}_i^{\perp} - \sum_{k \leq i < l} \mathbf{p}_i^{\perp}.$$

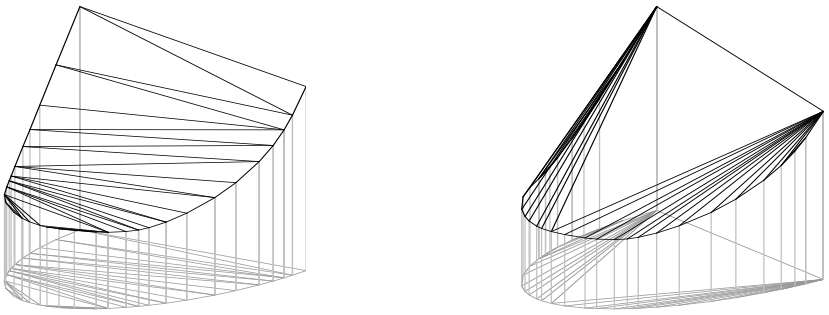
In this fashion we can specify n different liftings that result all in the same lower convex hull (combinatorially). Therefore, the arithmetic mean of all those lifting gives an accumulated lifting whose lower convex hull has the same combinatorial structure. Let δ_{ij} denote the number of points between $\mathbf{p}_i$ and $\mathbf{p}_j$ in clockwise order ($\delta_{ij} := (j-i) \mod n$). The average of all liftings can be expressed as

$$z_i^*(\mathbf{p}) := \frac{1}{n} \sum_{l \leq n} \langle \mathbf{a}_i^l, \mathbf{p}_i \rangle.$$

The height h_i^* of the points in the accumulated lifting is therefore

$$h_i^* = z_i^*(\mathbf{p}_i) = \frac{1}{n} \langle \mathbf{p}_i, \textstyle\sum_{j \leq n} \delta_{ij} \mathbf{p}_j^\perp \rangle.$$

Figure 5.24 shows an example of the accumulated lifting.



cppt max-cppt

Figure 5.24: The accumulated lifting for the cppt and max-cppt.

The heights of the accumulated lifting have a geometric interpretation. We call $m(\mathbf{p}_i) := 1/n \sum_{j \leq n} \delta_{ij} \mathbf{p}_j$ the *weighted mean* of $\mathbf{p}_i$. Furthermore, we observe that

$$\begin{aligned} m(\mathbf{p}_i) - m(\mathbf{p}_{i+1}) &= \frac{1}{n} \left(\sum_{j \leq n} \delta_{ij} \mathbf{p}_j - \sum_{j \leq n} \delta_{i+1,j} \mathbf{p}_j \right) \\ &= \frac{1}{n} \left((1-n)\mathbf{p}_i + \sum_{j \leq n : j \neq i} \mathbf{p}_j \right) \\ &= -\mathbf{p}_i. \end{aligned}$$

Thus the weighted points $m(\mathbf{p}_i)$ form a polygon whose boundary segments are given by the vectors $\mathbf{p}_1, \ldots, \mathbf{p}_n$ and whose center of gravity lies at the origin. We call this polygon *weighted mean polygon*. The accumulated lifting assigns to every point $\mathbf{p}_i$ the height $h_i^* = \langle \mathbf{p}_i, m(\mathbf{p}_i)^\perp \rangle$. Since $\mathbf{p}_i = m(\mathbf{p}_{i+1}) - m(\mathbf{p}_i)$ we can express the heights h_i^* as

$$h_i^* = \langle m(\mathbf{p}_{i+1}) - m(\mathbf{p}_i), m(\mathbf{p}_i)^\perp \rangle = \langle m(\mathbf{p}_{i+1}), m(\mathbf{p}_i)^\perp \rangle.$$

Theorem 5.25. *Let P be a convex point and let h_i^* be the area of the triangle spanned by $m(\mathbf{p}_{i+1}), m(\mathbf{p}_i)$ and $\mathbf{0}$, where $m(\cdot)$ denote the coordinates of the weighted mean polygon of P.*

(1) The cppt of P coincides with regular triangulation given by $\mathcal{T}_{h^*}$.

(2) The max-cppt of P coincides with regular triangulation given by $\mathcal{T}_{-h^*}$.

See Figure 5.25 for the geometric interpretation of the heights h^*. The left side of the Figure shows the point set P. The polygon induced by the weighted means of the points is depicted on the right side. The height of the vertex $\mathbf{p}_i$ equals the shaded area. Notice that we can modify Theorem 5.25 by replacing the vertex $\mathbf{0}$ by any other vertex. However, the computations are easier if we select the origin for our computations.

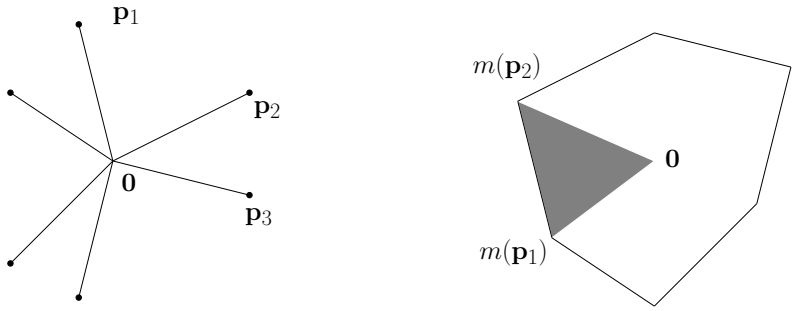


Figure 5.25: A geometric construction of the heights h^*.

The construction of the heights h^* is invariant under affine transformations in the sense that $\mathcal{T}_{h^*}$ stays the same. An affine transformation on the original point set corresponds to the same affine transformation (without translation) of the weighted mean polygon. Therefore, the triangle areas change only by a multiplicative constant. The affine independence is also a consequence of Theorem 5.22.

As further consequence of Theorem 5.22 we notice that the cppt is locally unstable. This means that the perturbation of the location of a single vertex changes the coordinates of every vertex in the weighted mean polygon. Therefore, the combinatorial structure of the cppt might change in parts that are far away from the perturbed vertex.

5.7 Pointed Delaunay-like Pseudo-Triangulations for Simple Polygons

Among the triangulations of a point set the Delaunay triangulation is one of the most prominent. It provides many interesting properties and is widely used for the design of geometric algorithms. We refer to [61] for a list of the most important properties.

Because the Delaunay triangulation fulfills many optimality criteria there exist different ways to define it. For the scope of the thesis the following definition is the most suitable.

Definition 5.26. *Let P be a point set in the plane and h^d a lifting function to the paraboloid given by $h^d(\mathbf{p}_i) = \|\mathbf{p}_i\|^2$. The triangulation $\mathcal{T}_{h^d}$ is called* Delaunay triangulation *of P.*

Our goal is to find a *pointed* pseudo-triangulation that is similar to the Delaunay triangulation. We cannot hope to find a ppt that shares all properties of the Delaunay triangulation. For example, a convex lifting of a ppt with vertices in the interior of the domain will be degenerated (flat) around theses vertices. On the other hand, if the interior of the domain contains no vertices a convex lifting of the ppt exists.

If P refers to a point set in convex position, every ppt is a regular triangulation. In this case the pointed Delaunay pseudo-triangulation should coincide with the Delaunay triangulation. In the following we generalize the above definition such that it defines the pointed Delaunay pseudo-triangulation for vertex-empty simple domains. For the rest of the section let $P = \{\mathbf{p}_1, \ldots, \mathbf{p}_n\}$ be the points of the domain D indexed in cyclic order. We assume that no three points of P lie on a line. The boundary edges of the domain are given by $E_B := \{(i, i+1) | 1 \leq i < n\} \cup \{(n, 1)\}$.

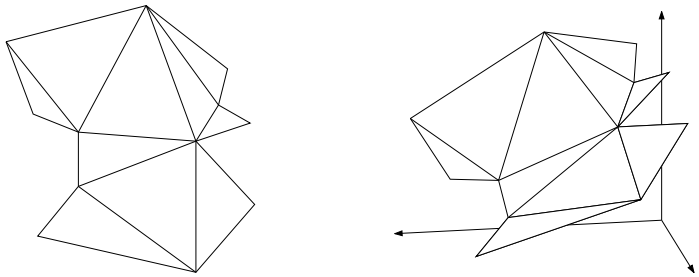
The Delaunay triangulation for simple polygons was introduced in [55, 93]. A more general approach is the concept of *constrained Delaunay triangulation* [24]. It allows to define a Delaunay-like triangulation under the restriction that a prescribed set of edges has to be contained in the triangulation. If the set of edges that has to appear in the constrained Delaunay triangulation equals the set E_B, the constrained Delaunay triangulation gives the Delaunay triangulation of the simple polygon. Recently Shewchuk established the concept of *constrained regular triangulations*, which extends the concept of constrained Delaunay triangulation naturally [81]. A Delaunay triangulation of a simple polygon can be computed in $O(n)$ time [25].

We use the formalism introduced in Section 5.4 to define the Delaunay triangulation of a simple polygon.

Definition 5.27. *Let P denote the points of a simple polygon D in cyclic order. The triangulation $\mathcal{T}_{h^d}$ defined by the maximal locally convex function with respect to the heights h^d on (D, P) is called the* Delaunay triangulation *of the simple polygon D.*

See Figure 5.26(a) for an example of a Delaunay triangulation of a polygon and Figure 5.26(b) for the associated lifting.

It is not hard to modify the above definition such that it defines a pointed Delaunay pseudo-triangulation. We partition the vertices P into corners and non-corners with respect to D. Since P is in general position a vertex can either be a corner or a non-corner. All corners have their large angle between the boundary edges. On the other hand, all non-corners have to realize an angle greater π inside the polygon. This can only be guaranteed when all non-corners are incomplete vertices. To achieve this, the specified upper bounds h_i



(a) (b)

Figure 5.26: A Delaunay triangulation of a simple polygon (a) and the corresponding lifting (b).

are selected such that they lie above the lifted surface. This ensures that for all non-corners of the domain $f_h^*(\mathbf{p}_i) < h_i$ holds.

We define

$$\tilde{h}^d(\mathbf{p}_i) := \begin{cases} \|\mathbf{p}_i\|^2 & \text{if } \mathbf{p}_i \text{ is corner in } P \\ \max_{\mathbf{p}_k}\{\|\mathbf{p}_k\|^2\} & \text{otherwise} \end{cases} \tag{5.19}$$

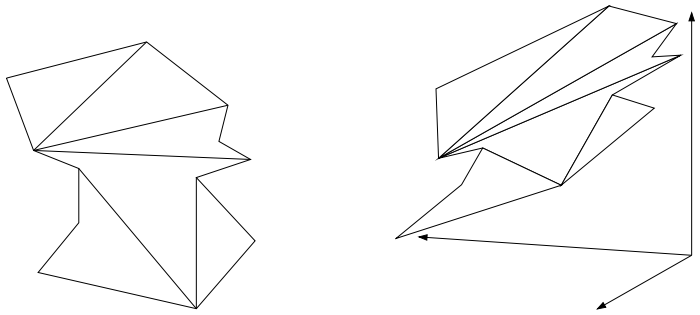
and obtain:

Definition 5.28 (pointed Delaunay pseudo-triangulation of a simple polygon). *Let P denote the points of a simple polygon D. The triangulation $\mathcal{T}_{\tilde{h}^d}$ defined by the maximal locally convex function with respect to the heights $\tilde{h}^d$ on (D, P) is called the* pointed Delaunay pseudo-triangulation *of the simple polygon D.*

An example of a pointed Delaunay pseudo-triangulation is shown in Figure 5.27(a). It is the same polygon that was used as example for the Delaunay triangulation in Figure 5.26. The lifted point set and the induced maximal locally convex surface is shown in Figure 5.27(b). Notice that all non-corners of the polygon are incomplete vertices. Thus every vertex that is incident to an angle greater than π. Since we have a vertex-empty domain the pointed Delaunay pseudo-triangulation can be constructed by a flip sequence of length $O(n^2)$ [5]. The execution of a flip (with update of the heights of the incomplete vertices) costs $O(n^3)$ in the worst case (see also [5]).

We show that the pointed Delaunay pseudo-triangulation of a simple polygon whose corners lie on the convex hull of P can be efficiently computed as linear program over the PPT-polytope. We call polygons with this property *pseudo-convex polygons*. To obtain a linear program that computes the pointed Delaunay pseudo-triangulation we use the observations made in Section 5.4. If D represents a convex polygon one can compute $\mathcal{T}_h$ for any height function h by the linear program stated in Theorem 5.13.

We extend the observations of Theorem 5.13 in the following. Our goal is to lift all corners onto the paraboloid. We have to assure that (1) all non-corners will be pointed, (2) the edges E_B appear in the ppt, and (3) the lifted surface is



(a) (b)

Figure 5.27: A pointed Delaunay pseudo-triangulation of a simple polygon (a) and the corresponding lifting (b).

locally convex. The corners of pseudo-convex polygons are exactly the convex hull vertices. Thus we have to lift only these vertices onto the paraboloid. We use the technique used in the proof of Theorem 5.13 to lift the corners. This gives us the Delaunay triangulation of the convex hull of P. We have no guarantee that the edges E_B appear in this triangulation. But as mentioned before in Section 5.4 we can enforce any (non-crossing and pointed) set of edges to be part of the LP-solution by making the corresponding inequalities in the PPT-polytope tight. Thus, we have additionally

$$\langle \mathbf{v}_i - \mathbf{v}_j, \mathbf{p}_i - \mathbf{p}_j \rangle = f_{ij} \quad \forall (i,j) \in E_B, \tag{5.20}$$

Let us finish with the formal LP description. We denote with $P_c \subseteq P$ the set of corners of D. The vertices P_c are labeled in cyclic order $\mathbf{p}_{c_1}, \mathbf{p}_{c_2}, \ldots$. The square brackets notation $[i,j,k]$ refers now to $[\mathbf{p}_{c_i}, \mathbf{p}_{c_j}, \mathbf{p}_{c_k}]$, and $[i,j,k,l]$ refers to $[(\mathbf{p}_{c_i}, \tilde{h}^d_{c_i})^T, (\mathbf{p}_{c_j}, \tilde{h}^d_{c_j})^T, (\mathbf{p}_{c_k}, \tilde{h}^d_{c_k})^T, (\mathbf{p}_{c_l}, \tilde{h}^d_{c_l})^T]$.

Theorem 5.29. *For a pseudo-convex polygon P (no vertex lies in the origin) the maximization of*

$$\sum_{\mathbf{p}_i \in P_c} \langle \mathbf{v}_i, \mathbf{p}_{c_i} \rangle \frac{[0, i-1, i, i+1]}{[0, i-1, i][0, i, i+1]} \tag{5.21}$$

over the facet of the PPT-polytope given by (5.20) *and* (5.4)–(5.6) *gives the pointed Delaunay pseudo-triangulation of the polygon P.*

Proof. Due to Corollary 5.10 we know that the solution of the linear program stated in the theorem corresponds to a non-equilibrium stress. Moreover, the forces acting at the interior vertices sum up to zero. At every boundary vertex $\mathbf{p}_{c_i}$ the resulting force equals $\mathbf{p}_{c_i}[0, i-1, i, i+1]/([0, i-1, i][0, i, i+1])$. We can apply Lemma 5.11 to construct an equilibrium stress. As consequence the height of the boundary vertices is determined by the pyramid-shaped shell, namely $\|\mathbf{p}_{c_i}\|^2$. All (unconstrained) edges of the pointed pseudo-triangulation have positive stress and are therefore convex (see Corollary 5.10). Notice that

the constrained edges might be concave if we consider the lifting with respect to the domain $conv(P)$. □

Notice that the pyramid tip in the lifting can be any point and is not restricted to lie in the origin (we leave out the computations). Thus, if we have a vertex $\mathbf{p}_i = \mathbf{0}$ we can modify the objective function by replacing $\mathbf{0}$ by some vertex $\mathbf{q}$ not contained in P.

Theorem 5.29 can be easily modified for the computation of any regular pseudo-triangulation of a pseudo-convex polygon. All we have to do is to adjust the heights that appear in the volume of the tetrahedra in equation 5.21.

In [77] we introduced a different linear program to compute the pointed Delaunay pseudo-triangulation for pseudo-convex polygons. The construction there is based on the LP formulation of Theorem 5.14. The geometric interpretation is given by the bowl-shaped shell. The method of Theorem 5.29 uses a pyramid-shaped shell instead. Our approach has the advantage that we do not have additionally to guarantee that the diagonals $(i, i+2)$ lie in the interior of the simple polygon (as done in [77]).

The constructed pointed Delaunay pseudo-triangulation is "Delaunay-like" for two reasons. First of all in the convex case it coincides with the Delaunay triangulation. This follows by definition but is also a consequence of Theorem 5.29. Second, if we start with a convex set, add new points on the boundary edges and move the new vertices slightly into the interior of the convex hull the set of diagonals defining the pointed Delaunay pseudo-triangulation stays the same. This is due to the fact that the lifted surface remains convex if we make the boundary edges concave such that they do not cross any interior diagonal of the Delaunay triangulation. Thus also for these "nearly convex" polygons, the pointed Delaunay pseudo-triangulations is equivalent to Delaunay triangulation of its convex hull.

Chapter 6

NP-Complete Problems for Pseudo-Triangulations

6.1 Preliminaries

In this section we show the NP-completeness of problems related to pseudo-triangulations. For an introduction to NP-completeness we redirect the reader to the classic book of Garey and Johnson [38] and Chapter 34 in the book of Cormen et al. [29].

All following reductions make use of the NP-completeness of Planar 3-Sat. For this reason we start our discussion with introducing this problem.

6.1.1 Planar 3-Sat

The problem Planar 3-Sat is a special version of the well known 3-Sat problem. As in 3-Sat we ask if a formula φ in conjunctive normal form with at most three literals per clause is satisfiable. We denote the variables of φ by $x_1, \ldots, x_n$ and the clauses of φ by $c_1, \ldots, c_m$. If a (negated) variable appears as literal in a clause we say the variable is part of the clause. For a formula φ we define the combinatorial graph $G_\varphi = (V_\varphi, E_\varphi)$ by

$$\begin{aligned} V_\varphi &= \{x_i \mid 0 < i \leq n\} \cup \{c_j \mid 0 < j \leq m\}, \\ E_\varphi &= \{(x_i, c_j) \mid x_i \text{ is part of } c_j\}. \end{aligned}$$

We call a formula φ planar if its graph G_φ is planar. Let us have a look at an easy example. The formula

$$\varphi = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \bar{x}_3 \vee x_4) \wedge (x_3 \vee x_2 \vee \bar{x}_4)$$

is in 3-CNF and planar, since the graph G_φ can be drawn without crossings as seen in Figure 6.1. We obtain as decision problem:

Planar 3-SAT
Input: A planar Boolean formula φ in 3-CNF with variables $x_1, \ldots, x_n$.
Question: Is there a truth assignment to the variables $x_1, \ldots, x_n$ such that φ is true?

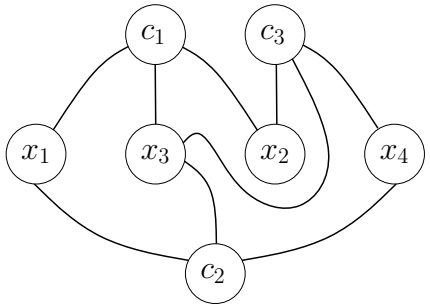


Figure 6.1: Drawing of G_φ.

The test if a formula is in 3-CNF and planar can be done in polynomial time (3-CNF is trivial, for planarity testing see Hopcroft and Tarjan [48]). Hence, Planar 3-Sat is in NP. Due to Lichtenstein:

Theorem 6.1 (Lichtenstein [56]). Planar 3-Sat *is NP-complete.*

Lichtenstein's proof implies also a stronger version of Planar 3-Sat. The problem remains NP-complete, even if the planar formulas have to result in embeddings where the vertices for the variables lie on a common straight line and no edge crosses this line (this was observed by Knuth and Raghunathan [53]).

6.2 The Existence of a Pseudo-Triangulation

We address the problem whether a given graph (embedded in the plane) contains a pseudo-triangulation or not. In the following we call a graph with given plane (straight-line) embedding *geometric graph*. Let $G = (V, E)$ be a geometric graph. We say the graph contains a pseudo-triangulation if we can delete a subset of edges, such that the remaining graph is a pseudo-triangulation. This problem is closely related to the Triangulation Existence problem (Tri) that asks the following:

Triangulation Existence (Tri)
Input: A geometric graph $G = (V, E)$.
Question: Is there a subset $E' \subseteq E$ such that $G' = (V, E')$ is a triangulation?

The NP-completeness of Tri was shown by Lloyd in 1977 [57]. The completeness proof relies on a reduction from CNF-Sat. Even though the ideas in the proof are elegant, the reduction is quite technical and the construction is hard to verify. We give a new proof for the NP-completeness of Tri that is simpler than the version of Lloyd. This is possible because we reduce from Planar 3-Sat, which was not known to be NP-complete in 1977. The structure provided by the instances of the planar version of 3-Sat helps to introduce a simple reduction. In the following we modify the new reduction to obtain results for two related problems.

Pseudo-Triangulation Existence (P-TRI)
Input: A geometric graph $G = (V, E)$.
Question: Is there a subset $E' \subseteq E$ such that $G' = (V, E')$ is a pseudo-triangulation?

Pointed Pseudo-Triangulation Existence (PP-TRI)
Input: A geometric graph $G = (V, E)$.
Question: Is there a subset $E' \subseteq E$ such that $G' = (V, E')$ is a pointed pseudo-triangulation?

6.2.1 NP-Completeness of Tri

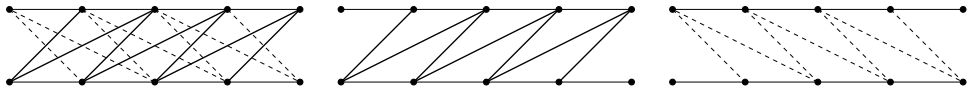
We observe that TRI is in NP, because the verification if a graph is a triangulation can be done in polynomial time. This argument holds also for P-TRI and PP-TRI. We prove the NP-completeness by the following reduction from PLANAR 3-SAT.

Basic idea for the reduction: The reduction transforms a formula φ (planar and in 3-CNF) into a geometric graph $G^t_\varphi = (V, E)$. G^t_φ contains a triangulation if and only if φ is satisfiable. The reduction starts with constructing a non-crossing (straight-line) embedding of the graph G_φ, which always exists for planar graphs [35]. Furthermore, we know that an $O(n) \times O(n)$ grid suffices to embed the graph (see [33, 79]). This guarantees that the representation of the coordinates is not too large. In the next step we replace the edges and vertices of G_φ by larger graphs. These substitution pieces are called *gadgets*. Notice that we have to blow up the linear sized grid embedding by a polynomial factor to create some "space" for the placement of the gadgets.

We introduce four different types of gadgets: Wire, Split, OR and NOT. The Wire-gadget represents the edges of G_φ. The Split-gadget is used to realize variable vertices with degree larger than two. Finally, we use the NOT and OR-gadget as substitution for every vertex c_i belonging to a clause. The wires have two states: true and false. A variable vertex v_i is replaced by a short wire. This wire represents the truth assignment of the corresponding variable. The Split-gadget has one input slot and two output slots. It assures that the output states equal the input state. The NOT-gadget and OR-gadget check the clauses of the formula φ. We construct these gadgets in such a way, that the NOT-gadget will invert the state of a wire. The OR-gadgets leads to "yes"-instances of the Graph G^t_φ if and only if the three input wires fulfill the logical OR (the state of at least one of them is true). This approach is reused for later reductions in this section. We continue with introducing the different gadgets.

Wire-gadget: As mentioned before the gadget stores and propagates the truth assignment of a literal. Figure 6.2(a) shows the gadget. We call the interior edges of the gadget *diagonals*. The gadget contains two triangulations – depicted in Figure 6.2. The triangulation that uses solid diagonals (Figure 6.2(b)) represents the state *true* - the triangulation that uses the dashed diagonals (Figure 6.2(c)) represents the state *false*. The selection of a single

diagonal predetermines the state of the wire. Because there is no triangle in the gadget that has a dashed and a solid diagonal, it is not possible to switch from true to false or vice versa. Notice that it is no problem to enlarge the wire. It



(a) (b) (c)

Figure 6.2: The Wire-gadget for TRI (a), and its two triangulations (b) and (c).

is also possible to perturb the vertex coordinates of the Wire-gadget slightly. Therefore, we can realize bent wires as shown in Figure 6.3.

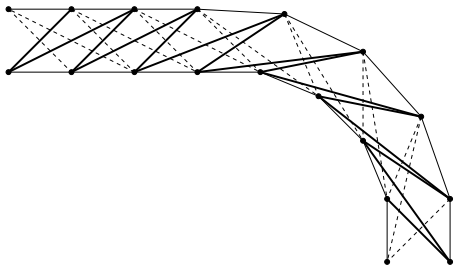
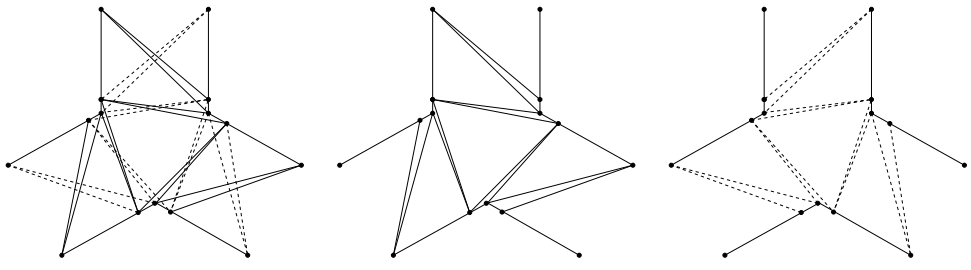


Figure 6.3: A bent Wire-gadget for TRI.

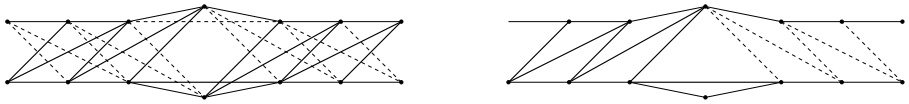
Split-gadget: The Split-gadget has three slots where wires can be attached. Figure 6.4(a) shows the gadget. As in the Wire-gadget there is no triangle that contains a solid and a dashed diagonal. Therefore, it is not possible to select a solid diagonal once you picked a dashed one (and vice versa). Observe that the gadget contains two triangulations (Figure 6.4(b) and (c)). Thus the three connected wires have to provide the same state to allow a triangulation of the Split-gadget.



(a) (b) (c)

Figure 6.4: The Split-gadget for TRI (a), and its two triangulations (b) and (c).

NOT-gadget: Realizing a negation is necessary to evaluate the clauses of φ. See Figure 6.5 for an illustration of the NOT-gadget. The gadget has to change the orientation of the diagonals. In the middle of the gadget every triangulation has to use a triangle with a solid and a dashed diagonal. There is no other triangle with mixed types of diagonals. Thus, the only two triangulations contained in the gadget reverse the orientation of the diagonals. As a consequence the state of the wire attached at the left end has to be opposite to the state of the wire attached at the right end.



(a) (b)

Figure 6.5: The NOT-gadget for Tri (a), and one of its two triangulations (b).

OR-gadget: The OR-gadget is used to evaluate the clauses of φ. The gadget has three input slots where wires can be attached. It is slightly more complex than the other gadgets, because it allows more than two triangulations. The gadget contains diagonals, which cannot be considered as *solid* or *dashed*. We call these diagonals *hybrid*. See Figure 6.6 for a picture of the gadget. We see that three input wires meet in a 9-gon, which is filled with 18 diagonals. Theses diagonals are the hybrid diagonals. The gadget (with edges) is symmetric under rotation by 120 degrees, but it is not symmetric by reflection. We show that the

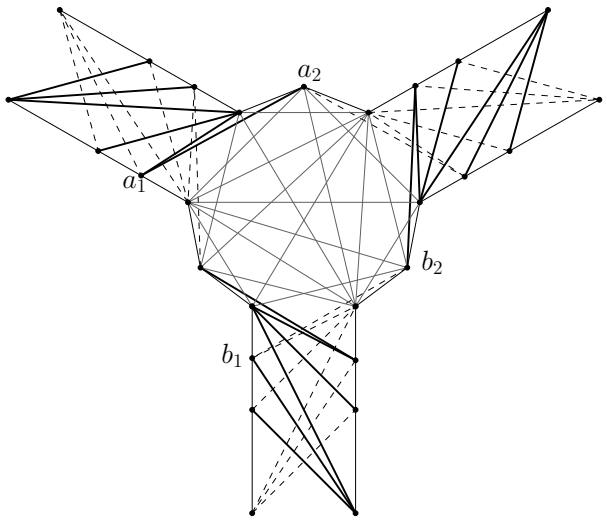
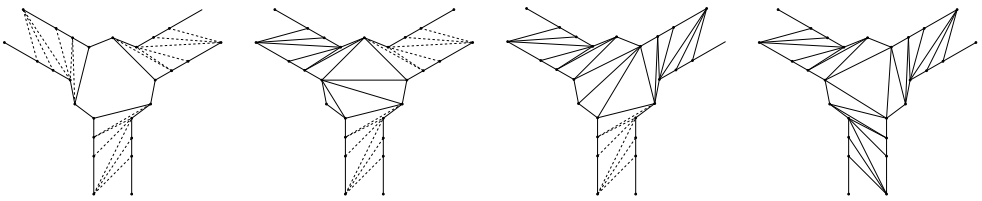


Figure 6.6: The OR-gadget for Tri.

gadget allows a triangulation for all possible input combinations, except when all wires use dashed diagonals. We observe that there are only three hybrid

diagonals crossing the solid diagonal (a_1, a_2) but there are 6 hybrid diagonals crossing the dashed diagonal (b_1, b_2). Having three input wires with dashed diagonals makes it impossible to find a triangulation of the 9-gon. The removal of all hybrid diagonals in this case creates an empty hexagon (Figure 6.7(a)) and therefore allows no triangulation. On the other hand, all other combinations of input states lead to triangulations (as seen in Figure 6.7(b)-(d)). All other combinations are symmetric versions of Figure 6.7. Hence, the OR-gadget works correctly.



(a) (b) (c) (d)

Figure 6.7: The OR-gadget for TRI with different input states.

After the substitution of the edges and vertices of G_φ by the gadgets, we might have holes and pockets inside the resulting graph. We triangulate them arbitrarily. Any edge of the graph that is not crossed by any other edge has to be contained in the triangulation (this is well known, but arises also as a consequence of the results of [7]). Thus, no boundary edge of a gadget can be deleted and the triangulation of the holes does not affect the functionality of the gadgets.

Clearly, a formula φ is satisfiable if and only if there exists a triangulation inside the constructed graph G^t_φ. We omit the detailed analysis of the construction of the embedding of G^t_φ. In particular, we leave out the explanation how to blow up the linear grid appropriately. The calculation of the exact coordinates is only a technicality and gives no deeper insight in the problem.

Notice that we reduce from PLANAR 3-SAT, which does not contain any numeric values. The strong NP-completeness of TRI follows.

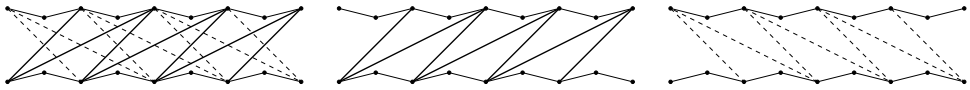
Theorem 6.2. TRI *is strongly NP-complete.*

6.2.2 New NP-Completeness Results

We modify the proof of Theorem 6.2 to obtain reductions for P-TRI and PP-TRI. We start with the reduction for PP-TRI. Let G^{pt}_φ be the geometric graph we built for φ. In the previous proof we constructed the gadgets (except the OR-gadget) such that exactly two triangulations are contained in every gadget (under the assumption that the boundary edges appear in the triangulation). We modify the gadgets such that exactly two pointed pseudo-triangulations are contained in every gadget (except the OR-gadget). Furthermore, all gadgets are constructed such that they have only pointed vertices at their boundary. This property is useful to enforce the boundary gadget edges to appear in every

pseudo-triangulation contained in G_φ^{pt}. Let us assume for now that all boundary gadget edges have to appear in every contained pseudo-triangulation (we prove this fact later with help of Lemma 6.3).

Wire-gadget: The Wire-gadget can be easily obtained from the one used for TRI. Exactly two pointed pseudo-triangulations are contained in the gadget. Again, we associate one of them with *true* (the one with solid diagonals) and one of them with *false* (the one with dashed diagonals). There exists no pseudo-triangle that uses dashed and solid diagonals and therefore switching between the states inside the Wire-gadget is not possible. See Figure 6.8 for the corresponding pictures. Small perturbations of the vertices are valid. Thus, we can realize bent wires without problems.



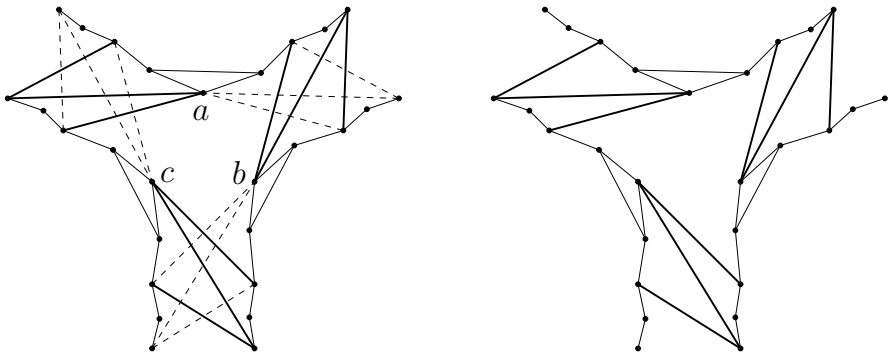
(a) (b) (c)

Figure 6.8: The Wire-gadget for PP-TRI (a), and its two pointed pseudo-triangulations (b) and (c).

Split-gadget: The new Split-gadget is depicted in Figure 6.9(a). Its central area has to be covered by a pseudo-triangle. Such a pseudo-triangle exists only if the state of all attached wires is the same. It is not possible to find a pseudo-triangulation for any other combination of the states. In these cases it is impossible to construct an angle greater than π at one of the points a, b, c. Figure 6.9(b) shows that the large central pseudo-triangle fits perfectly together with the wires if their state is true. The symmetric situation (all wires have state false) is the reflection of Figure 6.9(b).

NOT-gadget: The NOT-gadget is similar to the NOT-gadget for TRI. Due to the updated Wire-gadget no vertex is non-pointed. Figure 6.10 shows the gadget.

OR-gadget: The final gadget in our reduction is the OR-gadget. The gadget is based on the same ideas we used for the OR-gadget for TRI. We have again a 9-gon that includes a number of hybrid diagonals. If the state of an input wire is false (it uses dashed diagonals) 5 hybrid diagonals have to be removed. On the other hand, if the state of an attached wire is true (it uses solid diagonals) we have to remove only two hybrid diagonals. Observe that if all attached wires use dashed diagonals it is not possible to cover the 9-gon with pseudo-triangles (see Figure 6.12(a)). Notice that in all other cases it is possible to find a pointed pseudo-triangulation that is contained in the gadget. Figure 6.12(b)–(d) depicts all possible situations (up to symmetry).



(a) (b)

Figure 6.9: The new Split-gadget (a) and one of its pointed pseudo-triangulations (b).

(a) (b)

Figure 6.10: The new NOT-gadget (a) and one of its pointed pseudo-triangulation (b).

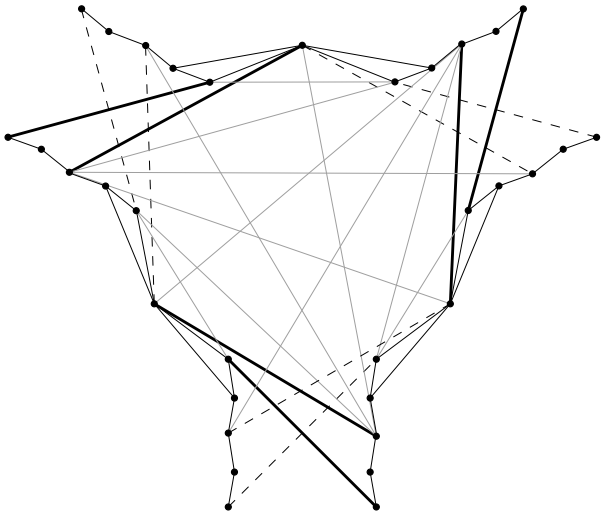
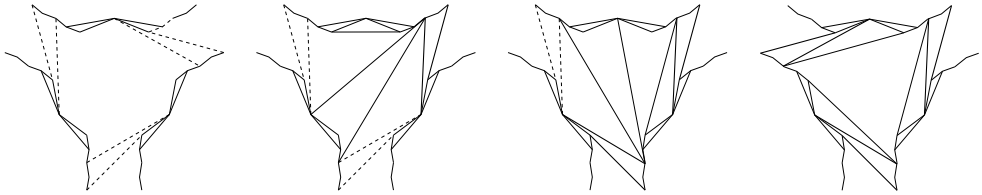


Figure 6.11: The OR-gadget for PP-Tri.



(a) (b) (c) (d)

Figure 6.12: The OR-gadget for PP-TRI with different input states.

The assembled gadgets contain a pointed pseudo-triangulation if and only if the formula φ is satisfiable. Again, it can be realized on a polynomial sized grid by scaling the linear sized grid embedding of G_φ (we omit the technical details again).

In the final step of the reduction we fill the remaining holes and pockets of G_φ^{pt}. This is done by pseudo-triangulating them arbitrarily without violating the pointedness condition of the vertices at the gadget boundaries. We can complete any non-crossing, pointed graph to a pointed pseudo-triangulation (see Streinu [84]). Therefore, all holes and pockets can be filled with pseudo-triangles while maintaining the pointedness property of the boundary vertices. We show that the functionality of the gadgets is not destroyed by these pseudo-triangles.

It suffices to show that the boundary edges of the gadgets have to be contained in every pointed pseudo-triangulation contained in G_φ^{pt} (this was assumed during the discussion).

Lemma 6.3. *Let e be an edge in a geometric graph G where the endpoints of e are pointed. If there is no edge e' in G ($e \neq e'$) that crosses e then every pseudo-triangulation contained in G has to contain e.*

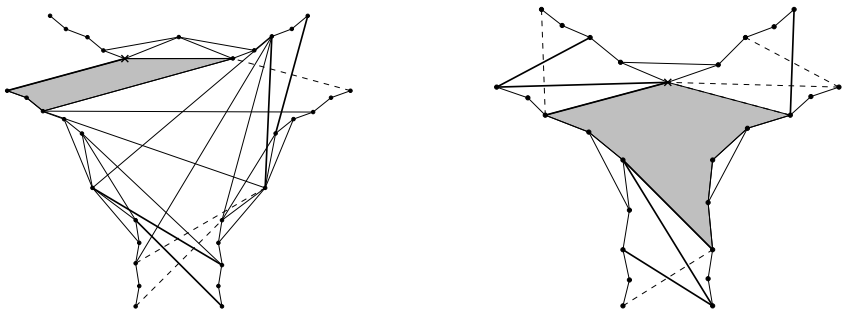
Proof. Assume that there is a pseudo-triangulation PT contained in G without e. Then the (deleted) edge e has to lie in the interior of some pseudo-triangle Δ. Assume that Δ has q vertices. Let PT_Δ be a pointed pseudo-triangulation of the points of Δ which includes Δ as face. PT_Δ has $2q-3$ edges. Adding e to PT_Δ gives one pointed vertex, because the edge count is now $2q-2$. Clearly, the non-pointed vertex has to show up at one of the endpoints of e. As a consequence, adding e to PT gives a non-pointed vertex at one of its endpoints as well. But this is impossible since these points are pointed in G. This contradicts the assumption. □

Due to Lemma 6.3 no boundary edge of the gadgets can be removed without constructing a pseudo-k-gon with $k > 3$. This finishes the proof of the NP-completeness by construction. The strong NP-completeness follows by the reasons given in the proof of Theorem 6.2. We conclude with:

Theorem 6.4. PP-TRI *is strongly NP-complete.*

The reduction $\varphi \to G_\varphi^{pt}$ can also be used to show the NP-completeness of P-Tri. Every yes-instance of Planar 3-Sat is mapped to a graph that contains a pointed pseudo-triangulation and hence a pseudo-triangulation. It remains to show that if G_φ^{pt} contains a pseudo-triangulation, then this pseudo-triangulation has to be pointed. This would imply that G_φ^{pt} contains a pseudo-triangulation only if φ is satisfiable.

Non-pointed vertices in G_φ^{pt} appear in the OR-gadget (6 per gadget) and in the Split-gadget (3 per gadget). These vertices cannot remain non-pointed in a pseudo-triangulation contained in G_φ^{pt}. If we keep the vertices non-pointed we have to delete all edges that are crossed by the edges incident to the small angles of the non-pointed vertices. Figure 6.13 depicts this situation. We observe that a pseudo-k-gon with $k > 3$ is unavoidable. Since all other vertices in G_φ^{pt} are



(a) (b)

Figure 6.13: Maintaining a non-pointed vertex (marked as cross) in the OR-gadget (a) or the Split-gadget (b) produces at least a pseudo-quadrilateral in every crossing free subset of edges.

already pointed, every pseudo-triangulation contained in G_φ^{pt} has to be pointed.

Theorem 6.5. P-Tri *is strongly NP-complete.*

6.3 Bounded-Degree Pseudo-triangulations

In the last section we considered the problem if it is possible to delete edges such that we obtain a (pointed) pseudo-triangulation. Now we look at the opposite setting. Given a graph that is not a pseudo-triangulation yet. Can we add edges such that this graph becomes a pseudo-triangulation? This problem can be answered easily. If the graph is non-crossing, then we can complete it to a triangulation (and hence a pseudo-triangulation). As mentioned before, if the graph is non-crossing and pointed we can complete it to a pointed pseudo-triangulation [84]. A more challenging question is, if we can complete the graph such that it can be completed to a pseudo-triangulation with special properties.

We know that for every point set there exists a pointed pseudo-triangulation with vertex degree at most 5 [51]. Notice that a similar result does not hold

for triangulations. There are point sets with n vertices that require a vertex degree of $n-1$. See Figure 6.14 for such a point set. Since an edge has to be contained in a triangulation if there is no other edge crossing it, a vertex with high degree cannot be avoided in this example. On the other hand, this point set has clearly a pseudo-triangulation with no vertex of degree larger than four. For the more general setting of constrained pseudo-triangulations

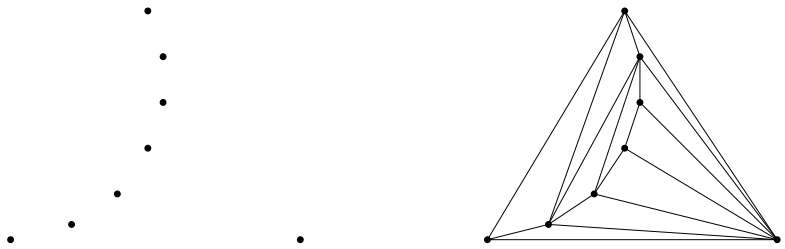


Figure 6.14: A point set with high vertex degree triangulation.

constant degree bounds exist as well [10].

We address the problem if a convex hull of a point set with given edge constraints can be completed to a pseudo-triangulation such that no vertex violates a given degree bound. As edge constraints we consider a set of edges, which has to appear in the pseudo-triangulation. The following incarnations of this question are studied:

k-Constrained Pseudo-Triangulation (k-CPT)
Input: A point set P and a set $C \subseteq P \times P$ of edge constraints.
Question: Is there a pseudo-triangulation of $conv(P)$ that includes C, whose maximal vertex degree is k?

k-Constrained Pointed Pseudo-Triangulation (k-CPPT)
Input: A point set P and a set $C \subseteq P \times P$ of edge constraints.
Question: Is there a pointed pseudo-triangulation of $conv(P)$ that includes C, whose maximal vertex degree is k?

The number k is not part of the input but part of the problem. Related to this question is the equivalent problem for triangulations, also known as the *min-max degree triangulation problem*, which was shown to be NP-complete by Jansen [50].

Theorem 6.6. *For any $k \geq 5$, k-CPT is strongly NP-complete.*

Proof. We prove the theorem by introducing a reduction form Planar 3-Sat. The key elements of the reduction are a set of gadgets that represent edges and vertices of a planar formula. The final result of the reduction is a graph G^3_φ that can be extended to a pseudo-triangulation with maximum vertex degree 5 if and only if φ is satisfiable. The vertices of G^3_φ give the set P and the edges induce the edge constraints C.

We assume that the formula φ satisfies the additional conditions of Knuth and Raghunathan [53], namely: G_φ can be embedded such that all variables

lie on a common line and the clause vertices are above and below this line. Furthermore, no edge crosses this line and every clause contains exactly three literals. The "stronger" version of PLANAR 3-SAT is necessary to show the 3-connectivity and planarity of G^3_φ. The explicit embedding of G^3_φ is computed later.

We use four different kind of gadgets: Wire, Variable, NOT and OR. The functionality of the gadgets is related to the set of gadgets used in proof of Theorem 6.4. The only difference is that we do not have a distinct Split-gadget. Instead we use a Variable-gadget to copy wire signals.

Our goal is an embedding of G^3_φ where every hole (induced by the gadget boundaries) is a convex polygon. For this purpose we construct a 3-connected planar graph G^3_φ based on G_φ. We first define the abstract graph G^3_φ by its gadgets (which are abstract subgraphs). The gadgets work correct for any convex embedding of G^3_φ. Due to Bárány and Rote the planarity and 3-connectedness guarantee a strictly convex polygonal grid embedding that can be computed in polynomial time [14].

Let us start with a high level description of the reduction and then discuss the gadgets more detailed one by one. Every vertex that represents a variable in G_φ is replaced by a Variable-gadget. All vertices that represent a clause are substituted by OR-gadgets. We connect the corresponding Variable-gadgets and OR-gadgets by a Wire-gadget (if the variable appears in the clause as literal) or by a NOT-gadget (if the negated variable appears in the clause as literal). The basic idea of the construction is that the Variable-gadgets provide two feasible triangulations, which we associate with *true* and *false*. Wire and NOT-gadget will propagate the assignments to the OR-clauses, where the evaluation of the clauses takes place.

We start with the discussion of the gadget for $k = 5$ and show later how to modify the gadgets for any $k > 5$:

Variable-gadget: In G^3_φ every variable is associated with a subgraph called *Variable-gadget*. The gadget is depicted in Figure 6.15(a). It contains several connection slots for Wire-gadgets or NOT-gadgets. We can realize any number of these slots. In contrast to Section 6.2 we do not need to "split" the signal of the wire because we realized multiple copies of it already at the Variable-gadget.

Wire-gadget: The Wire-gadget is used to propagate the truth assignment of the Variable-gadgets. It consists of a convex quadrilateral. The quadrilateral is attached at a triangle of the Variable-gadget. There are two ways to triangulate the gadget, but one of the triangulations is forbidden by the degree bound induced by the triangulation of the Variable-gadget. Figure 6.16 shows that there are exactly two ways to triangulate a Variable-gadget with connected Wire-gadgets without violating the degree bound. Notice that selecting one diagonal in the Variable-gadget predetermines the whole triangulation. As mentioned before, the orientation of the diagonals represents the truth assignment of the corresponding variable. We use the truth assignment depending on

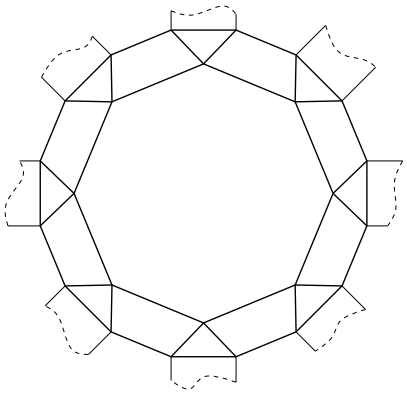
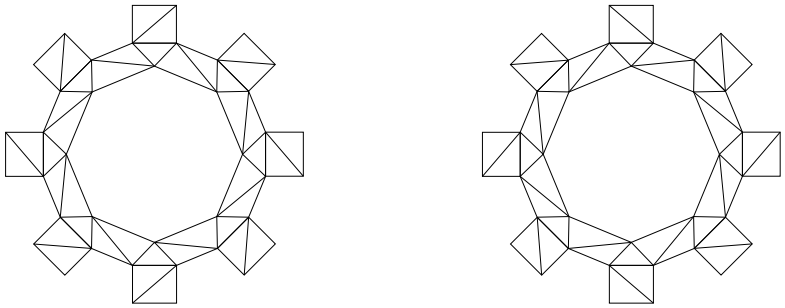


Figure 6.15: The Variable-gadget.

the triangulation as depicted in Figure 6.16.

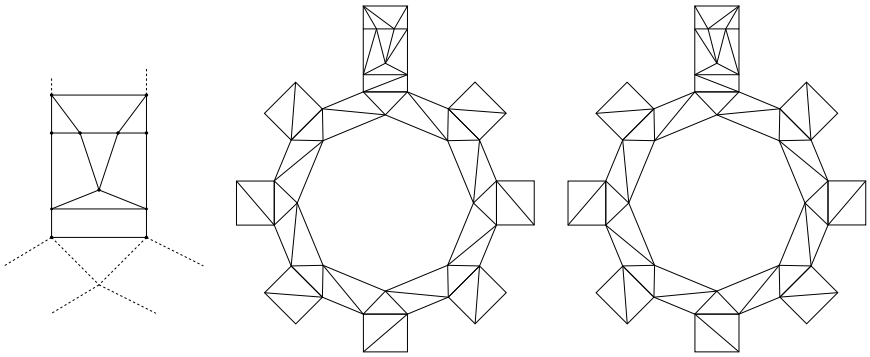


true *false*

Figure 6.16: The associated truth assignment of the Variable-gadget with attached wires.

NOT-gadget: The NOT-gadget is used like a Wire-gadget, but it reverses the orientation of the diagonals chosen in the adjacent Variable-gadget. The gadget is shown in Figure 6.17(a). Notice that if connected to a Variable-gadget (Figure 6.17(b)–(c), top-most connection slot) it has only two triangulations with vertex degree at most 5. Observe that depending on the triangulation of the Variable-gadget only one of these triangulations is feasible. As shown in Figure 6.17(b)–(c) the orientation of the "outer-most" diagonal changes (compared to the other Wire-gadgets attached to the Variable-gadget in the figure).

OR-gadget: The OR-gadget is needed to evaluate the clauses. The abstract graph G^3_φ contains only a convex hexagon with a vertex and three edges inserted as placeholder for the gadget (depicted in Figure 6.18). After the (convex) embedding of G^3_φ we delete the three gray edges and add the dotted edges from



(a) (b) (c)

Figure 6.17: The NOT-gadget (a) attached to a Variable-gadget with its two possible triangulations (b) and (c).

Figure 6.18. There are 8 possibilities how the diagonals at the input Wire/Not-gadget can be oriented, but due to symmetry we have only to check 4 cases. If all input slots correspond to *false*, we have to create a degree 6 vertex at the center (Figure 6.19(a)). All other combinations (Figure 6.19(b)–(d)) can be completed to a triangulation with valid degree bound.

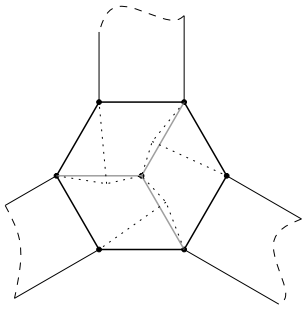
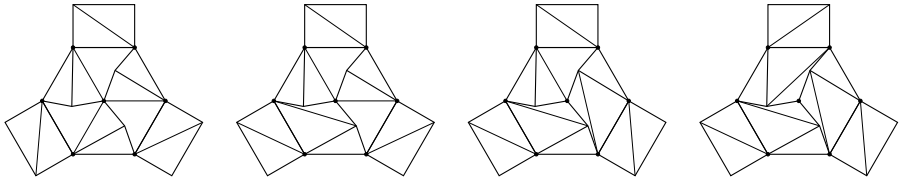


Figure 6.18: The placeholder of the OR-gadget. For the real gadget the gray edges are replaced by the dotted edges.



(a) (b) (c) (d)

Figure 6.19: The triangulation with smallest vertex degree for all types of input combinations.

Since G^3_φ needs to be 3-connected we connect the Variable-gadgets cyclically.

The direct connections between Variable-gadgets (by three Wire-gadgets) propagate no information. The created subgraph represents all variable vertices of G_φ (see Figure 6.20). Notice that G_φ^3 is 3-connected, because deleting any two vertices will not disconnect the graph. Furthermore, it is planar, because it can be drawn without crossings (edges do not have to be necessarily drawn as straight-lines). Due to the algorithm of Bárány and Rote [14] the graph G_φ^3

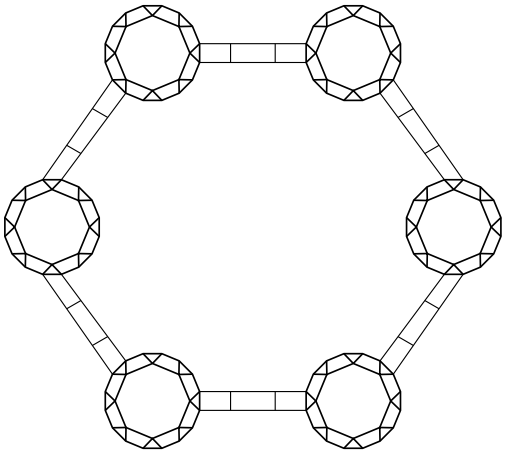


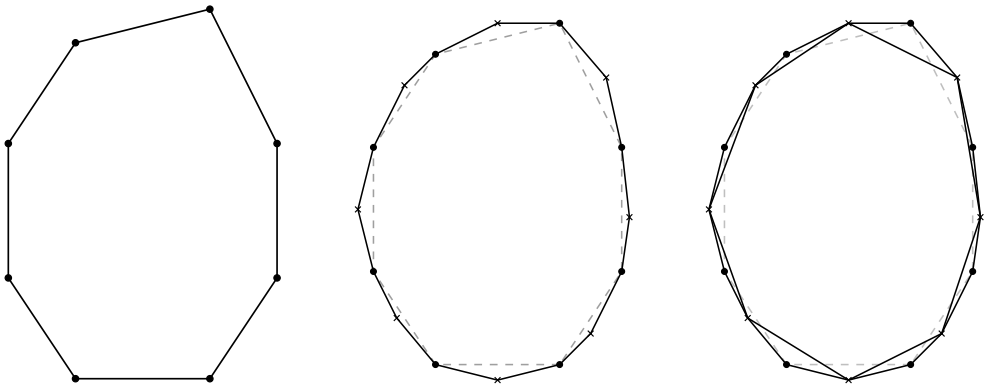
Figure 6.20: The ring of variables for a formula with 6 variables.

can be realized on a polynomial grid with strictly convex faces (in polynomial time). This assures that all gadgets work correctly. After the embedding we have to replace the placeholder for the OR-gadget by the real OR-gadget. This might require a further subdivision of the grid. By construction the faces of the gadget can be decomposed into pseudo-triangles with degree bound 5 if and only if φ is satisfiable.

It remains to show how to fill the holes (induced by the gadget boundaries) of the embedding of G_φ^3 with pseudo-triangles without violating the degree bound. The holes are *convex* polygons. We discuss now how to fill the holes without using a vertex with degree larger than 5 and without increasing the vertex degree of any "old" vertex. We scale by a polynomial factor to obtain enough "space" for the modifications. We omit the technical details.

Filling the holes works in 5 steps. Assume a hole has k boundary vertices. In the first step we subdivide every boundary segment of a hole by adding a new vertex on every boundary edge. We shift the new vertices such that the hole remains a strictly convex polygon. Notice that a small perturbation does not destroy the functionality of the gadgets. After the perturbation the gadgets consist of pseudo-quadrilaterals and pseudo-triangles instead of quadrilaterals and triangles. The correctness of the gadget is guaranteed if the diagonals in every quadrilateral are also the diagonals in the (perturbed) pseudo-quadrilateral. Since the perturbations can be made small this can be guaranteed.

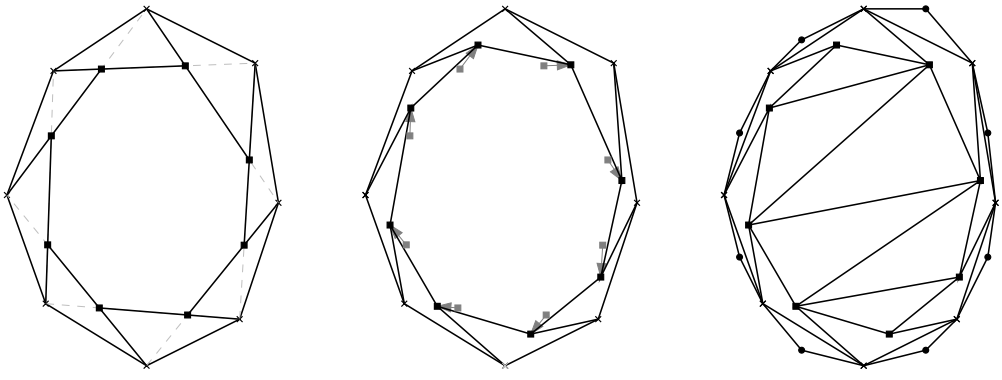
After the first step we have doubled the number of vertices on the boundary of the hole. Every other vertex has degree two. We connect the degree-two vertices in the next step cyclically (see Figure 6.21). This produces a convex



original hole after step 1 after step 2

Figure 6.21: Step 1 and 2 for filling the holes. The crosses mark newly added vertices. Original hole edges are drawn dashed.

k-gon where every vertex has degree 4. Let the vertices of the new k-gon be $\mathbf{p}_1, \ldots, \mathbf{p}_k$ labeled in cyclic order. For every $\mathbf{p}_i$ we define the vertex $\mathbf{q}_i$ as intersection of the diagonals $(\mathbf{p}_i, \mathbf{p}_{i+2})$ and $(\mathbf{p}_{i+1}, \mathbf{p}_{i+3})$ (index shift always modulo k). In the third step we add all segments $(\mathbf{p}_i, \mathbf{q}_i)$. This produces a degenerate situation (on every new segment we have three points on a line). We perturb the vertices $\mathbf{q}_i$ such that the quadrilateral $\mathbf{p}_i, \mathbf{p}_{i+1}, \mathbf{q}_i, \mathbf{q}_{i-1}$ becomes a pseudo-triangulation with $\mathbf{q}_{i-1}$ as non-corner. This can be achieved by placing $\mathbf{q}_i$ in the center between $\mathbf{p}_{i+1}$ and the old position of $\mathbf{q}_i$. The vertices $\mathbf{q}_i$ span a convex k-gon, where every vertex has degree three. Now we triangulate the inner most k-gon in a zig-zag manner. Every vertex degree increases by at most two and hence the degree bound of 5 holds.

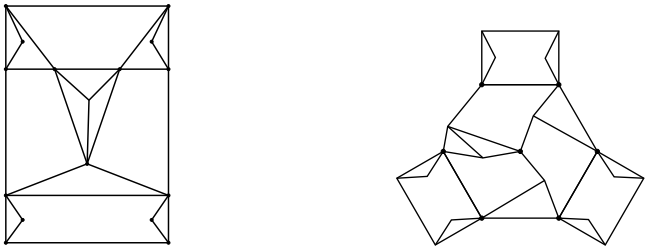


after step 3 after step 4 final

Figure 6.22: Steps 3–5 for filling the holes.

So far we have shown the NP-completeness of 5-CPT. The construction can

be modified such that it works for k-CPT ($k \geq 5$). This can be achieved by increasing the vertex degree of the "important" vertices of G^3_φ *after* its convex embedding. We show the modified gadgets for 6-CPT in Figure 6.23.



NOT-gadget Wire and OR-gadget

Figure 6.23: Gadget modifications for 6-CPT.

□

Theorem 6.7. *For any $k \geq 8$, k-CPPT is strongly NP-complete.*

Proof. We show first that 8-CPPT is NP-complete. The proof relies on the same ideas we used to prove the NP-completeness of Tri: We construct for a formula φ a geometric graph G^d_φ. This geometric graph induces the vertex set P and the edge constraints C.

The starting point for the construction of G^d_φ is an arbitrary straight-line embedding of G_φ on a polynomial grid. As done in the proof of Theorem 6.2 we construct G^d_φ by substituting vertices and edges of G_φ by gadgets. The gadgets used in the reduction are Wire, Split, NOT and OR. The gadgets are constructed to work for a degree bound of 5. Later we show how to modify the gadgets such that they work for degree 8 and higher. The reason why we need the rather large degree bound of 8 will be clear later, when we discuss how to fill the holes after substituting the gadgets. Nevertheless, it is easier to draw and to discuss the construction of the gadgets with the degree bound of 5.

Wire-gadget: The Wire-gadget is used to represent edges of G_φ. It also models the vertices that belong to the variables. Figure 6.24 shows an illustration of the gadget. We notice that there are exactly two ways how to complete the gadget to a pseudo-triangulation with vertex degree at most 5. We associate these pseudo-triangulations with *true* and *false*. Let us assume that the triangulation in the middle of Figure 6.24 corresponds to the state *true*, whereas the bottom-most triangulation in the figure corresponds to the state *false*. It is not possible to switch from true to false inside the wire, because this would result in a vertex of degree 6. Thus, the selection of one edge predetermines the selection of the remaining edges inside the Wire-gadget. Clearly, the position of the vertices can be perturbed without destroying the structure of the gadget. Thus we can realize bent wires easily.

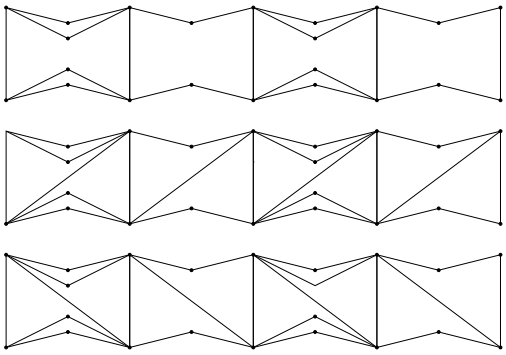


Figure 6.24: The wire gadget (on the top) and the two possible pseudo-triangulations beneath.

Split-gadget: Since we have only one copy of each variable, but different clauses where it could appear, we need the Split-gadget to propagate the assignment of the variables to the clauses. The Split-gadget is connected to three wires and it assures that the state of all three wires has to be the same. Figure 6.25(a) shows the gadget. The correctness of the gadget can be verified easily: An ingoing wire turns one vertex into a degree 5 vertex. This vertex is blocked for any other edges. There remains only one possibility to introduce the diagonal in pseudo-quadrilateral next to the degree 5 vertex. Moreover, there is only one possibility to insert the diagonal in the third pseudo-quadrilateral (without violating the vertex degree bound). Figure 6.25(b) and (c) shows the two possibilities how to pseudo-triangulate the Split-gadget.

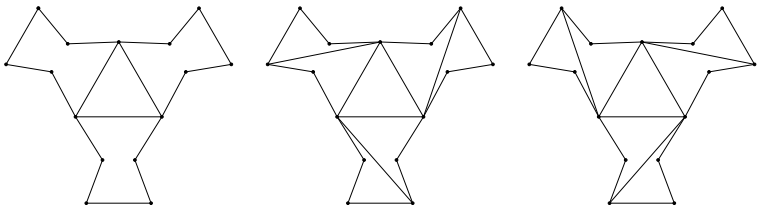
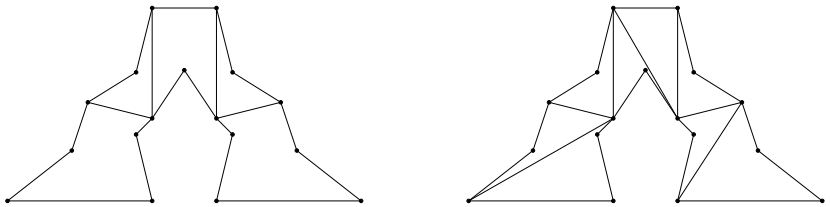


Figure 6.25: The Split-gadget (a) and its pseudo-triangulations (b) & (c).

NOT-gadget: The NOT-gadget (Figure 6.26(a)) works in combination with a Split-gadget. In particular, we use the Split-gadget to duplicate a wire. The two wires (which carry the same signal) are plugged into the two input slots of the NOT-gadget (at the bottom of the figure). Under the assumption that both input signals are the same there are only two pseudo-triangulations with degree at most 5 left. Both will reverse the direction of the diagonals. Figure 6.26(b) shows one of the two feasible pseudo-triangulations.



(a) (b)

Figure 6.26: The NOT-gadget (a) and its pseudo-triangulation (b).

OR-gadget: The OR-gadget consists of a pseudo-hexagon. We have three input slots where the wires have to be attached. Figure 6.27 shows the gadget.

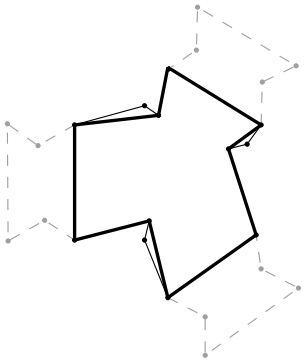


Figure 6.27: The pointed OR-gadget.

We study all possible pointed pseudo-triangulation of the pseudo-hexagon. Notice that there is a one-to-one mapping from the triangulations of a convex hexagon to the pseudo-triangulations of *this* pseudo-hexagon. The corners of the pseudo-hexagon and the vertices of the convex hexagon are indexed in cyclic order (1 to 6). We associate the triangulation that contains the three diagonals d_1, d_2 and d_3 with the pseudo-triangulation that contains the geodesics d_1, d_2 and d_3. Figure 6.28 shows one example of this bijection. The set of triangulations of a convex polygon is well understood. For a convex hexagon we have 14 distinct triangulations. We test for all triangulations if the corresponding pseudo-triangulations fulfills the degree bound. Only 7 pseudo-triangulations of the OR-gadget remain (listed in Figure 6.29).

Remember, that the state of a wire is considered as false, if its diagonals are aligned from bottom/left to top/right. When all incoming wires have state false, the vertices between wires and pseudo-hexagon have all vertex degree 4. Thus, we can only increase the degree from any of them by one. Convince yourself that all valid pseudo-triangulations shown in Figure 6.29 have at least one vertex with two incident diagonals at the boundary to the wire. It is impossible to cover the pseudo-hexagon with pointed pseudo-triangles when

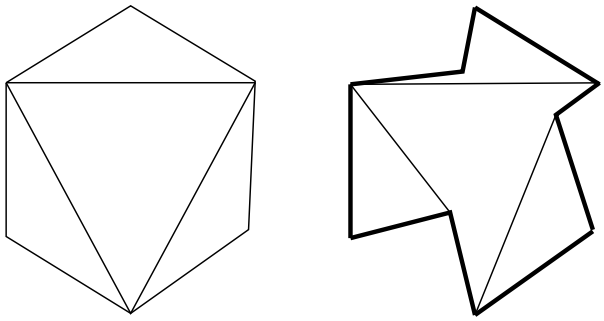


Figure 6.28: Bijection between triangulation of a convex hexagon and the pointed pseudo-triangulation of the OR-gadget.

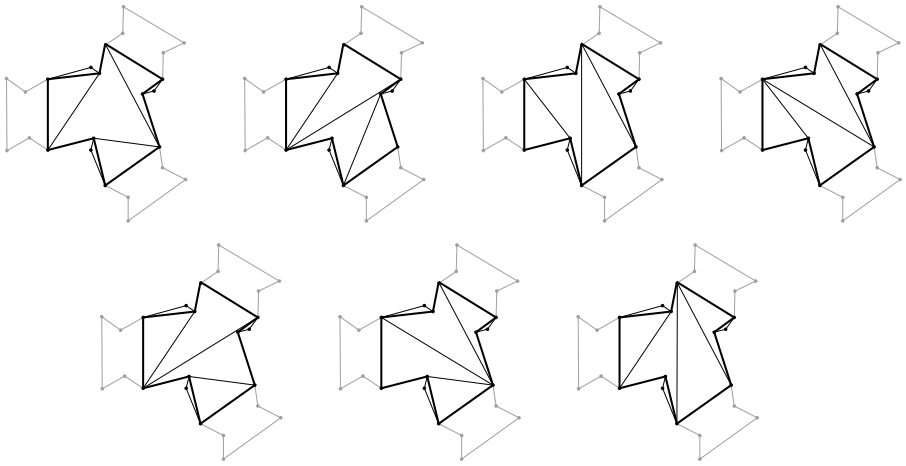


Figure 6.29: The 7 valid pseudo-triangulations of the pointed OR-gadget.

all inputs are false. On the other hand, any other input combination can be finished to a pointed pseudo-triangulation without violating the degree bound. The corresponding pseudo-triangulations are depicted in Figure 6.30.

The discussion shows that φ is satisfiable if and only if the gadgets can be decomposed into pseudo-triangles with no vertex degree greater 5. The gadgets can be modified easily for any $k \geq 5$. We can add artificial edges to increase the vertex degree of the vertices of G^d_φ. Figure 6.31 shows the modifications for the Wire-gadget such that it works for 8-CPT. The additional edges are drawn dashed. The OR-gadget needs no adjustments. It will be connected to a wire in such a way that the vertex degree at the intersection is 7 (for 8-CPT).

Only slight modifications are necessary for the NOT-gadget and the Split-gadget. Both are depicted in Figure 6.32. Also these gadget have to be connected to wires such that the intersection vertices will get vertex degree 7 (for 8-CPT).

The reduction so far produces edge constraints that induce "holes" by the gadget boundaries. It is not clear if these holes can be completed to a pointed

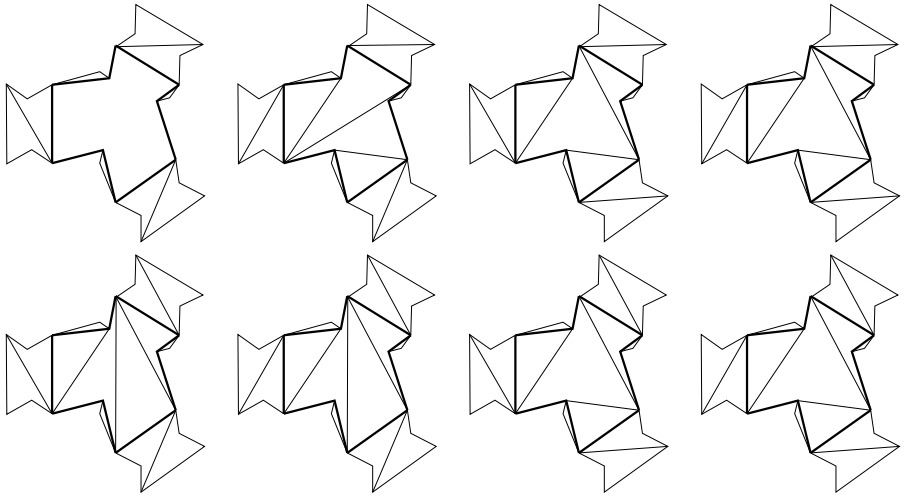


Figure 6.30: All possible input combinations for the pointed OR-gadget

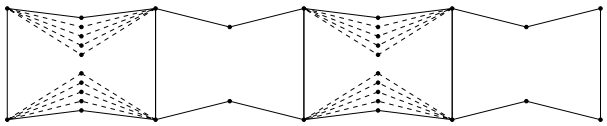
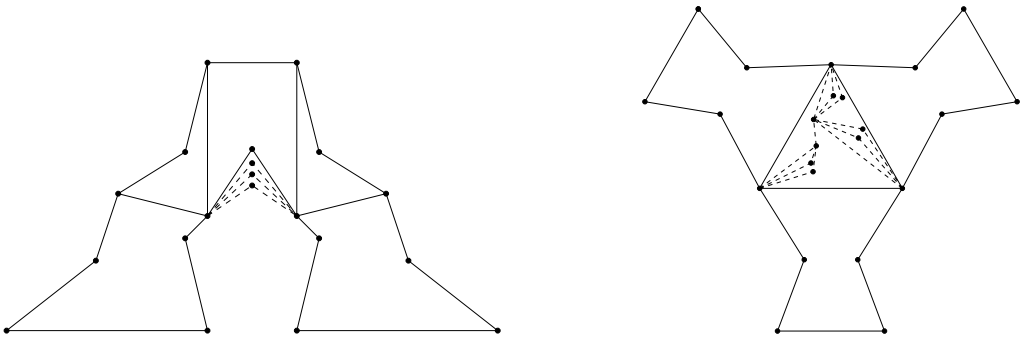


Figure 6.31: The Wire-gadget modified for 8-CPT.



(a) (b)

Figure 6.32: The NOT-gadget and the Split-gadget modified for 8-CPT.

pseudo-triangulations without violating the degree bound. We show how to fill the holes of the final embedding used in the reduction to 8-CPPT.

We assemble the graph G_φ^d with gadgets for 8-CPPT. In order to fill the holes we first modify the boundary of the OR-gadgets as shown in Figure 6.33. With this small modification every other vertex at the boundary has degree

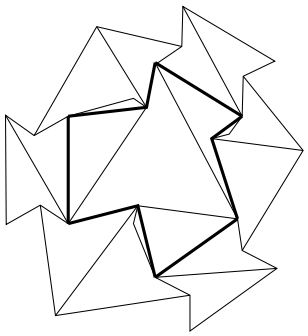


Figure 6.33: Preparation step for pseudo-triangulating the holes.

two or three. At the boundary the large angles of the vertices are alternatively inside or outside the hole. We observe that every vertex with degree 7 has its large angle inside the hole. We can therefore place a pseudo-triangle inside the hole such that the degree 7 vertex lies on a concave 2-chain of it. We link these new pseudo-triangles together (by "glueing" them with triangles) as shown in Figure 6.34. We obtain a polygon inside the "hole" where every vertex has at most degree three. On the other hand we increased the vertex degree only at the gadget boundary vertices with degree two or three. These vertices have now vertex degree 4 or 5 and therefore the degree bound of 8 holds.

Due to Aichholzer et al. [10, Theorem 3] we know that we can always find (efficiently) a pointed pseudo-triangulation for a simple polygon with maximum vertex degree 7. In our setting a vertex of the new boundary of the hole (after inserting the pseudo-triangles) has degree at most three. Thus we have +1 for the degree count compared to the simple polygon case where every boundary vertex has degree two. This implies that we can fill the holes with no vertex of degree greater than 8.

We assume that the polynomial grid is large enough to provide space for the embedding of the gadgets. We omit the technical details for the explicit embedding. □

In the formulation of k-CPT and k-CPPT we the parameter k is part of the problem. The problems without fixed k (the vertex degree bound is part of the input) is a more general question. The NP-completeness of this formulation is an immediate consequence of our results. Moreover, our results are stronger, since they show that the NP-completeness is not related to the actual degree bound, as long as k at least 5 (for k-CPT) or 8 (for k-CPPT).

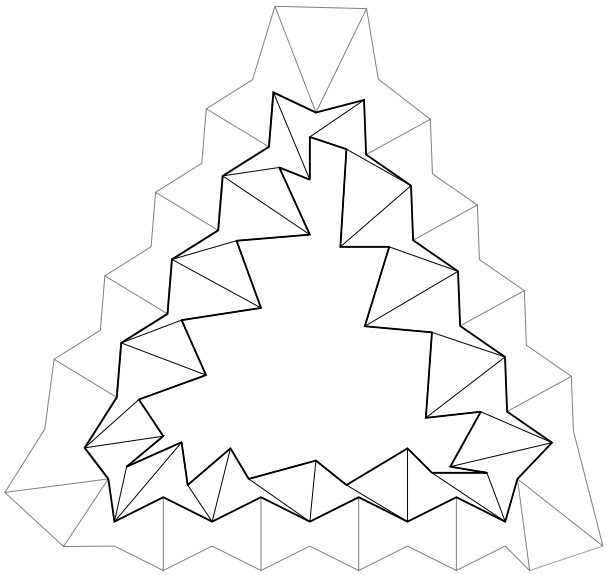


Figure 6.34: Filling the holes with pointed vertices. The additional pseudo-triangles are drawn with thicker edges.

Chapter 7

Conclusion and Open Problems

The lifting technique is a powerful tool to study geometric objects. We have shown how it can be applied to (1) compute small grid embeddings of 3-polytopes and (2) to compute regular (pseudo)-triangulations as linear programs over the PPT-polytope. Even though the lifting of planar graphs is well understood there exists a number of open question that we discuss briefly in the following.

In the weakest formulation the lifting technique lifts a planar simplicial complex to a polyhedral terrain. The crucial point in the proof of the Maxwell-Whiteley theorem is the contractibility of cycles on the surface. This argument is not restricted to piecewise linear surfaces and can be applied to smooth surfaces as well. The question arises, if there is an analogous theorem of the Maxwell-Whitely Theorem for smooth surfaces. Such a theorem requires formalisms and methods of differential geometry. Further studies should discuss if the lifting technique (of planar graphs) is a special (discrete) version of an more general method.

Another way to extend the lifting technique is to apply it for graphs and surfaces with higher genus. The projection of any 3d-mesh into the plane produces an equilibrium stress. However, additional conditions must hold as well. Assume for example that we project a torus in the plane. Clearly any contractible cycle can be contracted and assures the equilibrium. On the other hand, there are cycles that are not contractible. These cycles are responsible for additional conditions. It is an open and challenging task to obtain a general lifting condition that includes the additional conditions for surfaces with higher genus.

Integer Realizations of 3-Polytopes

We come now to the results obtained in this thesis and discuss related open questions. In the first part we studied the realization of 3-polytopes on the grid. We were able to improve the upper bound of Richter-Gebert [72] substantially in Theorem 3.15. The discussion in Chapter 4 showed that the gap between upper a lower bound is still big. As mentioned in Conjecture 4.1 we believe that the truth is closer to the upper bound. Proving or disproving Conjecture 4.1

is an intriguing challenge. It seems reasonable to study grid embeddings of a special types of 3-polytopes first. The so-called *stacked polytopes* were suggested as candidates for small grid-embeddings. A polytope is called stacked, if its graph can be constructed by a K_4 and a sequence of stacking operations, where a stacking operation replaces a triangular face by three new triangular faces sharing a new vertex. For special types of stacked polytopes polynomial size grid embeddings exist [94]. The general case is still open.

Open Problem 1 (Ziegler [95]). *Can every stacked 3-polytope be realized on a polynomial grid?*

Instead of asking for integral coordinates one can also aim at realizing 3-polytopes with integer edge lengths.

Open Problem 2 (Richter-Gebert [72]). *Can every 3-polytope be realized with integer edge lengths?*

The (probably harder) 2d case is open as well.

Open Problem 3 (Harborth). *Can a planar (3-connected) graph be embedded in the plane with integer edge lengths?*

Open Problem 3 is listed in the open problem book of Brass, Moser and Pach [20]. Related open problems can be found there. The problem goes back to Harborth and can be answered positively for the graphs of the Platonic solids [46]. It seems that problems of this type are hard to solve, even if they have a simple formulation. The following number theoretic problems goes back to Richard Guy [44] and is also listed in [20].

Open Problem 4. *Is it possible to place a point relative to a square such that the euclidean distances between the point and each of the corners of the square are rational?*

The integral embedding algorithm for 3-polytopes in Chapter 3 uses a plane spring embedding as intermediate embedding. As crucial feature the embedding supports an equilibrium stress – also at the boundary. Tutte's spring embedding finds application in remeshing, morphing, texture mapping, and other fields (see for example [26, 36, 37, 42]). Our refined construction, which assures an equilibrium stress, might be applicable in these areas to obtain new results. Another interesting question is how we can find an embedding of a planar 3-connected graph with complete equilibrium stress (prescribed for the interior) for any outer face. Remember that the embedding algorithm in Chapter 3 assumes that the outer faces has at most 5 vertices.

Topics in Pseudo-Triangulations

In the second part of the thesis we addressed problems related to pseudo-triangulations. We were able to characterize regular triangulations in the polytope of pointed pseudo-triangulation. We showed that the canonical linear

program over the PPT-polytope results in a pointed pseudo-triangulation with nice properties (e.g. affine independence). The local behavior of the pseudo-triangulation of the canonical linear program depends on the global point set. Our hope is still to find a pseudo-triangulation in the PPT-polytope that can be characterized by an easy local geometric condition. The existence of such a pointed pseudo-triangulation in the PPT-polytope could enable faster enumeration algorithms based on the *reverse search* technique [13]. This might lead to an enumeration scheme similar to the one of Bereg [17] and faster compared to the scheme of Brönnimann et al. [21].

We were able to find a new characterization of pointed pseudo-triangulations with the help of linear programming duality. In particular, we showed that every moment-free load can be canceled by some pointed pseudo-triangulation with equilibrium stresses that are positive at the interior edges. We have used this property to turn a non-regular triangulation into a regular one. As mentioned in Chapter 5 this modification has potential applications in physics (e.g. modeling of diffusion processes). Our approach yields a regular triangulation that might have edge crossings (which can be removed at the expense of new vertices). A different approach might lead to regular triangulations that are more similar to the original triangulation.

Open Problem 5 (Wardetzky[88]). *Given a non-regular triangulation $\mathcal{T}_{nr}$ on the point set P. Does there exists a regular triangulation T_r such that the graph theoretic distance between any two points in both triangulations differs only by a constant.*

Other variants of this problem are also interesting. For example, we can ask how man flips are necessary to turn a non-regular triangulation into a regular one. Furthermore, we can ask these questions not only for "flat" triangulation, but more generally for triangulation of two-manifolds.

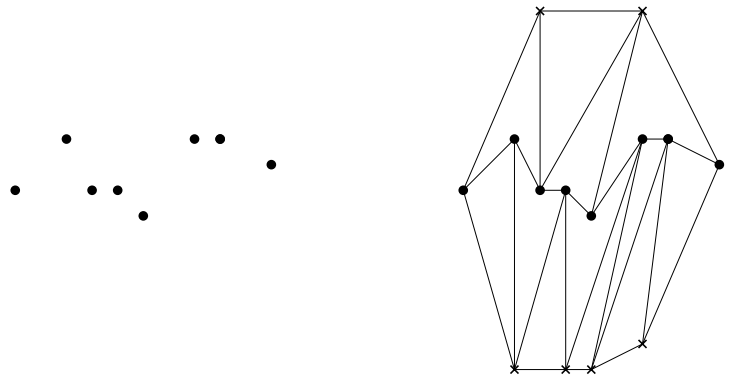
In general many problems on non-regular triangulations are open. As observed in [7] not every non-regular triangulation has such a cyclically dependence like the one in the example shown in Figure 5.19(a). What is known about the maximal number of edges with negative stresses for a given triangulation? We can efficiently decide if we need negative stresses, but determining the number of negative stressed edges seems to be NP-complete. Are certain triangulation problems easier if we restrict ourselves to regular triangulations? For example is the minimum weight regular triangulation problem NP-complete?

In Chapter 6 we investigated some complexity questions related to pseudo-triangulations. We were able to prove several NP-completeness result. It would be nice to show the NP-completeness of the generalized problem for the k-constrained pointed pseudo-triangulation problem.

Open Problem 6. *Show that the question if a given point sets supports a pseudo-triangulation with maximum degree 4 is NP-complete.*

Related to this question is the open problem proposed in [80], which asks how many Steiner points might be necessary to guarantee a pointed pseudo-triangulation with maximum vertex degree 4. One approach to solve this problem is the following: Take the x-monotone path including all the points and add

Steiner points at the outside (with large absolute y-coordinate) to construct a degree 4 pseudo-triangulation. See Figure 7.1 for a sketch of this construction. One can observe that $\Theta(n)$ additional points suffice.



(a) (b)

Figure 7.1: A point set (a) and its degree 4 pseudo-triangulation with $O(n)$ Steiner points drawn as crosses (b).

Open Problem 7. *Is it sufficient for any point set to add $o(n)$ Steiner points such that the new point set supports a pseudo-triangulation with maximum vertex degree 4?*

Another open question related to pseudo-triangulation that we have not addressed in the thesis was mentioned in [78].

Open Problem 8 (Rote et al. [78]). *How can we find a pseudo-triangulation contained in a geometric graph with the minimum number of edges. Is the corresponding decision problem NP-complete?*

Related to the last problem is the question how to find a minimum weight (pointed) pseudo-triangulation. Constant factor approximations exist [43] but the complexity status is still open. Recently Mulzer and Rote proved the NP-completeness of the minimum weight triangulation problem [60]. The reduction in their proof is complicated and does not provide an easy generalization for the (pointed) pseudo-triangulation scenario. Most recently it was observed that the minimum weight pseudo-triangulations contains a pointed vertex for sufficiently large point sets with not too many points on the convex hull [9].

Bibliography

[1] D. M. Acketa and J. D. Žunić. On the maximal number of edges of convex digital polygons included into a square grid. *Počítače a Umelá Inteligencia*, 1(6):549–548, 1982.

[2] D. M. Acketa and J. D. Žunić. On the maximal number of edges of convex digital polygons included into an $m \times m$-grid. *J. Comb. Theory Ser. A*, 69(2):358–368, 1995.

[3] P. K. Agarwal, J. Basch, L. J. Guibas, J. Hershberger, and L. Zhang. Deformable free-space tilings for kinetic collision detection. *I. J. Robotic Res.*, 21(3):179–198, 2002.

[4] A. Aggarwal, L. J. Guibas, J. Saxe, and P. W. Shor. A linear-time algorithm for computing the Voronoi diagram of a convex polygon. *Discrete Comput. Geom.*, 4(6):591–604, 1989.

[5] O. Aichholzer, F. Aurenhammer, P. Brass, and H. Krasser. Pseudo-triangulations from surfaces and a novel type of edge flip. *SIAM Journal on Computing*, 32:1621–1653, 2003.

[6] O. Aichholzer, F. Aurenhammer, P. Brass, and H. Krasser. Spatial embedding of pseudo-triangulations. In *Proc. 19^{th} Ann. ACM Symp. Computational Geometry*, volume 19, pages 144–153, San Diego, California, USA, 2003.

[7] O. Aichholzer, F. Aurenhammer, S.-W. Cheng, N. Katoh, G. Rote, M. Taschwer, and Y.-F. Xu. Triangulations intersect nicely. *Discrete & Computational Geometry*, 16:339–359, 1996.

[8] O. Aichholzer, F. Aurenhammer, and T. Hackl. Pre-triangulations and liftable complexes. *Discrete & Computational Geometry*, 38(4):701–725, 2007.

[9] O. Aichholzer, F. Aurenhammer, T. Hackl, and B. Speckmann. On (pointed) minimum weight pseudo-triangulations. In P. Bose, editor, *Proceedings of the 19th Annual Canadian Conference on Computational Geometry, CCCG*, pages 209–212. Carleton University, Ottawa, Canada, 2007.

[10] O. Aichholzer, M. Hoffmann, B. Speckmann, and C. D. Tóth. Degree bounds for constrained pseudo-triangulations. In *Proceedings of the 15th Annual Canadian Conference on Computational Geometry, CCCG*, pages 155–158, 2003.

[11] F. Aurenhammer. Power diagrams: properties, algorithms, and applications. *SIAM Journal on Computing*, 16(1):78–96, 1987.

[12] F. Aurenhammer and H. Krasser. Pseudo-simplicial complexes from maximal locally convex functions. *Discrete & Computational Geometry*, 35(2):201–221, 2006.

[13] D. Avis and K. Fukuda. Reverse search for enumeration. *Discrete Applied Mathematics*, 65(1-3):21–46, 1996.

[14] I. Bárány and G. Rote. Strictly convex drawings of planar graphs. *Documenta Math.*, 11:369–391, 2006.

[15] J. Basch, J. Erickson, L. J. Guibas, J. Hershberger, and L. Zhang. Kinetic collision detection between two simple polygons. *Comput. Geom.*, 27(3):211–235, 2004.

[16] S. Bereg. Transforming pseudo-triangulations. *Information Processing Letters*, 90(3):141–145, 2004.

[17] S. Bereg. Enumerating pseudo-triangulations in the plane. *Comput. Geom.*, 30(3):207–222, 2005.

[18] L. Billera, P. Filliman, and B. Sturmfels. Constructions and complexity of secondary polytopes. *Advances in Mathematics*, 83:155–179, 1990.

[19] N. Bonichon, S. Felsner, and M. Mosbah. Convex drawings of 3-connected planar graphs. *Algorithmica*, 47:399–420, 2007.

[20] P. Braß, W. Moser, and J. Pach. *Research Problems in Discrete Geometry.* Springer, 2005.

[21] H. Brönnimann, L. Kettner, M. Pocchiola, and J. Snoeyink. Counting and enumerating pointed pseudotriangulations with the greedy flip algorithm. *SIAM J. Comput.*, 36(3):721–739, 2006.

[22] J. Cantarella, E. D. Demaine, H. Iben, and J. O'Brien. An energy-driven approach to linkage unfolding. In *Proceedings of the 20th Annual ACM Symposium on Computational Geometry (SoCG 2004)*, pages 134–143, Brooklyn, New York, June 9–11 2004.

[23] B. Chazelle, H. Edelsbrunner, M. Grigni, L. J. Guibas, J. Hershberger, M. Sharir, and J. Snoeyink. Ray shooting in polygons using geodesic triangulations. *Algorithmica*, 12:54–68, 1994.

[24] L. P. Chew. Constrained Delaunay triangulations. *Algorithmica*, 4(1):97–108, 1989.

[25] F. Y. L. Chin and C. A. Wang. Finding the constrained Delaunay triangulation and constrained Voronoi diagram of a simple polygon in linear time. *SIAM J. Comput.*, 28(2):471–486, 1998.

[26] É. Colin de Verdière, M. Pocchiola, and G. Vegter. Tutte's barycenter method applied to isotopies. *Computational Geometry: Theory and Applications*, 26(1):81–97, Aug. 2003. A conference version appeared in *Proc. Canad. Conf. on Computational Geometry (CCCG)*, 2001.

[27] R. Connelly. Rigidity. In P.M.Gruber and J.M.Wills, editors, *Handbook of convex geometry*, volume A, pages 223–272. North-Holland, 1993.

[28] R. Connelly, E. D. Demaine, and G. Rote. Straightening polygonal arcs and convexifying polygonal cycles. *Discrete and Computational Geometry*, 30:205–239, 2003.

[29] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. The MIT Press, 2nd edition, 2001.

[30] H. Crapo and W. Whiteley. Statics of frameworks and motions of panel structures, a projective geometric introduction. *Structural Topology*, 6:43–82, 1982.

[31] H. Crapo and W. Whiteley. Plane self stresses and projected polyhedra I: The basic pattern. *Structural Topology*, 20:55–78, 1993.

[32] L. Cremona. *Le figure reciproche nella statica grafica*. Milano, Hoepli, 1872. English translation, Graphical Statics, Merchant Books, 2007, reprint.

[33] H. de Fraysseix, J. Pach, and R. Pollack. How to draw a planar graph on a grid. *Combinatorica*, 10(1):41–51, 1990.

[34] H. Edelsbrunner. *Geometry and topology for mesh generation*. Cambridge University Press, New York, NY, USA, 2001.

[35] I. Fáry. On straight lines representation of planar graphs. *Acta Sci. Math. Szeged*, 11:229–233, 1948.

[36] M. S. Floater. Parametrization and smooth approximation of surface triangulations. *Comput. Aided Geom. Des.*, 14(3):231–250, 1997.

[37] M. S. Floater and C. Gotsman. How to morph tilings injectively. *J. Comput. Appl. Math.*, 101(1-2):117–129, 1999.

[38] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[39] I. Gel'fand, M. Karpanov, and A. Zelevinski. *Discriminants, Resultants and Multidimensional Determinats*. Mathematics: Theory and Applications. Birkhäuser, 1994.

[40] J. E. Goodman, R. Pollack, and B. Sturmfels. Coordinate representation of order types requires exponential storage. In *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing*, pages 405–410. ACM, 1989.

[41] M. T. Goodrich and R. Tamassia. Dynamic ray shooting and shortest paths in planar subdivisions via balanced geodesic triangulations. *J. Algorithms*, 23(1):51–73, 1997.

[42] S. J. Gortler, C. Gotsman, and D. Thurston. Discrete one-forms on meshes and applications to 3d mesh parameterization. *Comput. Aided Geom. Des.*, 23(2):83–112, 2006.

[43] J. Gudmundsson and C. Levcopoulos. Minimum weight pseudo-triangulations. *Comput. Geom.*, 38(3):139–153, 2007.

[44] R. K. Guy. *Unsolved problems in number theory.* Springer-Verlag, New York, 2nd edition, 1994.

[45] R. Haas, D. Orden, G. Rote, F. Santos, B. Servatius, H. Servatius, D. Souvaine, I. Streinu, and W. Whiteley. Planar minimally rigid graphs and pseudo-triangulations. *Computational Geometry, Theory and Applications*, 2005.

[46] H. Harborth, A. Kemnitz, M. Möller, and A. Süssenbach. Ganzahlige planare Darstellungen der Platonischen Körper. *Elemente Math.*, 42:118–122, 1987.

[47] P. Harman, editor. *The Scientific Letters and Papers of James Clerk Maxwell*, volume 2. Cambridge University Press, 1995.

[48] J. Hopcroft and R. Tarjan. Efficient planarity testing. *J. ACM*, 21(4):549–568, 1974.

[49] J. E. Hopcroft and P. J. Kahn. A paradigm for robust geometric algorithms. *Algorithmica*, 7(4):339–380, 1992.

[50] K. Jansen. One strike against the min-max degree triangulation problem. *Comput. Geom. Theory Appl.*, 3:107–120, 1993.

[51] L. Kettner, D. Kirkpatrick, A. Mantler, J. Snoeyink, B. Speckmann, and F. Takeuchi. Tight degree bounds for pseudo-triangulations of points. *Computational Geometry: Theory and Applications*, 25(1&2):1–12, 2003.

[52] D. G. Kirkpatrick, J. Snoeyink, and B. Speckmann. Kinetic collision detection for simple polygons. *Int. J. Comput. Geometry Appl.*, 12(1-2):3–27, 2002.

[53] D. E. Knuth and A. Raghunathan. The problem of compatible representatives. *SIAM J. Discret. Math.*, 5(3):422–427, 1992.

[54] C. Lee. The associahedron and triangulations of the n-gon. *European Journal of Combinatorics*, 10:551–560, 1989.

[55] D. T. Lee and A. Lin. Generalized Delaunay triangulation for planar graphs. *Disc. and Comp. Geometry*, 1:201–217, 1986.

[56] D. Lichtenstein. Planar formulae and their uses. *SIAM J. Comput.*, 11(2):329–343, 1982.

[57] E. L. Lloyd. On triangulations of a set of points in the plane. In *Proc. 18th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 228–240, 1977.

[58] J. C. Maxwell. On reciprocal figures and diagrams of forces. *Phil. Mag. Ser.*, 27:250–261, 1864.

[59] J. C. Maxwell. On Bow's method of drawing diagrams in graphical statics, with illustrations from Peaucellier's linkage. *Cambridge Phil. Soc. Proc.*, 2:407–414, 1876.

[60] W. Mulzer and G. Rote. Minimum-weight triangulation is np-hard. *J. ACM*, 55(2), 2008.

[61] O. R. Musin. Properties of the Delaunay triangulation. In *Proceedings of the thirteenth Annual Symposium on Computational Geometry*, pages 424–426. ACM Press, 1997.

[62] D. Orden, G. Rote, F. Santos, B. Servatius, H. Servatius, and W. Whiteley. Non-crossing frameworks with non-crossing reciprocals. *Discrete Comput. Geom.*, 32:567, 2004.

[63] D. Orden and F. Santos. The polytope of non-crossing graphs on a planar point set. *Discrete Comput. Geom.*, 33:275, 2005.

[64] G. Panina. New counterexamples to A. D. Alexandrov's hypothesis. *Adv. Geom.*, 5(2):301–317, 2005.

[65] G. Panina. On hyperbolic virtual polytopes and hyperbolic fans. *Central European Journal of Mathematics*, 4(2):270–293, 2006.

[66] M. Pocchiola and G. Vegter. Order types and visibility types of configurations of disjoint convex plane sets. Technical Report LIENS, January 1994.

[67] M. Pocchiola and G. Vegter. Computing the visibility graph via pseudo-triangulations. In *Proceedings of the 11th Annual Symposium on Computational Geometry (SoCG)*, pages 248–257. ACM Press, 1995.

[68] M. Pocchiola and G. Vegter. Pseudo-triangulations: Theory and applications. In *Symposium on Computational Geometry*, pages 291–300, 1996.

[69] M. Pocchiola and G. Vegter. The visibility complex. *Int. J. Comput. Geometry Appl.*, 6(3):279–308, 1996.

[70] A. Ribó Mor. *Realization and Counting Problems for Planar Structures: Trees and Linkages, Polytopes and Polyominoes.* PhD thesis, Freie Universität Berlin, 2006.

[71] A. Ribó Mor, G. Rote, and A. Schulz. Embedding 3-polytopes on a small grid. In *SCG '07: Proceedings of the twenty-third Annual Symposium on Computational Geometry*, pages 112–118, New York, NY, USA, 2007. ACM.

[72] J. Richter-Gebert. *Realization Spaces of Polytopes*, volume 1643 of *Lecture Notes in Mathematics*. Springer, 1996.

[73] J. Richter-Gebert and G. M. Ziegler. Realization spaces of 4-polytopes are universal. *Bull.Amer.Math.Soc.*, 32:403, 1995.

[74] G. Rote. The number of spanning trees in a planar graph. In *Oberwolfach Reports*, volume 2, pages 969–973. European Mathematical Society - Publishing House, 2005.

[75] G. Rote, F. Santos, and I. Streinu. Expansive motions and the polytope of pointed pseudo-triangulations. *Discrete and Computational Geometry–The Goodman-Pollack Festschrift*, 25:699–736, 2003.

[76] G. Rote, F. Santos, and I. Streinu. Pseudo-triangulations — a survey. In J. E. Goodman, J. Pach, and R. Pollack, editors, *Surveys on Discrete and Computational Geometry—Twenty Years Later*, volume 453 of *Contemporary Mathematics*, pages 343–410. American Mathematical Society, 2008.

[77] G. Rote and A. Schulz. A pointed Delaunay pseudo-triangulation of a simple polygon. In *Proceedings of the 21st European Workshop on Computational Geometry*, pages 77–80, Eindhoven, 2005.

[78] G. Rote, C. A. Wang, L. Wang, and Y.-F. Xu. On constrained minimum pseudotriangulations. In *Computing and Combinatorics, 9th Annual International Conference, COCOON*, pages 445–454, 2003.

[79] W. Schnyder. Embedding planar graphs on the grid. In *Proc. 1st ACM-SIAM Sympos. Discrete Algorithms*, pages 138–148, 1990.

[80] A. Schulz. New results on pseudo-triangulations with low vertex degree. In *Proceedings of the 17th Annual Canadian Conference on Computational Geometry, CCCG*, pages 130–133, 2005.

[81] J. R. Shewchuk. Updating and constructing constrained Delaunay and constrained regular triangulations by flips. In *SCG '03: Proceedings of the nineteenth annual symposium on Computational geometry*, pages 181–190, New York, NY, USA, 2003. ACM Press.

[82] B. Speckmann and C. D. Tóth. Allocating vertex pi-guards in simple polygons via pseudo-triangulations. *Discrete & Computational Geometry*, 33(2):345–364, 2005.

[83] E. Steinitz. Encyclopädie der mathematischen Wissenschaften. In *Polyeder und Raumteilungen*, pages 1–139. 1922.

[84] I. Streinu. A combinatorial approach to planar non-colliding robot arm motion planning. In *IEEE Symposium on Foundations of Computer Science*, pages 443–453, 2000.

[85] T. Thiele. Extremalprobleme für Punktmengen. Master's thesis, Freie Universität Berlin, 1991.

[86] W. T. Tutte. Convex representations of graphs. *Proceedings London Mathematical Society*, 10(38):304–320, 1960.

[87] W. T. Tutte. How to draw a graph. *Proceedings London Mathematical Society*, 13(52):743–768, 1963.

[88] M. Wardetzky. Personal communication.

[89] M. Wardetzky, S. Mathur, F. Kälberer, and E. Grinspun. Discrete Laplace operators: no free lunch. In *SGP '07: Proceedings of the fifth Eurographics symposium on Geometry processing*, pages 33–37, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association.

[90] H. S. White. Cremona's work. *Bulletin of the American Mathematical Society*, 24:238–243, Feb. 1918.

[91] W. Whiteley. Motion and stresses of projected polyhedra. *Structural Topology*, 7:13–38, 1982.

[92] H. Whitney. 2-isomorphic graphs. *Amer. J. Math.*, 55:245–254, 1933.

[93] T. M. Yeung. Generalization of Delaunay triangulation. Master's thesis, Department of EE/CS, Northwestern University, 1980.

[94] F. Zickfeld. *Geometric and Combinatorial Structures on Graphs.* PhD thesis, Technical University Berlin, December 2007.

[95] G. M. Ziegler. Personal communication.

[96] G. M. Ziegler. *Lectures on polytypes*, volume 152 of *Graduate texts in mathematics*. Springer, New York, 1995.

List of Figures

Index

Appendix A

Boundary Stresses of the Equilibrium Embedding with Pentagonal Boundary Face

An important part in Chapter 3 was to generate a 2d embedding with non-trivial equilibrium stress. To construct such an equilibrium embedding even for the boundary points we solve the equation system given by the equations (3.4) and (3.5).

Case 5A

If equation (3.9) holds we have

$$\mathbf{p}_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \mathbf{p}_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \mathbf{p}_3 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \mathbf{p}_4 = \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

and obtain as solution for the remaining point:

$$\begin{aligned} x_5 &= \frac{(\tilde{\omega}_{13} - \tilde{\omega}_{25} - \tilde{\omega}_{24})(\tilde{\omega}_{35} + \tilde{\omega}_{13} - \tilde{\omega}_{24})}{\tilde{\omega}_{35}\tilde{\omega}_{14} + \tilde{\omega}_{14}\tilde{\omega}_{25} + \tilde{\omega}_{25}\tilde{\omega}_{24} + \tilde{\omega}_{13}\tilde{\omega}_{35} - \tilde{\omega}_{35}\tilde{\omega}_{25}}, \\ y_5 &= \frac{\tilde{\omega}_{35} + \tilde{\omega}_{13} - \tilde{\omega}_{24}}{\tilde{\omega}_{35} + \tilde{\omega}_{25}}. \end{aligned}$$

The boundary stresses are

$$\omega_{12} = \frac{\tilde{\omega}_{13}(\tilde{\omega}_{25}^2 + \tilde{\omega}_{24}\tilde{\omega}_{35} + 2\tilde{\omega}_{24}\tilde{\omega}_{25} - \tilde{\omega}_{13}\tilde{\omega}_{25}) + \tilde{\omega}_{14}(\tilde{\omega}_{25}^2 + \tilde{\omega}_{25}\tilde{\omega}_{35} + \tilde{\omega}_{24}\tilde{\omega}_{25} + \tilde{\omega}_{35}\tilde{\omega}_{24})}{\tilde{\omega}_{35}\tilde{\omega}_{25} - \tilde{\omega}_{14}\tilde{\omega}_{25} - \tilde{\omega}_{25}\tilde{\omega}_{24} - \tilde{\omega}_{13}\tilde{\omega}_{35} - \tilde{\omega}_{35}\tilde{\omega}_{14}} - \tilde{\omega}_{12},$$

$$\omega_{23} = \frac{\tilde{\omega}_{13}\tilde{\omega}_{25} + \tilde{\omega}_{25}\tilde{\omega}_{35} + \tilde{\omega}_{24}\tilde{\omega}_{25}}{-\tilde{\omega}_{25} - \tilde{\omega}_{35}} - \tilde{\omega}_{23},$$

$$\omega_{34} = \frac{\tilde{\omega}_{14}(\tilde{\omega}_{35}^2 + \tilde{\omega}_{35}\tilde{\omega}_{13} + \tilde{\omega}_{25}\tilde{\omega}_{35} + \tilde{\omega}_{13}\tilde{\omega}_{25}) + \tilde{\omega}_{24}(\tilde{\omega}_{35}^2 + \tilde{\omega}_{13}\tilde{\omega}_{25} + 2\tilde{\omega}_{13}\tilde{\omega}_{35} - \tilde{\omega}_{35}\tilde{\omega}_{24})}{\tilde{\omega}_{35}\tilde{\omega}_{25} - \tilde{\omega}_{14}\tilde{\omega}_{25} - \tilde{\omega}_{25}\tilde{\omega}_{24} - \tilde{\omega}_{13}\tilde{\omega}_{35} - \tilde{\omega}_{35}\tilde{\omega}_{14}} - \tilde{\omega}_{34},$$

$$\omega_{45} = \frac{\tilde{\omega}_{24}\tilde{\omega}_{25} + \tilde{\omega}_{25}\tilde{\omega}_{14} + \tilde{\omega}_{14}\tilde{\omega}_{35} + \tilde{\omega}_{24}\tilde{\omega}_{35}}{\tilde{\omega}_{13} - \tilde{\omega}_{24} - \tilde{\omega}_{25}} - \tilde{\omega}_{45},$$

$$\omega_{15} = \frac{\tilde{\omega}_{13}\tilde{\omega}_{25} + \tilde{\omega}_{35}\tilde{\omega}_{13} + \tilde{\omega}_{14}\tilde{\omega}_{25} + \tilde{\omega}_{14}\tilde{\omega}_{35}}{\tilde{\omega}_{24} - \tilde{\omega}_{35} - \tilde{\omega}_{13}} - \tilde{\omega}_{15}.$$

Case 5B

If equation (3.9) does not hold we set

$$\mathbf{p}_1 = \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \mathbf{p}_2 = \begin{pmatrix} 1 \\ y_2 \end{pmatrix}, \mathbf{p}_3 = \begin{pmatrix} 1 \\ y_3 \end{pmatrix}, \mathbf{p}_4 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \mathbf{p}_5 = \begin{pmatrix} -1 \\ 0 \end{pmatrix}.$$

This leads to the solution

$$y_2 = -2\frac{\tilde{\omega}_{24}\tilde{\omega}_{13} + \tilde{\omega}_{24}\tilde{\omega}_{35} + \tilde{\omega}_{25}\tilde{\omega}_{13} + 2\tilde{\omega}_{25}\tilde{\omega}_{35} - \tilde{\omega}_{13}^2 - 2\tilde{\omega}_{13}\tilde{\omega}_{35} - \tilde{\omega}_{35}\tilde{\omega}_{14}}{\tilde{\omega}_{24}\tilde{\omega}_{35} + 2\tilde{\omega}_{25}\tilde{\omega}_{13} + 2\tilde{\omega}_{25}\tilde{\omega}_{35}},$$

$$y_3 = 2\frac{\tilde{\omega}_{24}\tilde{\omega}_{13} + \tilde{\omega}_{24}\tilde{\omega}_{35} + \tilde{\omega}_{25}\tilde{\omega}_{13} + 2\tilde{\omega}_{25}\tilde{\omega}_{35} - \tilde{\omega}_{24}^2 - 2\tilde{\omega}_{24}\tilde{\omega}_{25} - \tilde{\omega}_{14}\tilde{\omega}_{25}}{\tilde{\omega}_{24}\tilde{\omega}_{35} + \tilde{\omega}_{25}\tilde{\omega}_{13} + 2\tilde{\omega}_{25}\tilde{\omega}_{35}},$$

$$\omega_{12} = -\tilde{\omega}_{24} - 2\tilde{\omega}_{25} - \tilde{\omega}_{12},$$

$$\omega_{23} = \frac{-\tilde{\omega}_{25}(\tilde{\omega}_{13}^2 + 2\tilde{\omega}_{13}\tilde{\omega}_{35} + 2\tilde{\omega}_{24}\tilde{\omega}_{35}) - \tilde{\omega}_{35}(\tilde{\omega}_{24}^2 + \tilde{\omega}_{25}\tilde{\omega}_{14})}{2\tilde{\omega}_{35}(\tilde{\omega}_{24}+\tilde{\omega}_{25}-\tilde{\omega}_{13}-\frac{1}{2}\tilde{\omega}_{14})+2\tilde{\omega}_{25}(\tilde{\omega}_{13}+\tilde{\omega}_{35}-\tilde{\omega}_{24}-\frac{1}{2}\tilde{\omega}_{14})-(\tilde{\omega}_{13}-\tilde{\omega}_{24})^2} - \tilde{\omega}_{23},$$

$$\omega_{34} = -\tilde{\omega}_{14} - 2\tilde{\omega}_{15} - \tilde{\omega}_{34},$$

$$\omega_{45} = \tilde{\omega}_{24} - 2\tilde{\omega}_{35} - \tilde{\omega}_{13} - \tilde{\omega}_{45},$$

$$\omega_{15} = \tilde{\omega}_{13} - 2\tilde{\omega}_{25} - \tilde{\omega}_{24} - \tilde{\omega}_{15}.$$

Curriculum Vitae

Personal Information:

Name:	André Schulz
Date of Birth:	January 3rd, 1977
Place of Birth:	Frankfurt (Oder), Germany

Education:

1983 – 1991	Primary school, Lebus.
1991 – 1996	High School, *Carl Friedrich Gauß*, Frankfurt (Oder).
1996 – 2003	Study of Computer Science, *BTU Cottbus*, Germany, Diploma.
2003 – 2005	Ph. D. Studies, DFG Project, *Pseudotriangulierungen und Bewegungen von Gelenkssystemen* at *Freie Universität Berlin*.
since 2005	Ph.D. Studies as scientific assistent at *Freie Universität Berlin*.