Karlsruhe Institute of Technology

Embedding Graphs on Non-Standard Grids

Master Thesis of

Daniel Patejdl

At the Department of Informatics
Institute of Theoretical Computer Science

Reviewers:	Prof. Dr. Dorothea Wagner
	Prof. Dr. Peter Sanders
Advisors:	Dr. Marcus Krug
	Dr. Ignaz Rutter

Time Period: 1st November 2011 – 30th April 2012

Statement of Authorship

I hereby declare that this document has been composed by myself and describes my own work, unless otherwise acknowledged in the text.

Karlsruhe, 30th April 2012

Deutsche Zusammenfassung

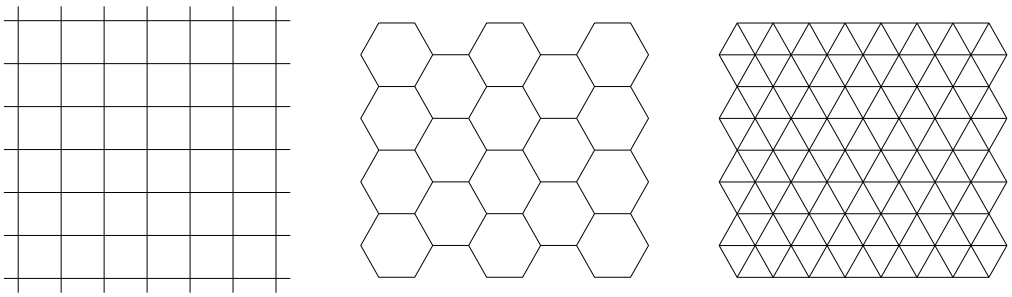
Graphen eignen sich nicht nur zur algorithmischen Lösung von Problemen, sondern auch dazu, in visualisierter Form Zusammenhänge für das menschliche Auge darzustellen. Eine Variante der Darstellung ist die Einbettung von Graphen in Gitter. Das Standard-Gitter ist das orthogonale Gitter, welches durch rechtwinklig verlaufende Linien konstruiert wird. In dieser Arbeit werden Einbettungen von Graphen in Nicht-Standard-Gitter betrachtet, die bisher nur rudimentär erforscht sind. Zum einen das Bienenwabengitter, eine Anordnung von regulären, zusammenhängenden Sechsecken, auf dem verschiedene Algorithmen zum Zeichnen von Bäumen präsentiert werden. Die von den Algorithmen auf dem Gitter gezeichneten Bäume werden anschließend hinsichtlich ihres Platzverbrauchs, ein häufig auftretendes Optimierungskriterium, analysiert. Zum anderen wird ein regulär trianguliertes Gitter, auch hexagonales Gitter genannt, thematisiert, welches aus einer regelmäßigen Teilung der zweidimensionalen Ebene in Dreiecke hervorgeht. Durch eine Scherung des Gitters erhält man ein Gitter, welches strukturelle Ähnlichkeiten zum orthogonalen Gitter aufweist. Diese Ähnlichkeiten erlauben es, bekannte algorithmische sowie komplexitätstheoretische Ergebnisse vom orthogonalen auf das hexagonale Gitter zu übertragen. Es wird gezeigt, dass das Problem, einen gegebenen 4-planaren Graphen ins hexagonale Gitter einzubetten, sodass alle Knoten auf Gitterkreuzungen liegen und alle Kanten durch kürzeste Wege auf dem Gitter verlaufen, wie im orthogonalen Fall NP-schwer ist. Eine eingeschränkte Variante des Problems, bei dem zusätzlich eine Abbildung von Knoten des Graphen auf Punkte des Gitters gegeben ist, erweist sich ebenfalls NP-schwer; erneut wie im orthogonalen Fall. Es wird ein Algorithmus vorgestellt, der bestimmte Instanzen des Problems effizient löst.

Contents

1. Introduction

In informatics, graphs are often used as a theoretical approach to solving problems algorithmically. However, graphs are not at all restricted to this kind use. They can as well serve as a visualization concept for all kinds of relations, for example computer networks, route maps, and different types of charts. Oftentimes, graphs are drawn or embedded in the Euclidean plane. Thus, edges that are incident to the same vertex may form arbitrarily small angles at the vertex. The higher the degree of a vertex gets, the more difficult it may become to read or identify information in the drawing and the less visually appealing the drawing will be to the human eye. This quality criterion is referred to as *angular resolution*. Therefore, when drawing a graph, one should choose angles as big as possible. One possibility to deal with this issue is to use grids as the underlying drawing area for the graph in question. There are further quality criteria which are subject to obtimization, such as the area occupied by a drawing, the number of edge bends and crossings introduced, the length of edges, and criteria regarding the symmetry or shape of drawings.

When using grids, vertices are usually mapped to intersections of the grid lines and edges are chosen to run only along the lines of the grid. There are variants that allow edges to also run off the grid. The standard grid considered is the *orthogonal grid*, which is an orthogonal or perpendicular arrangement of two sets of lines in the plane—see Figure 1.1a. Usually, one set of lines is chosen to run horizontally from left to right, while the other set



(a) The orthogonal grid. (b) The honeycomb grid. (c) The hexagonal grid.

Figure 1.1: The orthogonal grid is the standard grid considered when drawing or embeddings graphs on a grid. This thesis, however, focuses on two non-standard grids: the honeycomb grid and the hexagonal grid.

of lines is chosen to run vertically from the bottom to the top. More precisely, horizontal lines have a slope of zero and vertical lines have a slope of one. A lot of time and effort has been invested with respect to drawing or embedding graphs on the orthogonal grid, as we will see in Chapter 2 on related work.

There are various other, non-standard grid structures which, so far, have only been studied rudimentarily at best. In this thesis, we will study two of them to provide fundamental insight into these structures. The first such grid structure is the *honeycomb grid*, which will be dealt with in Chapter 3. The honeycomb grid is a tesselation of regular hexagons in the plane—see Figure 1.1b. We will port a drawing concept from the orthogonal grid, the *h-v drawing*, to the honeycomb grid. An h-v drawing of a graph is essentially a drawing such that each edge is realized either as a rightward horizontal or a downward vertical line segment, but not both. Furthermore, no edges intersect. We will show how to draw complete binary trees on the honeycomb grid according the h-v drawing convention by using an algorithmic idea due to Crescenzi *et al.* [CP97]. Then, we introduce another drawing concept which we refer to as *sector drawing*. Basically, a sector drawing partitions the honeycomb grid into three regions (*sectors*). Edges are neither allowed to cross sector borders, and thus are required to run in only one sector, nor must they cross each other. Additionally, for both h-v and sector drawings, we require vertices to be placed only on the intersections of the grid lines. We analyze both types of drawings especially with respect to the area needed by such drawings.

The second non-standard grid structure we study is the *hexagonal grid*, also referred to as the *triangular grid*, which we will focus on in Chapter 4. The hexagonal grid consists of three sets of lines at 60° angles to each other—see Figure 1.1c—and, at least visually, resembles the honeycomb grid to some extent. Our studies are greatly motivated by the fact that we can use fundamental knowledge gained from research conducted regarding the orthogonal grid [KKRW10]. In particular, we will consider the problem of Geodesic Point-Set Embeddability (Geodesic PSE): Given a graph G and a finite set P of points on the grid, we ask whether G can be embedded such that the vertices of G are mapped to the points in P and all edges of G are represented by shortest paths (*geodesics*) running only on the grid. The problem is known to be NP-hard on the orthogonal grid, and we will show that it is NP-hard on the hexagonal grid as well. We will then consider another problem initially introduced on the orthogonal grid, named Labeled Geodesic Point-Set Embeddability (Labeled Geodesic PSE). Here, in addition to the input graph, we are given a mapping that determines the correspondence between vertices of the graph and points on the grid. Labeled Geodesic PSE is NP-hard on the orthogonal grid, and we will prove that is NP-hard on the hexagonal grid, too. Moreover, we shall see that certain problem instances of Labeled Geodesic PSE, which we will refer to as *sparse*, can be solved efficiently. The corresponding algorithm, as well as the hardness proofs, are based on results carried out by Katz *et al.* [KKRW10].

Finally, Chapter 5 summarizes our results and concludes this thesis by giving a brief outlook regarding possible future work.

2. Related Work

Already in 1948, István Fáry [Fár48] considered straight-line representations of planar graphs. He showed that every planar graph has an embedding without crossings in which edges are straight line segments and the vertices are points[1], which is also referred to as *Fáry's theorem*, and such straight-line embeddings are called *Fáry embeddings*.

In 1960, William Thomas Tutte did further research on the topic and investigated especially convex representations of graphs in the plane [Tut60]. In 1963, he presented an algorithm that computes convex representations of 3-connected planar graphs [Tut63] by solving a set of linear equations. Using results by Lipton *et al.* [LRT79], the algorithm can be implemented to run in $\mathcal{O}(n^{1.5})$ time, where n denotes the number of vertices of the graph. There are further authors who in the 1960s [AP61] devoted their time to graph drawings and embeddings, among them Donald Knuth [Knu63]. Later, Thomassen [Tho80] characterized the class of graphs that admit a convex drawing, and Chiba *et al.* [CON85] gave a linear-time algorithm for this class using Thomassen's characterization. The algorithm can be extended to general planar graphs producing straight-line drawings, where the faces of each 3-connected component are drawn as convex polygons [TDBB88].

Of special interest has always been the "appropriate" visualization of graphs in terms of a huge variety of aesthetic criteria, such as the number of edge bends, the lengths of edges, the area needed for the visualization of the graph, criteria regarding shape, symmetry and crossings in the drawing of the graph, angular resolution, and many more. For this, researchers have begun to study the problem of finding embeddings of graphs into grids, that is, settling the question, whether the vertices of a graph can be mapped to intersections of the grid lines, and whether the edges of that graph can be routed along the lines of the grid, connecting the corresponding vertices. Such embeddings into grids may result in (more) appealing visualizations or drawings; thus the interest in the topic. There are further applications of graph embeddings into grids that justify the research, for example applications in VLSI chip design. A related problem is *point-set embeddability*, where, in addition to the input graph, a bijection between vertices and grid points is given as part of the input.

In the following, we will first consider previous work related to grid embeddings or drawings and present some general results. Since point-set embeddings render an important part

[1] Actually, Fáry [Fár48], Wagner [Wag36], and Stein [Ste51] showed this independently.

of this thesis, we then outline previous results related to this kind of grid embeddings. Afterwards, we focus on optimization criteria, namely area end edge bends, and we will cover a variety of related work with respect to those two criteria. In this thesis, we will analyze our drawings with respect to the area required as well. For a given graph G, the area occupied by a drawing or embedding of G is subject to minimization. In many cases, the total number of edge bends, and the maximum number of bends per edge are subject to minimization as well. More often than not, it is the case that both criteria are considered for minimization simultaneously. As such, it is difficult to consider one without having to consider the other as well, and thus a certain overlapping cannot be avoided when trying to classify previously done work. Lastly, we focus on an important tool used for graphs drawings, the *canonical ordering*, and present results that are based on this concept.

Grid Embeddings.

The standard grid considered in the task of embedding graphs into grids is the orthogonal grid. One of the most important results is a $\mathcal{O}(n^2 \log n)$-time algorithm by Tamassia [Tam87], later improved to $\mathcal{O}(n^{7/4}\sqrt{\log n})$ by Tamassia and Garg [GT97], and then recently again by Cornelsen and Karrenbauer to $\mathcal{O}(n^{3/2})$ [CK12], which draws a 4-planar graph orthogonally with the minimum number of edge bends, while preserving a given planar representation (*combinatorial embedding*[2]) of the graph. Tamassia's algorithm is based on a reduction to a minimum-cost flow problem in a network and includes constraints respecting upper bounds on the number of edge bends [Tam90, TDBB88]. The drawings produced require an area that is in $\mathcal{O}(n^2)$, which is worst-case optimal [TDBB88]. To be more exact, the grid size is bounded by $(n+1) \times (n+1)$, as Biedl [Bie96] points out. Heretofore, Storer conjectured the problem to be NP-hard and gave three heuristics that produce drawings with $2n+4$ bends, if the graph is biconnected, and $2.4n+4$ bends otherwise. The area required for the drawings is in $\mathcal{O}(n^2)$. While no detailed time complexity analysis is given, the algorithms appear to run in $\mathcal{O}(n^3)$ time [Sto84, Tam90]. Tamassia and Tollis [TT89] then improve those results. They give a linear-time algorithm that constructs grid embeddings with at most $2n+4$ edge bends for 2-connected graphs, and $2.4n+2$ bends otherwise. They additionally show that edge lengths are in $\mathcal{O}(n)$, that every edge bends at most four times and, for 2-connected graphs, that all but two edges bend at most twice. The area requirement of the embeddings is in $\mathcal{O}(n^2)$. With reference to a work by Valiant [Val81], the authors point out that the bounds on the area requirement and length of the edges are both asympotically worst-case optimal. Biedl and Kant [BK98] also give a linear-time algorithm, yet with better bounds regarding area requirement and edge bends. Their embeddings occupy a $n \times n$ grid using $2n+2$ edge bends at most; each edge bends at most twice (unless the given graph is the octahedron). Also, their algorithm handles both planar and non-planar graphs as well as non-biconnected graphs. The main idea is to construct orthogonal drawings of the biconnected components of the graph using the *st-ordering*, and then merging those drawings into one orthogonal drawing of the entire graph. Given a biconnected graph $G = (V, E)$ and $s \neq t \in V$, an ordering $s = v_1, v_2, \ldots, v_n = t$ of the vertices of G is called an *st*-ordering, if for all vertices v_j, $1 < j < n$, there exist $1 \leqslant i < j < k \leqslant n$ such that $\{v_i, v_j\}, \{v_j, v_k\} \in E$ [LEC75]. Di Battista *et al.* [BLV93] show that, if a planar embedding is not given, the problem of finding orthogonal grid embeddings with the minimum number of edge bends is polynomially solvable for 3-planar biconnected graphs, and Tamassia shows that it becomes NP-complete for 4-planar graphs [GT94].

[2] A combinatorial embedding of a graph can be defined by giving, for each vertex v, a cyclic order of the edges incident to v.

Next, we consider results that are related to point-set embeddings, particularly *geodesic* point-set embeddings. Here, we are given a graph and, additionally, a set of points to which the vertices of the graph must be mapped. Furthermore, edges are required to be embedded as geodesics, that is, as shortest possible connections between their endpoints with respect to the underlying metric. This problem is referred to as Geodesic Point-Set Embeddability, or as simply Geodesic PSE [KKRW10]. Katz *et al.* [KKRW10] show that this problem is NP-hard on and off the orthogonal grid. In this thesis, we will study the complexity of Geodesic PSE on a non-standard grid, the *hexagonal grid* (or *triangular grid*), which is a grid formed by a triangular tiling of the plane. By further restricting the placement of vertices on the grid, we obtain a problem named Labeled Geodesic Point-Set Embeddability (Labeled Geodesic PSE). In this scenario, we are given a bijection between vertices of the graph and points on the grid. Katz *et al.* [KKRW10] show that this problem is NP-hard on the orthogonal grid. As for Geodesic PSE, we study the complexity of Labeled Geodesic PSE on the hexagonal grid.

Area Minimization.

An important optimization criterion is the area occupied by a drawing, which we also consider in this work. In general, minimizing the area needed by a drawing (that is entirely on the grid) corresponds to an NP-hard optimization problem [TDBB88]. For binary trees, the problem becomes solvable in polynomial time if one allows edges to leave the grid [SR83]. For input graphs having degree at most 4, Papakostas and Tollis [PT96] give a linear-time algorithm that produces drawings occupying an area of at most $0.76n^2$, while introducing at most $2n + 2$ edge bends, and each edge in such a drawing bends at most twice. The algorithm is based on forming and placing pairs of vertices of the graph. Another linear-time algorithm is presented for graphs having degree at most 3. The occupied area of a drawing is at most roughly $n^2/4$, and, if the graph is biconnected, the drawing has at most $\lfloor 3n/2 \rfloor + 3$ bends, which is optimal up to a factor of 2 (regardings edge bends). Moreover, Tamassia, Tollis and Storer show that 2-connected graphs can be embedded in an $n \times n$ grid using $2n + 4$ bends [Sto84, TT89]. A lower bound of $2n - 2$ edge bends for 2-connected planar graphs is given by Papakostas and Tollis [PT95]. Kant shows that triconnected graphs having maximum degree four can be embedded into an $n \times n$ grid with at most three bends per edge [Kan92a]. Even and Granot present an algorithm for orthogonal drawings of 4-connected graphs introducing at most three bends per edge [EG94]. Schäffter [Sch95] drops the limitation of producing crossing-free drawings and presents an $\mathcal{O}(n^2)$-time algorithm that introduces at most two bends per edge.

Further area optimization is done by Rahman, Nakano and Nishizeki [RNN96]. They give a grid drawing algorithm for plane graphs, that is, planar graphs that are already embedded in the plane, with lower bounds $W + H < n/2$, $WH \leqslant n^2/16$, where W is the width and H the height of the rectangular grid. The graphs they consider have degree exactly 3, except for vertices on the rectangular outer cycle, which have degree exactly 2. Schnyder [Sch89] proves that every planar graph has a straight-line embedding on a $2n - 5$ by $2n - 5$ grid, which can be computed in $\mathcal{O}(n \log n)$ time. An improved bound of $(2n - 4) \times (n - 2)$ is presented by de Fraysseix, Pach and Pollack [dFPP88, dFPP90]. The according algorithm uses $\mathcal{O}(n)$ space and runs in $\mathcal{O}(n \log n)$ time. They left open the problem of whether the time complexity can be improved. Chrobak and Payne [CP95] then improved the complexity to $\mathcal{O}(n)$, which is optimal, while the grid size remained the same. Later, Schnyder [Sch90] gave a $\mathcal{O}(n)$-time algorithm to further improve the bound on the grid size to $(n - 2) \times (n - 2)$. His algorithm can be implemented in parallel in $\mathcal{O}(\log n \log \log n)$ time using $n/\log n$ processors on a parallel random access machine

(PRAM), as is shown by Fürer, He, Kao and Raghavachari [FHKR92]. If randomization is used, the complexity improves to $\mathcal{O}(\log n)$ expected time. Regarding the lower bound on the grid size, de Fraysseix, Pach and Pollack [dFPP88, dFPP90] also show an example graph that requires a grid size of at least $(2n/3 - 1) \times (2n/3 - 1)$.

Xin He shows [He95] that it is possible to embed internally triangulated plane graphs without non-empty triangles[3] on a $W \times H$ grid, such that $W + H \leqslant n$, $W \leqslant (n+3)/2$ and $H \leqslant 2(n-1)/3$, if edges are allowed to run off the grid. The embedding can be computed in $\mathcal{O}(n)$ time. He [He97] obtains the same width and height bounds to straight-line embed 4-connected plane graphs on a $W \times H$ grid, again in linear time. A restriction on the given plane graph is that it must have at least four vertices on its external face.

Bend Minimization.

When dealing with grid structures, another important criterion for optimization is minimizing the number of edge bends needed for an embedding. As already noted above, there is a $\mathcal{O}(n^{3/2})$-time algorithm by Cornelsen and Karrenbauer [CK12] for embedding 4-planar graphs orthogonally introducing the minimum number of edge bends, given a planar representation of the graph in advance. Their algorithm is based on Tamassia's flow network [Tam87]. If, however, such a planar representation of the graph is not given as part of the input, the problem of finding the bend-minimum orthogonal grid embedding becomes NP-complete [GT94]. More precisely, Tamassia *et al.* [GT94] show that it is NP-hard to approximate the minimum number of bends in a planar orthogonal grid drawing with a $\mathcal{O}(n^{1-\varepsilon})$ error, for any $\varepsilon > 0$.

Otten and van Wijk [OVW78] introduce the *visibility representation*, where vertices are represented by horizontal and edges are represented by vertical line segments, such that each edge only touches the two vertex segments of its endpoints (meaning that edges do not cross any other vertex segments). They show that every planar graph admits a visibility representation. Rosenstiehl and Tarjan [RT86], as well as Tamassia and Tollis [TT86] then independently presented linear-time algorithms for constructing visibility representations of planar graphs, the latter authors using the *st*-ordering, and the former authors using the *bipolar orientation* (also called *st-orientation*), which is a certain orientation and partition of the edges of a graph into two sets. The *st*-orientation is a concept equivalent to the *st*-ordering, and they both can be obtained from each other. The grid size required for the representations is $(2n-5) \times (n-1)$ [Kan94] for both algorithms. Tamassia and Tollis [TT87] show that their aforementioned algorithm can be modified to compute grid embeddings having $\mathcal{O}(n)$ edge bends. Kant and He [KH94] improve the bound on the grid size to at most $(n-1) \times (n-1)$ for 4-connected planar graphs, and then Kant [Kan94] again improves this bound to at most $\lfloor (3n-6)/2 \rfloor \times (n-1)$ for general planar graphs. Lin, Lu and Sun [LLS05] point out that, by following the convention of placing the endpoints of vertex segments on grid points, one can easily see that any visibility representation of a graph can be made no higher than $n-1$. Therefore, it is only the width which is subject to further minimization. The authors use canonical orderings and Schnyder's *realizer*[4] [Sch89, Sch90] to further improve the bound on the width. For triangulated graphs, they improve the bound to $\lfloor (22n - 40)/15 \rfloor$. Further, if the graph has no internal node of degree 3, the bound improves to $\lfloor (4n - 9)/3 \rfloor$, and if the graph has no internal node of degree 5, the bound improves to $\lfloor (4n - 7)/3 \rfloor$. If the graph is 4-connected, the width is at most $n-1$, which matches the best known result of Kant and He [KH94]. Their according algorithm incrementally draws the graph using a greedy approach and works in linear time.

[3] Given a graph G, a non-empty triangle of G is a triangle of G containing some vertices in its interior.
[4] Similar to the *st*-ordering, a realizer is an orientation and partition of the edges into three sets.

Zhang and He [ZH05] give yet a better bound on the width. Using techniques by Lin, Lu and Sun [LLS05] and the simple visibility representation algorithm given by Rosenstiehl, Tarjan, Tamassia and Tollis [RT86, TT86], they show that any plane graph can be embedded on a grid whose width is bounded by $\lfloor(13n-24)/9\rfloor$. The problem of finding the most compact visibility representations of plane graphs remains open. Rosenstiehl and Tarjan [RT86] have conjectured the problem to be NP-hard.

Di Battista, Tamassia and Tollis [DBTT92] consider a special type of visibility representations, the *constrained visibility representation*, where prespecified edges are constrained to be vertically aligned, and give a $\mathcal{O}(n)$-time algorithm that computes a constrained visibility representation of a given planar graph. The resulting representations require $\mathcal{O}(n)$ area and use only integer coordinates. Concentrating on bend minimization, Fößmeier, Kant and Kaufmann [FKK97] introduce *2-visibility* representations or drawings of graphs, a more general type of visibility representation, where edges may run vertically and also horizontally. Furthermore, vertices are now represented by boxes—no longer as line segments—to distinguish them from edges. They present a polynomial-time algorithm to compute bend-minimum orthogonal drawings, where every edge bends at most once. They also give various upper and lower area bounds. Bose *et al.* [BDHS97] study further visibility representations with applications in via minimization in VLSI design. They find classes of graphs that allow to be drawn as *rectangle-visibility graphs*, which are graphs such that vertices are represented by rectangles and edges are realized as horizontal or vertical lines of sight.

Woods [Woo82] presents a $\mathcal{O}(n^2)$-time algorithm that produces drawings according to the *mixed standard*, where vertices are mapped to grid-line intersections and edges are drawn as polygonal chains that bend only on grid-line intersections. The drawings produced have $\mathcal{O}(n^2)$ edge bends and require $\mathcal{O}(n^2)$ area.

Biedl and Kaufmann [BK97] present algorithms to produce orthogonal drawings for arbitrary graphs. They consider *static* and *incremental* scenarios. In the static scenario, the input graph is given entirely in advance. The drawings produced by the according algorithm occupy at most a $(m/2+n/2)\times(m/2+n/2)$ grid, and every edge bends at most once, thus introducing at most m bends, where m is the number of edges in the graph. In the incremental scenario, the graph is given incrementally, that is, one node at a time, and a vertex needs to have a fixed placement before the next is given. Thus, once a vertex has been placed, it cannot be changed later. Here, the algorithm produces drawings on a grid whose size is at most $(m/2+n)\times(2m/3+n)$. Again, edges bend at most once, resulting in at most m overall bends. Next, using the canonical ordering, they consider planar and outerplanar graphs and give an algorithm that embeds a planar 3-connected graph on a grid of size $(m-n+1)\times\min\{m/2, m-n+1\}$ with $m-n$ edge bends. All their algorithms have linear time complexity.

Apart from the canonical ordering, another powerful tool for constructing visibility representations is the *regular edge labeling* (defined by Kant [KH97], for example). Kant and He [KH97] show that both are actually tightly connected and that a canonical ordering algorithm leads to a regular edge labeling algorithm. Regular edge labelings are not restricted to the construction of visibility representations, however. For example, they are also used by He [He95, He97]—the results were already described further above.

Canonical Orderings.

An important tool named the *canonical ordering*[5], due to de Fraysseix, Pach and Pollack [dFPP88, dFPP90], which is a special ordering on the vertices of a graph, is used by Kant [Kan96] to show that every triconnected planar graph admits a planar convex straight-line grid drawing on a $(2n-4) \times (n-2)$ grid, that every triconnected graph having degree at most 4 admits a planar orthogonal grid drawing on an $n \times n$ grid with at most $\lceil 3n/2 \rceil + 4$ edge bends, and if $n > 6$, then every edge bends at most twice. Moreover, Kant shows that every planar graph with maximum degree 3 admits a planar orthogonal grid drawing with at most $\lfloor n/2 \rfloor + 1$ bends on an $\lfloor n/2 \rfloor \times \lfloor n/2 \rfloor$ grid, and that every triconnected planar graph having degree at most d admits a planar polyline grid drawing on a $(2n-6) \times (3n-9)$ grid, with a minimum angle larger than $2/d$ radians and at most $5n-15$ edge bends. In the latter result, every edge bends at most three times. Kant and Chrobak [CK97] (again using the canonical ordering), and Schnyder and Trotter [ST92] independently show how to find a convex straight-line embedding of a 3-connected plane graph on a $(n-2) \times (n-2)$-sized grid in linear time. Chrobak and Nakano [CN95] present a linear-time algorithm that embeds a plane graph having at least three vertices into a grid of width at most $\lfloor 2(n-1)/3 \rfloor$. Using results by de Fraysseix, Pach and Pollack [dFPP88, dFPP90], they show that this bound is tight—each dimension of the grid needs to be at least $\lfloor 2(n-1)/3 \rfloor$, even if the other dimension is allowed to be unbounded [CN95].

De Fraysseix, Pach and Pollack [dFPP88, dFPP90] (already mentioned above) also use the canonical ordering in their $\mathcal{O}(n \log n)$-time algorithm to compute straight-line embeddings of maximal planar graphs. To find such an embedding, they first calculate a canonical ordering of the vertices and then they place vertices on the grid using the ordering. The algorithm uses a quite sophisticated data structure first introduced by Chazelle [Cha85] for rectangle range queries on a set of points. An algorithm improving the time bound to $\mathcal{O}(n)$ is given by Chrobak and Payne [CP95], also mentioned above. They embed vertices, one at a time, using the canonical ordering, at each stage adjusting the current partial embedding. Whenever a vertex v needs to be moved, they associate with v a set of other vertices that also move due to v. Their method does not require any complex data structures. Instead, they manage the information required during the algorithm, such that there is sufficient local information available to find a tentative embedding of each new vertex. Later, vertices are moved to their final positions using a tree-like data structure.

Another work based on canonical orderings is presented by Goodrich and Wagner [GW98]. They give an algorithm that draws planar graphs on a $\mathcal{O}(n) \times \mathcal{O}(n)$ grid using polylines that have at most two bends per edge and asympotitcally optimal worst-case angular resolution. More significantly, they show how to adapt this algorithm to draw any planar graph using cubic Bézier curves, with all vertices and control points[6] placed within an $\mathcal{O}(n) \times \mathcal{O}(n)$ integer grid such that the curved edges achieve a curvilinear analogue of good angular resolution. Both algorithms can be implemented to work in linear time. A result that is closely related to canonical orderings is presented by Gutwenger and Mutzel [GM98]; they refer to the canonical ordering as the *shelling order*. The authors present a linear-time algorithm based on ideas by Kant [Kan96], which computes a planar polyline grid drawing of a plane graph having degree at most d on a $(2n-5) \times (3n/2-7/2)$-sized grid with the smallest angle being greater than $2/d$. The drawing has at most $5n-15$ edge bends; every edge bends at most three times and has a length that is in $\mathcal{O}(n)$. Their results

[5] And a special case thereof, the *lmc-ordering*, short for *leftmost canonical ordering*.

[6] Four control points parametrically define a cubic Beziér curve.

significantly improve those presented earlier by Kant [Kan96], Kant and Chrobak [CK97], and de Fraysseix, Pach and Pollack [dFPP88, dFPP90], especially regarding angular resolution. According to Gutwenger and Mutzel, the minimum size of angles in those previous results is in $\Omega(1/n^2)$. Badent *et al.* [BBC11] note that the algorithm by Kant to compute canonical orderings is neither easy to code nor is its correctness easily understood [BBC11]. Therefore, the authors present an easier approach which computes a uniquely determined *lmc*-ordering, which they introduce as the *leftist canonical ordering*.

3. The Honeycomb Grid

In this chapter, we present algorithms that are based on the *honeycomb grid*, that is, a regular arrangement of interconnected, regular hexagons in the plane (see Figure 3.1). To the best of our knowledge, there are no known results with respect to the honeycomb grid so far. As such, our work introduces a new area of research.

In Section 3.1, we consider so-called *h-v drawings* of trees and present a simple algorithm that draws complete binary trees on the honeycomb grid according to the h-v drawing convention. Next, we will adopt an algorithmic idea by Crescenzi *et al.* [CP97] for drawing area-optimal h-v drawings of complete binary trees on the orthogonal grid, such that it can be used to create h-v drawings on the honeycomb grid, even though the resulting drawings are no longer area-optimal.

Section 3.2 thereafter considers what we introduce as *sector drawings* of trees having degree at most 3. We present an algorithm that produces sector drawings of trees whose every inner node has degree exactly 3. We extend this algorithm to one that produces sector drawings of general trees having degree at most 3.

3.1 h-v Drawings of Complete Binary Trees

This section deals with h-v drawings of complete binary trees. After having introduced the notion of h-v drawings on the honeycomb grid, we present a simple idea to compute such drawings of complete binary trees without worrying about the area the resulting trees occupy. We then quickly refer to an algorithm introduced by Crescenzi *et al.* [CP97] in order to achieve drawings that are more area-efficient.

We exactly adopt the definition of an *h-v drawing* given by Crescenzi *et al.* [CP97].

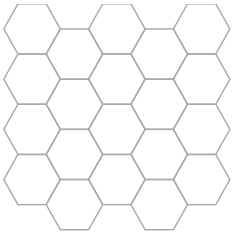


Figure 3.1: The honeycomb grid.

Definition 3.1.1 (h-v drawing [CP97]**).** An *h-v drawing* of a binary tree t is a drawing of t such that:

1. Nodes are points with integer coordinates.
2. Each edge is either a rightward-horizontal or downward-vertical straight-line segment from a node to one of its children (that is, an h-v drawing is not strictly upward).
3. Edges do not intersect.
4. If t_1 and t_2 are immediate subtrees of a node u, the enclosing rectangles of the drawings of t_1 and t_2 are disjoint.

See Figures 3.2a–3.2d for examples. Note that this definition refers to drawings on the *orthogonal grid*, i.e., a grid consisting of regularly arranged squares, such that single line segments are drawn only horizontally and vertically.

In the following, let the x- and y-axis run horizontally and vertically, respectively. We would like to port the concept of h-v drawings to the honeycomb grid, and then develop algorithms that produce such drawings.

To do so, we are going to map horizontal line segments of length 1 on the orthogonal grid to certain line segments having length 4 on the honeycomb grid. That line segment on the honeycomb grid, from left to right, consists of a horizontal line segment of length 1, followed by a diagonal upward-right line segment of length 1, followed again by a horizontal line segment of length 1, and finally followed by a diagonal right downward line segment of length 1. The two end points of the composed line segment then share the same y-coordinate.

The vertical line segments of length 1 on the orthogonal grid are mapped in a similar way: The according line segments of length 4 on the honeycomb grid, from left to right, consist of only two different types of line segments: Left and right downward ones, each of them having length 1. The order of the four line segments that form the composed segment is then: Left downward, right downward, left downward, right downward. The two end points of the composed line segment share the same x-coordinate. Figure 3.3a and 3.3b illustrate the two mappings.

The mapping as defined allows for only certain grid points and line segments of the honeycomb grid to be used as nodes and edges of a tree, depending on which grid point we draw the first node of a tree. The grid points are given by the lower left corners of the hexagons of the honeycomb grid. Other grid points (i.e., corners of a hexagon) cannot be used for tree nodes, according to our definition. Valid line segments of the grid that are possible candidates for later tree edges also result from the definition of the first node

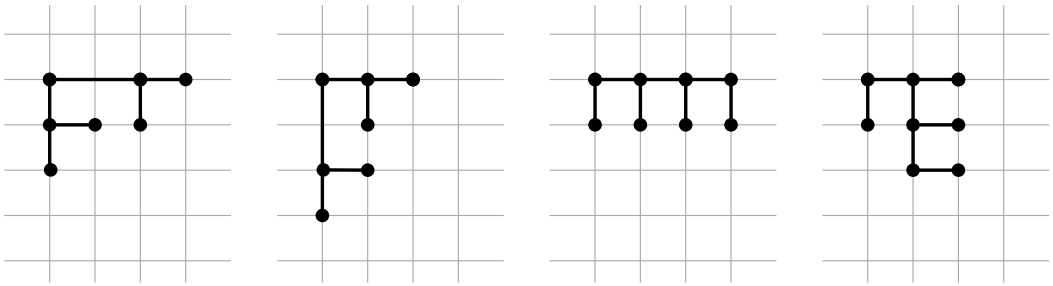
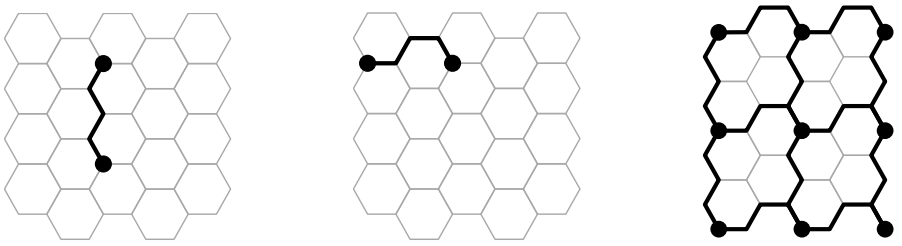


Figure 3.2: A few examples of h-v drawings on the orthogonal grid.



(a) (b) (c)

Figure 3.3: A vertical (a) and horizontal edge (b) according to our definition. The restricted grid (c) when considering only nodes that are placed in the lower left corner of a hexagon. Since edges have length 4, not all lower left corners of the hexagons are valid candidates for tree nodes. Note that not even all line segments of the restricted grid may be used for later tree edges, since we do not allow edge crossings in our h-v drawing.

drawn on the grid. Since every edge according to our definition has length 4, not all lower left corners of the hexagons can be used for later tree nodes.

By considering only those valid grid points, we restrict the entire honeycomb grid to a grid that looks as shown in Figure 3.3c. Note that not all line segments of the restricted grid may be used for later tree edges, because we do not allow edges to cross each other. Moreover, we cannot draw cyclic graphs in a planar way (that is, with no edges crossing) when using this kind of restriction, whereas in the orthogonal case, any planar graph having degree at most 4 can be drawn in a planar way [BK98].

We denote by $\mathcal{C}$ a complete binary tree, and by $\mathcal{C}_h$ we denote a complete binary tree having height h. The number of nodes of a tree will be referred to by n. For some $h \geqslant 0$, it is easy to see that $n = 2^{h+1} - 1$.

Theorem 3.1.1. *Algorithm 1 computes an h-v drawing (embedding) of $\mathcal{C}_h$ on the honeycomb grid for some desired $h \geqslant 0$.*

Proof. For $h = 0$ the algorithm returns a single node, i.e., a complete binary tree of height 0. Given a complete binary tree t of height h, the `extend` function (5) builds a complete binary tree having height $h + 1$ in the following way: First, it creates a new root at coordinates $(0, 0)$. Then, to build its two subtrees, a copy of the current tree t is created. The existing tree t is placed four line segments (thus forming a valid edge) below the new root, its copy is placed $4 \cdot 2^h$ line segments to the right of the new root. Finally, the algorithm connects the new root with its two subtrees, using one rightward and one downward edge, resulting in $\mathcal{C}_{h+1}$. □

Since the occupied area of a drawing is a criterion of interest for us, we need to define first the notion of "area" in the context of this work. Here and in the following, the occupied area is measured by the area of the smallest surrounding rectangle of a drawing.

Theorem 3.1.2. *For some given $h \geqslant 0$, the h-v drawing (embedding) computed by Algorithm 1 occupies an area on the honeycomb grid that is in $\mathcal{O}(n \log n)$, where n denotes the number of nodes of $\mathcal{C}_h$[1].*

[1] Recall that $n = 2^{h+1} - 1$.

Proof. Let W_i denote the width of an embedding of $\mathcal{C}_i$, an let H_i denote its height (that is, length in units along the x-axis, and length in units along the y-axis, respectively). By construction, since every downward edge has length 4, it follows that $H_i = 4i$. Moreover, and again by construction, we have $W_i = 2 \cdot W_{i-1} + 4$. Solving this recurrence yields $W_i = 4 \cdot (2^i - 1)$. The area $W_i H_i$ of the embedding is thus equal to $4 \cdot (2^i - 1) \cdot 4i$. Given a complete binary tree consisting of n nodes, using the fact that $i = \lfloor \log_2(n) \rfloor$ results in an area of

$$4 \cdot (2^i - 1) \cdot 4i = 16 \cdot (2^{\lfloor \log_2(n) \rfloor} - 1) \cdot \lfloor \log_2(n) \rfloor \in \mathcal{O}(n \log n).$$

□

Theorem 3.1.3. *An h-v embedding on the orthogonal grid that occupies an area of A induces an h-v embedding on the honeycomb grid that occupies an area of $16A$.*

Proof. This follows immediately from the previous theorem. □

This bound can trivially be improved to $8A$ by shortening all vertical edges by two line segments. More precisely, we remove from the four line segments of a vertical edge the lower two segments to obtain a vertical edge of length 2.

Corollary 3.1.1. *An h-v embedding on the orthogonal grid that occupies an area of A induces an h-v embedding on the honeycomb grid that occupies an area of $8A$.*

Note that the resulting tree embedding is not area-optimal. On the orthogonal grid, complete binary trees can be drawn on $\mathcal{O}(n)$ area [BBB+09, CP97]. An algorithm for area-optimal embeddings on the orthogonal grid is given by Crescenzi *et al.* [CP97]. It can also be used for h-v drawings on the restricted honeycomb grid (recall Figure 3.3c) to achieve area-optimal embeddings, since edges on the restricted honeycomb grid are four times the length of according edges on the orthogonal grid; a constant factor that can be neglected asymptotically. The only difference lies in the underlying grid structure. Crescenzi *et al.* distinguish area-optimal and "useful" h-v drawings in their approach. "Useful" h-v drawings are not area-optimal, but interestingly, they are needed in their algorithm to achieve area-optimality.

Procedure hvDrawingOfCBT

Input: Integer $h \geqslant 0$ denoting the desired height of the tree
Output: Complete binary tree t having height h

```
1 begin
2     t' ← createNode(0, 0)
3     h' ← 0
4     while h' < h do
5         t' ← extend(t', h')
6         h' ← h' + 1
7     t ← t'
```

Procedure extend

Input: Integer $h \geqslant 0$, h-v drawing t of $\mathcal{C}_h$
Output: h-v drawing t' of $\mathcal{C}_{h+1}$

```
1 begin
2     root_new ← createNode(0, 0)
3     t_rightChild
      ← moveRight(copy(t), 4 · 2^h)
4     t_leftChild ← moveDown(t, 4)
5     t' ← connect(root_new, t_leftChild, t_rightChild)
```

Algorithm 1: Creating an h-v drawing of a complete binary tree having height $h \geqslant 0$.

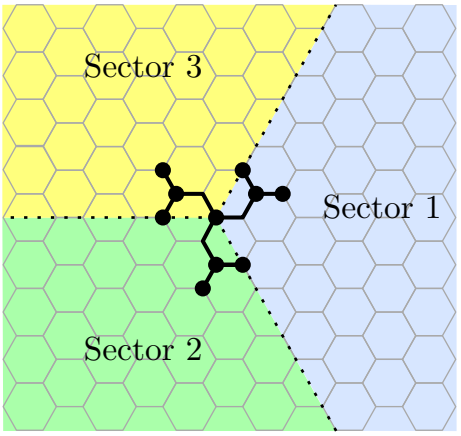


Figure 3.4: A sector drawing of a complete tree of height 2 on the honeycomb grid.

3.2 Sector Drawings of Trees

In this section we consider a type of drawings that we refer to as *sector drawings* on the honeycomb grid. The idea is to partition the honeycomb grid into three sectors of equal size. The partitioning of the honeycomb grid will originate in the root node of some given tree, and the root node will also define the sector boundaries. An exact definition of sector drawings is given further below.

After having introduced sector drawings, we will present an algorithm to draw complete binary trees according to the sector drawing convention. Then, we will generalize the algorithm to one that produces sector drawings of arbitrary complete trees having degree at most 3.

To start with, we will give a definition of sector drawings.

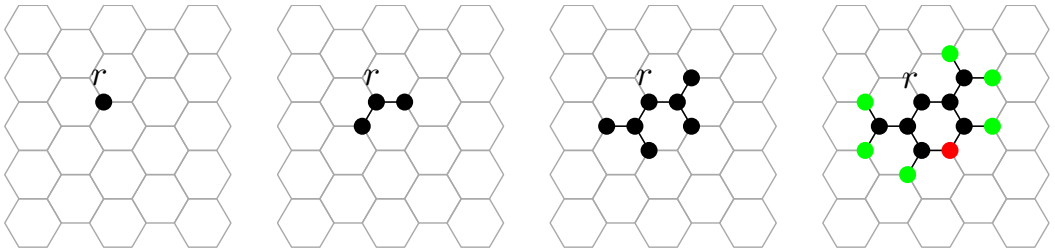
Definition 3.2.1 (Sector Drawing). A *sector drawing* of a tree t is a drawing of t such that:

1. Nodes are placed on intersections of grid lines.
2. Edges do not intersect.
3. Edges do not cross sector borders.
4. If t_1 and t_2 are immediate subtrees of a node u, then t_1 and t_2 are disjoint, i.e., they are placed in different sectors of u.

Note that nodes are not forbidden to be placed on the sector borders. The main idea behind sector drawings is to guarantee that there is enough space for drawing a given tree in a planar way when using a top-down approach, i.e., when starting the drawing at the tree's root node.[2] Figure 3.4 shows the three sectors that make up the grid, as well as a sample sector drawing of a tree.

Since sector drawings are planar embeddings of trees, we may not have edge crossings in our drawing. Furthermore, edges may not cross sector borders. It is therefore important to guarantee that there is enough free space for the drawing on the honeycomb grid, such that a planar embedding is possible. The minimum length of the edges thus depends on the height of the tree. For instance, it is not possible to find a sector drawing of a complete binary tree having height 3 or more, such that all edges have length 1.

[2]Actually, we will not require a root node to be given, but we will root the tree at some arbitrarily chosen node.



(a) $h = 0$. (b) $h = 1$. (c) $h = 2$. (d) $h = 3$.

Figure 3.5: For $h \leqslant 2$, there exist sector drawings of $\mathcal{C}_h$, such that all edges have length 1. For $h = 3$ (and thus also for $h \geqslant 3$), this is no longer the case. Root nodes are named r, the green and red nodes are level-3 nodes. Green nodes are nodes that can be drawn according to our sector drawing definition. The red node is conflicting, since it is the child of two parent nodes.

Lemma 3.2.1 For $h > 2$, a sector drawing of $\mathcal{C}_h$ does not exist, if all edges are required to have length 1.

Proof. For $h \leqslant 2$, this is possible, as Figure 3.5c shows. For $h > 2$, consider the following: Generally, given a complete binary tree $\mathcal{C}_h$, one can always find a complete binary tree in $\mathcal{C}_h$ that has height h', $0 \leqslant h' \leqslant h$, by simply removing the $h - h'$ lowest levels in $\mathcal{C}_h$ and the corresponding edges. Figure 3.5d shows that already for $h = 3$ a sector drawing does no longer exist. Consequently, since we can always find a complete binary tree in $\mathcal{C}_h$ having height $h' = 3 \leqslant h$, no such sector drawing can exist for $h > 2$. □

In other words, for $h > 2$, every embedding of $\mathcal{C}_h$ contains a "broken" embedding of $\mathcal{C}_3$ and thus, the embedding of $\mathcal{C}_h$ itself cannot be a sector drawing.

In the following, we will describe an inductive idea to find sector drawings of trees having degree at most 3. Considering such a tree of height h, we will use edges whose lengths decrease exponentially when traversing the tree top-down, starting with edges of length 2^h that connect the tree's root with its children, moving on to edges of length 2^{h-i}, $2 < i < h$, which connect inner nodes on levels i and $i+1$, ending with edges of length 2, that connect inner nodes on level h with the tree's leaves.

Let T be a rooted tree consisting of n nodes having degree at most 3, and let T_1 and T_2 be the subtrees of T (T_i may be empty). Let r be the root of T. We prove by induction that we can always find a drawing Γ of T such that the following three properties hold:

1. Γ is entirely contained in an equilateral parallelogram P having a vertex v,
2. r is placed inside P and can be connected to v using a path that is not crossed by Γ,
3. Γ is drawn with no edges crossing.

For $n = 1$, without loss of generality, consider the drawing shown in Figure 3.6, which obviously satisfies all three properties. For $n > 1$, by induction hypothesis, we can draw the subtrees T_1 and T_2 in a way such that the desired properties hold. Let Γ_1 and Γ_2 be the drawings of T_1 and T_2, respectively. Let P_1 and P_2 be the parallelograms that contain Γ_1 and Γ_2, respectively. In order to draw T, consider Figure 3.7. We "glue together" P_1 and P_2 as shown, which allows us to find a point we can map the root node r to. Let Γ be the drawing consisting of Γ_1, Γ_2, r and the two edges that connect Γ_1 and Γ_2 with

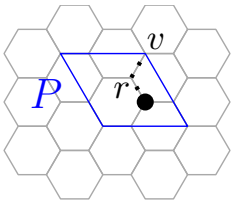


Figure 3.6: The inductive construction of sector drawings—base case.

their parent node r. Then, we can derive a new parallelogram P that contains Γ, that is, P is given by the smallest equilateral parallelogram that properly contains P_1 and P_2 and has the vertex v, as shown in the figure. The way we "glue together" the parallelograms ensures that we can find a path p on the grid along a straight-line segment that is not crossed by Γ: Again due to the induction hypothesis, the right side of P_1 is not intersected by Γ_1. Thus, we let p start in r and end in v. Since the right side of P_1 is not intersected by Γ_1, and since Γ_2 does not intersect the right side of P_1 either, we let p move along the straight-line connection between r and v whenever possible, and in all other cases, we let p move rightwards of the straight-line connection, as can be seen in the figure.

The way we arrange P_1 and P_2 makes sure that Γ can be drawn without edges crossing. Also, it is due to this arrangement, that the equilateral parallelogram P properly contains Γ and that we can identify a vertex v of P suited for our further construction. In other words, the first invariant holds. Moreover, as described above, r is placed inside P, and r can be connected to v via a path p that is not crossed by Γ. Thus, the invariants 2. and 3. from above are fulfilled as well, which concludes the inductive construction.

The idea induces an algorithm, say $\mathcal{A}$, that finds a sector drawing of a tree having degree at most 3. We will now study the area occupied by such an embedding.

Theorem 3.2.1. *The sector drawing computed by algorithm $\mathcal{A}$ occupies an area on the honeycomb grid that is in $\mathcal{O}(n^2)$.*

Proof. Let A be the area as shown in Figure 3.8. Recall that we have defined the area of a drawing to be the area of the smallest surrounding rectangle of that drawing. The occupied area of a sector drawing of a given tree having height h is at most the size of a sector drawing of $\mathcal{C}_h$. The area can be calculated as shown in Figure 3.8: Let W be the width, and let H be the height of the rectangle shown in blue color. To measure the rectangle's area $A = WH$, we will investigate W and H separately.

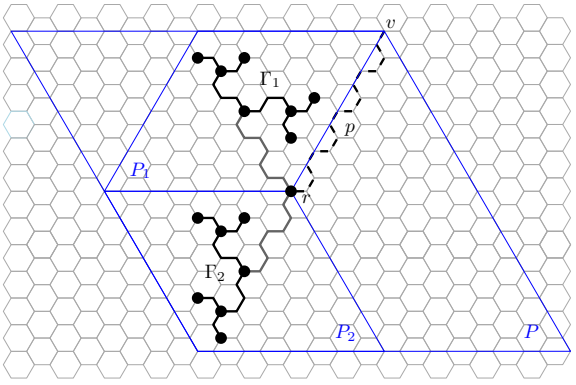


Figure 3.7: The inductive construction of sector drawings—inductive step.

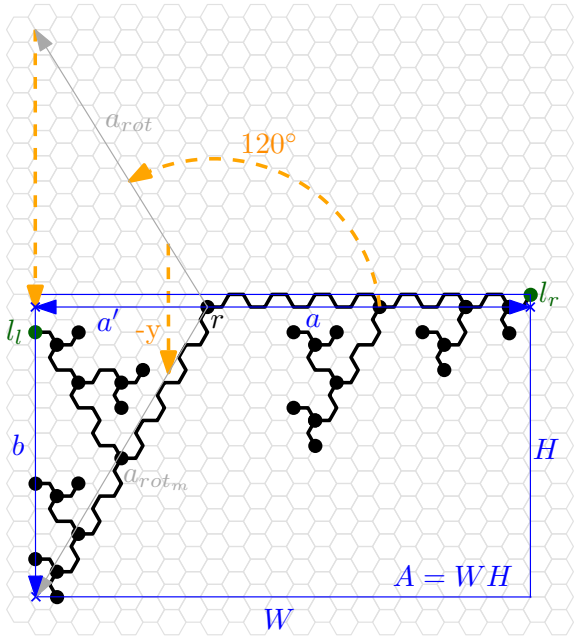


Figure 3.8: Determining the area $A = WH$ occupied by a sector drawing.

Width W:

To start with, we are interested in the length $\|a\|$ of vector a which is shown in the figure. Consider the path p that starts in r and ends in the rightmost leaf l_r. We can measure $\|a\|$ by regarding each line segment of p as a vector, projecting these vectors onto the x-axis and then adding up the lengths of the resulting vectors. By defining a horizontal line segment to have length 1 and some trigonometry we obtain a length of $\cos(\pi/3) = 1/2$ for non-horizontal line segments. The path p consists of $\sum_{i=1}^{h} 2^i$ line segments, of which there are

$$\frac{1}{2}\sum_{i=2}^{h} 2^i + 1 = 2^h - 1$$

horizontal and $2^h - 1$ non-horizontal segments. Thus, a is of the form

$$a = \begin{pmatrix} 1 \cdot \left(2^h - 1\right) + \frac{1}{2} \cdot \left(2^h - 1\right) \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{3}{2}\left(2^h - 1\right) \\ 0 \end{pmatrix},$$

and the x-component of a is also the length of that vector. Next, we consider the vector a' as shown in the figure, whose x-component is equal to the x-coordinate of some leftmost leaf l_l. Then, the width W is obviously equal to $\|a\| + \|a'\|$. We need to find a': First, we rotate a by 120° counterclockwise. This is equals a rotation by $2\pi/3$ radians. Formally, we calculate

$$a_{rot} = \begin{pmatrix} \cos\frac{2\pi}{3} & -\sin\frac{2\pi}{3} \\ \sin\frac{2\pi}{3} & \cos\frac{2\pi}{3} \end{pmatrix} \cdot a = \begin{pmatrix} -\frac{3}{4}\left(2^h - 1\right) \\ \frac{3\sqrt{3}}{4}\left(2^h - 1\right) \end{pmatrix}.$$

Then, we project the resulting vector a_{rot} onto the x-axis to obtain vector a'. This can be achieved by calculating

$$a' = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \cdot a_{rot} = \begin{pmatrix} -\frac{3}{4}\left(2^h - 1\right) \\ 0 \end{pmatrix}.$$

Now we can finally calculate W as

$$W = \|a\| + \|a'\| = \frac{3}{2}\left(2^h - 1\right) + \frac{3}{4}\left(2^h - 1\right) = \frac{9}{4}\left(2^h - 1\right).$$

Height H:

To find H, we first reflect the vector a_{rot} across the x-axis to obtain a_{rot_m}:

$$a_{rot_m} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \cdot a_{rot} = \begin{pmatrix} -\frac{3}{4}\left(2^h - 1\right) \\ -\frac{3\sqrt{3}}{4}\left(2^h - 1\right) \end{pmatrix}.$$

It is now easy to see that b can be written as

$$b = a_{rot_m} - a' = \begin{pmatrix} 0 \\ -\frac{3\sqrt{3}}{4}\left(2^h - 1\right) \end{pmatrix}.$$

Clearly, the length of vector b equals H, i.e.,

$$H = \|b\| = \frac{3\sqrt{3}}{4}\left(2^h - 1\right).$$

We can calculate the occupied area A of the sector drawing as

$$\begin{aligned} A = WH &= (\|a\| + \|a'\|) \cdot \|b\| \\ &= \left(\frac{9}{4}\left(2^h - 1\right)\right) \cdot \left(\frac{3\sqrt{3}}{4}\left(2^h - 1\right)\right) \\ &= \frac{27\sqrt{3}}{16}\left(2^h - 1\right)^2. \end{aligned}$$

Finally, substituting $h = \log_2(n-1) - 1$ yields

$$\begin{aligned} A &= \frac{27\sqrt{3}}{16}\left(2^h - 1\right)^2 \\ &= \frac{27\sqrt{3}}{16}\left(2^{\log_2(n-1)-1} - 1\right)^2 \\ &= \frac{27\sqrt{3}}{16}\left(\frac{n-1}{2} - 1\right)^2 \\ &= \frac{27\sqrt{3}}{16}\left(\frac{n^2}{4} - \frac{3n}{2} + \frac{9}{4}\right) \in \mathcal{O}(n^2). \end{aligned}$$

□

For $n \geqslant 3$, the latter term is smaller than $0.731n^2$. The area can be reduced by choosing edges that are only half as long, i.e., starting at the root node, edges have length 2^{h-1} instead of length 2^h, and so on. Edges that are incident to leaves then have length 1 instead of 2. Asymptotically, however, the area occupied by such a sector drawing is still quadratic; the difference lies in constant factors.

The algorithm above can be extended to one that draws a complete tree of height h, whose every inner node has degree exactly 3, by simply drawing a third child in the remaining free sector of the root node r. Figure 3.9 depicts such a sector drawing of a complete tree having height 5.

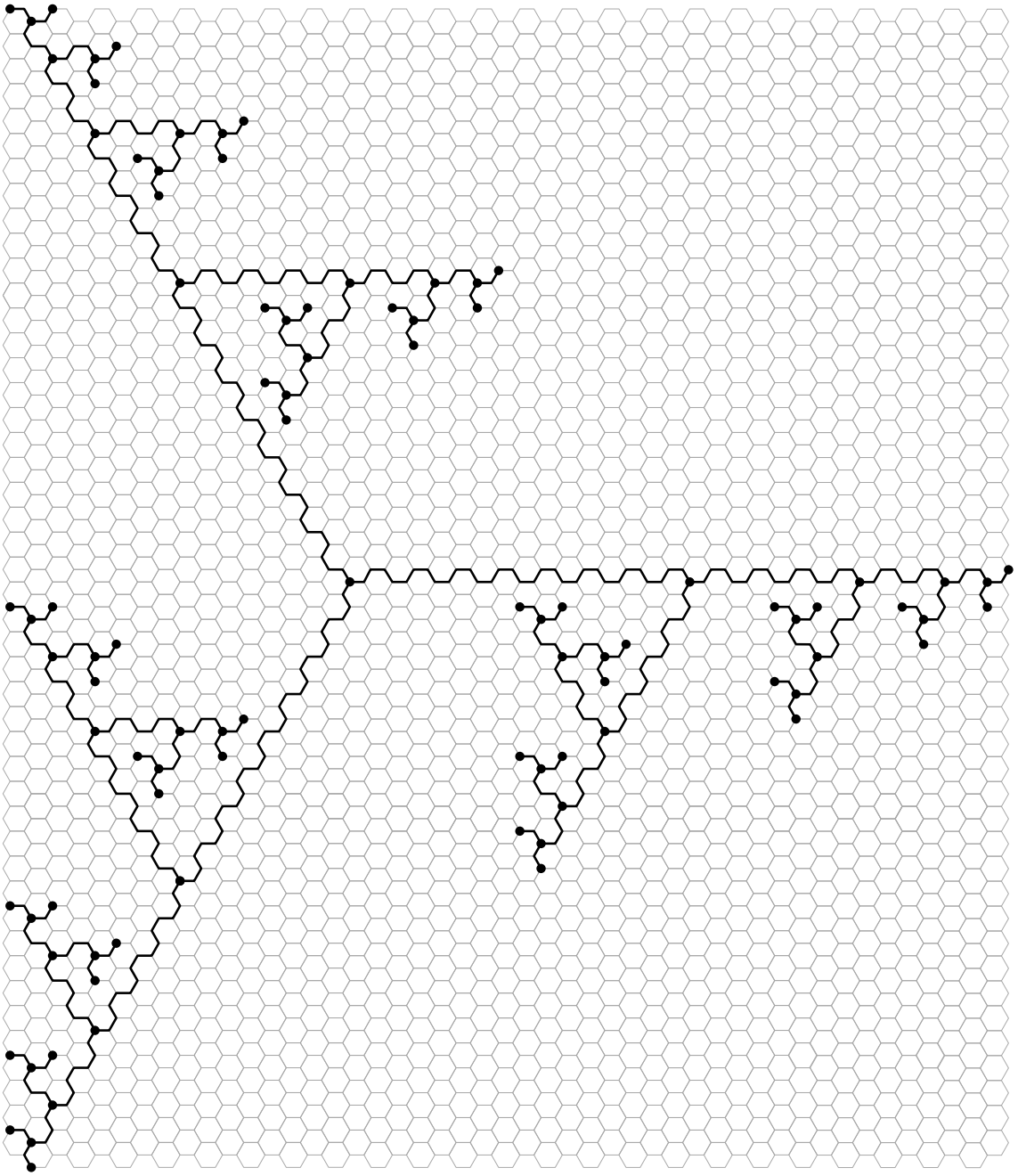


Figure 3.9: A sector drawing of a complete tree of height 5.

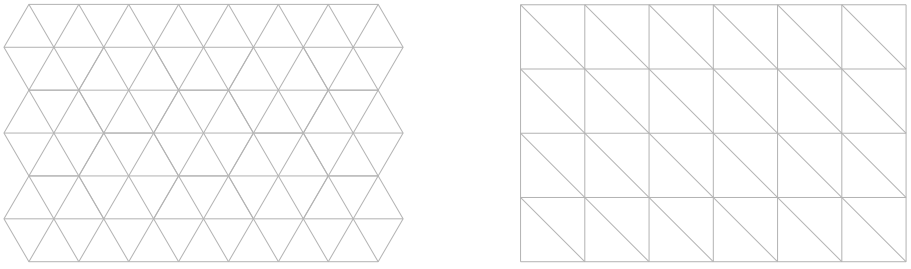
4. The Hexagonal Grid

In this chapter, we focus on algorithms and complexity regarding the *hexagonal grid*, also referred to as the *triangular grid*. The hexagonal grid can be visualized in several reasonable ways: Figure 4.1a shows the grid as a regular arrangement of interconnected hexagons, while Figure 4.1b shows that the hexagonal grid can be regarded as an extension of the orthogonal grid to which one type of diagonals is added. As Bachmeier *et al.* [BBB+09] do, we refer to the latter grid as the *sheared grid*. Both are equivalent and can be obtained from each other via shearing operations. We shall switch between the two representations whenever suited.

Figures 4.2a and 4.2b show that is easy to apply algorithms that work on the orthogonal grid or honeycomb grid to the hexagonal grid as well, since the latter contains both the orthogonal and honeycomb grids. However, it is not clear whether it is possible to get superior results resulting from more degrees of freedom on the grid.

Research done so far yields only few results regarding this type of (non-standard) grid. Kant [Kan92b] presents a linear-time algorithm to draw triconnected planar graphs on a linear-sized hexagonal grid such that at most one edge bends. Regarding angular resolution as an optimization criterion, Kant shows how to draw planar graphs of degree at most 3 in a planar way (that is, such that no edges cross) using straight lines only, such that the minimum angle is at least $\pi/6$.

Bachmaier *et al.* [BBB+09] also consider straight-line drawings motivated by the fact that the hexagonal grid allows visually appealing drawings of up to 5-ary trees. They restrict



(a) The hexagonal grid. (b) The sheared (hexagonal) grid.

Figure 4.1: Two different representations of the same grid.

their graphs specifically to ternary trees and give area bounds for different drawing styles they consider. Their work also includes an interesting result regarding computational complexity: It is NP-hard to draw unordered penta trees on a hexagonal grid within minimal area or with minimal edge length. And more generally, drawing trees on a hexagonal grid within a prescribed area or with unit length edges is NP-hard.

Tamassia [Tam87] presents an $\mathcal{O}(n^2 \log n)$-time[1] algorithm to obtain bend-minimum orthogonal drawings of planar graphs on the orthogonal grid and shows that the algorithm can be extended to work on the hexagonal grid.

Aziza and Biedl [AB04] present an algorithm that draws all graphs having degree at most 6 on the hexagonal grid using $3.5n + 3.5$ edge bends, possibly occupying exponential space, that is, area, on the grid. If more bends are allowed, the occupied area is within quadratic bounds.

In the following, we consider algorithms for geodesic drawings of graphs on the hexagonal grid. That is, given a graph $G = (V, E)$, we study drawings such that vertices are mapped to intersections of the grid lines and edges exclusively run along the grid lines. Moreover, each edge $e = uv \in E$ is realized as a shortest possible connection between u and v. This problem is referred to as Geodesic Point-Set Embeddability (Geodesic PSE) and was first introduced by Katz *et al.* [KKRW10].

In Section 4.1, we introduce definitions and show fundamental properties of shortest path connections (*geodesics*), which are the basis for the following sections.

Section 4.2 deals with the computional complexity of Geodesic PSE with respect to the hexagonal grid. Katz *et al.* [KKRW10] showed that Geodesic PSE is NP-hard on the orthogonal grid, and we will show that it is NP-hard on the hexagonal grid as well using a similar proof.

Afterwards, in Section 4.3, we focus on the problem of Labeled Geodesic Point-Set Embeddability (Labeled Geodesic PSE). Here, in addition to the input graph G, we are given a mapping that determines the correspondence between vertices of G and points on the grid. As on the orthogonal grid [KKRW10], we will see that Labeled Geodesic PSE turns out to be NP-hard on the hexagonal grid. However, we will provide an algorithm that solves certain problem instances, which we will identify as *sparse*, efficiently.

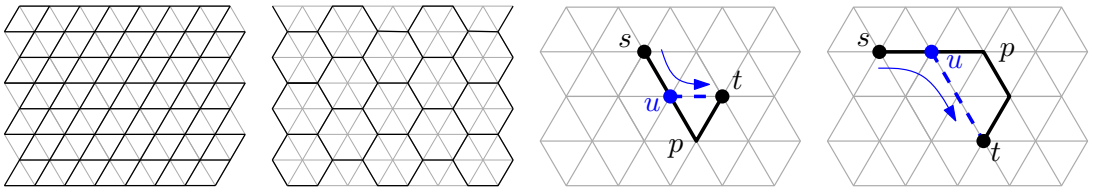
4.1 Basic Definitions and Properties

To begin with, we will provide a few basic definitions and show properties of geodesics, that is, shortest paths between two points, that will be helpful throughout this chapter. All points and connecting paths, representing vertices and edges of a graph, are assumed to be *on* the hexagonal grid.

A *point* $p = (p_x, p_y) \in \mathbb{R}^2$ consists of two coordinates p_x and p_y. In order to be able to speak about coordinates, we have choose a coordinate system for our grid, which is shown in Figure 4.3. We choose a natural, horizontal x-axis and have two possibilities left to pick a y-axis: from the upper left to the lower right, or from the lower left to the upper right. We choose the latter variant and define the y-axis to run from the lower left to the upper right.

Let u and v be two points on the grid. A *path*, say p, is a connection between u and v consisting of line segments of the grid. By $|p|$ we denote the length of that path, which is determined by the sum of the (Euclidean) lengths of the line segments occupied by p.

[1] n denotes the number of vertices of the graph.



(a) Hexagonal and sheared orthogonal grids.

(b) Hexagonal and honeycomb grids.

(c) Geodesics may not have 60° edge bends.

(d) Geodesics may not bend into the same direction more than once in a row.

Figure 4.2: (a), (b): Both the orthogonal and honeycomb grids can be embedded into the hexagonal grid. The orthogonal grid can be embedded by shearing it to the right by 45°, the honeycomb grid can be embedded by removing the interior line segments of the hexagonal grid. (c), (d): Geodesics on the grid may neither have 60° edge bends nor may they bend into the same direction more than once. In both cases, this is due to the triangle inequality—a property of the metric on the hexagonal grid.

A *geodesic* (*path*) between u and v is determined by a shortest path on the grid. Using this definition we obtain a metric $d : \mathbb{R}^2 \to \mathbb{R}$ on the hexagonal grid, where the distance $d(u, v)$ between u and v is defined as the sum of the (Euclidean) lengths of the line segments on the geodesic path from u to v.

A *geodesic embedding* of a graph G is an embedding on the hexagonal grid such that all edges are mapped to geodesics.

Lemma 4.1.1 Let p be a geodesic (path) between two points s and t. Then the following properties hold:

(i) p has no 60° edge bends.

(ii) p does not bend into the same direction more than once in a row.

Proof.

(i) Consider a path p between two nodes s and t that contains at least one 60° edge bend. Considering specifically the point on the grid where the edge turns by 60°, we can find two adjacent grid points—e.g. points u and t in Figure 4.2c—and use them to shorten the path to t. Clearly, p cannot be geodesic. More formally, this is due to the triangle inequality that holds for d. Per definition, p is a shortest path connecting s and t, and thus it minimizes the distance between s and t. Therefore, the distance equals $d(s, t)$, where d is the metric on the hexagonal grid for which the triangle inequality needs to hold.

(ii) Consider an s-t path p that, at some point u, bends twice into the same direction, as shown in Figure 4.2d. We can leave the path p towards t right before u, resulting

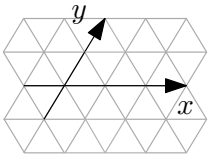


Figure 4.3: Choosing a coordinate system for the hexagonal grid.

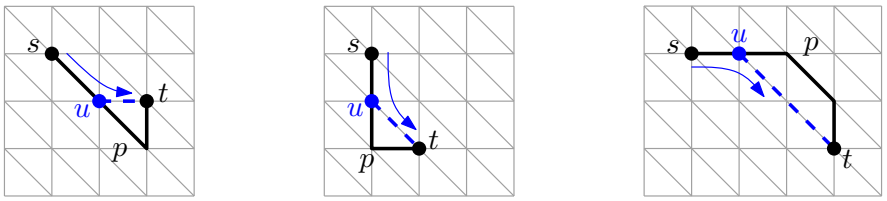


Figure 4.4: On the sheared grid, (a) geodesics must not have 45° edge bends and (b) downward geodesics must not have 90° edge bends (versus 60° on the hexagonal grid). Moreover, (c) geodesics may not bend into the same direction more than once, as is the case on the hexagonal grid. In all three cases, this is again due to the triangle inequality.

in a new s-t path that is shorter than p. Formally, this is again due to the triangle inequality. Thus, p cannot be a shortest path.

□

Similarly, switching our perspective to the sheared grid, geodesics must not have 45° edge bends and downward geodesics must not have 90° edge bends. The other property remains unchanged. See Figures 4.4a-4.4c for illustration.

We will now come to a further important finding. Let p be a point on the hexagonal grid. We denote by $p(x)$ or p_x the x-coordinate, and by $y(p)$ or p_y the y-coordinate of p. Consider the points s_1 and t and the set of all geodesics between them, shown in blue color (Figure 4.5). Since 60° edge bends are forbidden, all s_1-t geodesics lie on an orthogonal grid that is sheared to the right by 30°. We refer to such geodesics that go from some "lower left" point s_1 to some "upper right" point t, i.e., $x(s_1) \leqslant x(t)$ and $y(s_1) \leqslant y(t)$, as *upward geodesics*. Accordingly, consider points s_3 and t and the set of all geodesics between them, shown in red color. In this case, all s_3-t geodesics lie on an orthogonal grid that is sheared to the left by 30°. *Downward geodesics* are all those that go from some "upper left" point s to some "lower right" point t, i.e., $x(s) \geqslant x(t)$ and $y(s) \geqslant y(t)$. Note that a geodesic can be both downward and upward according to the definition. In general, whenever some point s is positioned left of or on the straight line l_l, all s-t geodesics run on the orthogonal grid sheared to the right. And whenever s is situated right of or on l_r, all s-t geodesics run on the orthogonal grid sheared to the left. Moreover, if s is positioned right of l_l *and* left of

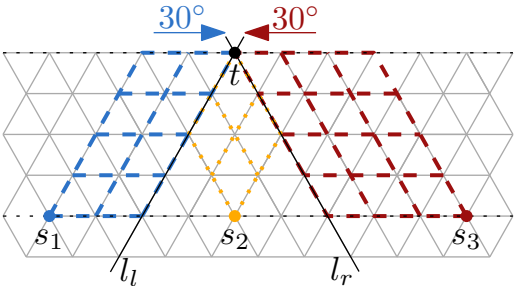


Figure 4.5: Upward geodesics between s_1 and t exclusively lie on an orthogonal grid sheared to the right by 30°, while downward geodesics connecting t and s_3 exlusively lie on an orthogonal grid sheared to the left by 30°. If some point s_2 is situated right of the line l_l, but left of l_r, then the s_2-t geodesics run on both type of grids, but never along horizontal line segments.

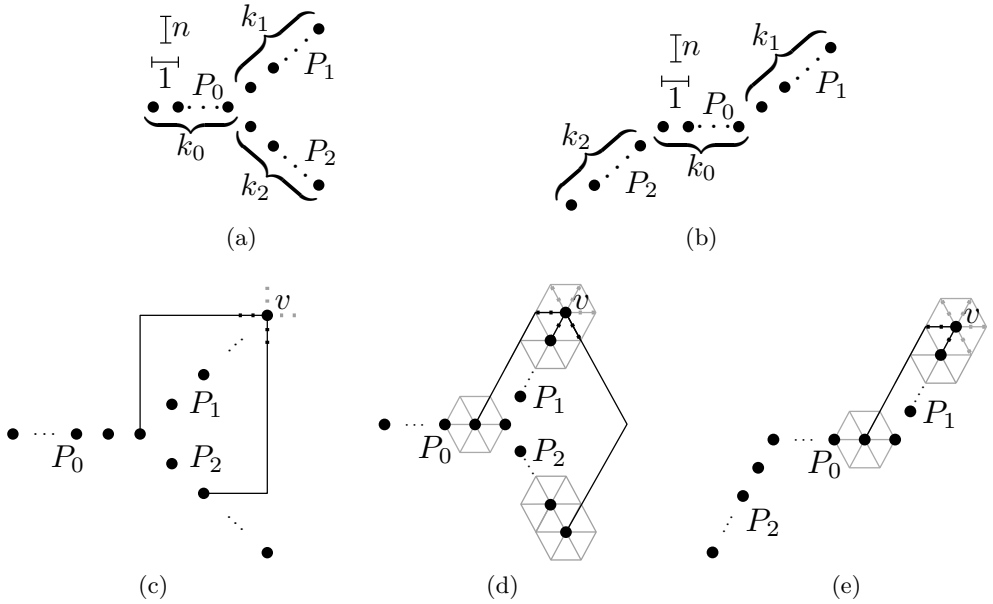


Figure 4.6: (a): The point set $P(k_0, k_1, k_2) = P_0 \cup P_1 \cup P_2$ used for the proof of NP-hardness on the orthogonal grid. (b): The modified point set used for our proof on the hexagonal grid. (c): At most two geodesics can go from v to any point in $P(k_0, k_1, k_2)$. (d): The situation changes on the hexagonal grid, since v can now connect to up to three points in $P(k_0, k_1, k_2)$ using a geodesic path. (e): Modifying P_2 accordingly again results in at most two geodesics going from v to any point in $P(k_0, k_1, k_2)$. A third geodesic is impossible due to the modification of P_2.

l_r, all *s-t* geodesics run on *both* types of grids, but never across horizontal line segments, since this would imply 60° edge bends, which are forbidden according to Lemma 4.1.1. Figure 4.5 demonstrates the three types of geodesics on either or both types of sheared orthogonal grids.

4.2 Geodesic Point-Set Embeddability

In the following, we draw our attention to the problem of Geodesic Point-Set Embeddability (Geodesic PSE) [KKRW10]: Given a graph G and a finite set of points P, we ask whether G can be embedded on the grid, i.e., whether the vertices of G can be mapped to the points in P, and all edges of G are represented by geodesic chains that run on the grid. Katz *et al.* [KKRW10] show that the problem is NP-hard on the orthogonal grid by reduction from Hamiltonian Cycle Completion (HCC), which in turn is shown to be hard by reduction from Hamiltonian Cycle in the same work. We will show that the problem is hard on the hexagonal grid, too.

Let G be a planar, non-Hamiltonian, cubic graph. We show that the problem is already hard for such graphs.

Theorem 4.2.1. Geodesic PSE *is* NP*-hard on the hexagonal grid.*

Prior to proving this, we need to introduce some more notation. Let k_0, k_1, k_2 be non-negative integers. By $n = |V|$ we denote the number of vertices of G. Further, let the point sets $P_0 = \{(-j, 0) \mid j = 0, \ldots, k_0 - 1\}$, $P_1 = \{(j, nj) \mid j = 1, \ldots, k_1\}$, $P_2 = \{(j, -nj \mid j =$

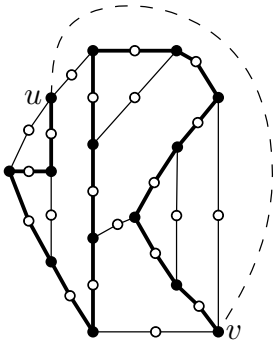


Figure 4.7: The sample graph G' as used by Katz *et al.* [KKRW10]. The bold black edges indicate a Hamiltonian path connecting u and v and its subdivision vertices. The dashed line indicates the edge that is required to complete the Hamiltonian path to a Hamiltonian cycle.

$1, \ldots, k_2\}$ and $P(k_0, k_1, k_2) = P_0 \cup P_1 \cup P_2$ be given (Figure 4.6a). Katz *et al.* [KKRW10] use the point set $P(k_0, k_1, k_2)$ as an important part of the proof that Geodesic PSE is NP-hard. Let us consider particularly point v of Figure 4.6c. On the orthogonal grid, there can be at most two geodesics from v to points in $P(k_0, k_1, k_2)$. On the hexagonal grid the situation changes, as Figure 4.6d shows. Here, there are up to three geodesics that may go from v to points in $P(k_0, k_1, k_2)$. Since the proof for the orthogonal grid crucially relies on the property that there are at most *two* such geodesics, we cannot simply reuse $P(k_0, k_1, k_2)$ as is for our proof of hardness. Instead, we modify $P(k_0, k_1, k_2)$ by setting $P_2 = \{(-i, -nj) \mid i = k_0, k_0 + 1, \ldots, k_0 + k_2 \wedge j = 1, \ldots, k_2\}$, as illustrated in Figure 4.6b. That way, we can restore the property that there are at most two geodesics from v to points in $P(k_0, k_1, k_2)$, shown in Figure 4.6e, since v can no longer connect to points in P_2 using a geodesic path. On a side note, the construction as shown in Figure 4.6b also works for the hardness proof on the orthogonal grid.

Proof. The proof is by reduction from HCC and resembles the one by Katz *et al.* [KKRW10]. Suppose we are given an instance $G = (V, E)$ of HCC. Let $k = n/2 + 1$. We construct a new graph $G' = (V', E')$ by subdividing every edge of G by a vertex of degree 2. Thus, G' has $|V'| = |V| + |E| = n + 3n/2 = 2n + n/2 = 2n - 1 + k$ vertices. We now show that G' can be embedded on $P(2n - 1, k_1, k_2)$, for some k_1, k_2 such that $k_1 + k_2 = k$, if and only if G is a *yes*-instance of HCC.

Suppose G is a *yes*-instance of HCC. Then we can find two vertices u and v such that $G + uv$ is a planar, Hamiltonian graph and $G + uv$ has an embedding such that u and v are incident to at most two faces on the same side of the Hamiltonian cycle. For better understanding, we consider the same sample graph as Katz *et al.* [KKRW10] do in their work (Figure 4.7) and embed it on the sheared grid. Original vertices of G are drawn as black disks, while subdivision vertices are represented by circles. We can embed the Hamiltonian path connecting u and v and the according subdivision vertices on a horizontal path consisting of $n + (n - 1) = 2n - 1$ points. Furthermore, we embed the faces inside the Hamiltonian cycle above and the faces outside the cycle below that path, maintaining the combinatorial embedding—see Figure 4.8. Since G is cubic, the vertices of G' have degree at most 3. Original vertices in our embedding have at most one edge going up (upward right) or one going down (downward left), except u and v, which both have one edge going up and one edge going down. Let k_1 and k_2 denote the number of edges inside and outside the cycle, respectively. Still maintaining the combinatorial embedding, we map the

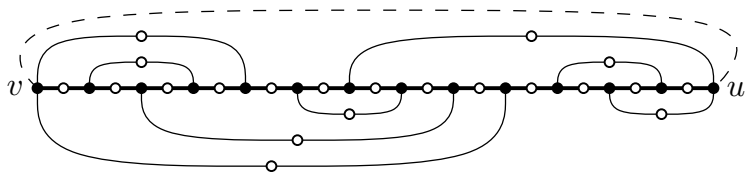


Figure 4.8: The sample graph G' drawn in a way such that the Hamiltonian path is embedded on a horizontal line consisting of $2n-1$ points.

remaining subdivision vertices that are not part of the Hamiltonian path to $k = k_1 + k_2$ points in $P_1 \cup P_2$, totalling $2n-1+k$ points, as desired, and draw the according edges as shown in Figure 4.9. Each vertex $v \in P_1 \cup P_2$ has two neighbors, a "left" neighbor v_l and a "right" neighbor v_r, according to their x-coordinates. If $v \in P_1$, we route the edge vv_l with one bend and the edge vv_r with two bends. If, on the other hand, $v \in P_2$, we route the edge vv_l with two bends and the edge vv_r with one bend.

Conversely, suppose G has a geodesic embedding on $P(2n-1, k_1, k_2)$, $k_1 + k_2 = k$. The k vertices that are mapped to points in $P_1 \cup P_2$ are incident to at most $2k = n+2$ edges.[2] Since G is a cubic graph, it has $3n/2$ edges. Thus, by subdividing every edge of G by a vertex of degree 2, G' has $(3n/2) \cdot 2 = 3n$ edges. Of these remain $3n - (n+2) = 2n-2$ for connecting points in P_0. Since there are $2n-1$ points in P_0, and by construction of G', P_0 induces a path π that alternates between original vertices of degree 3 and subdivision vertices having degree 2. Let s and t be the two endpoints of P_0. Now, either both s and t have degree 3 or both of them have degree 2. Suppose they have degree 2. Since $P_0 = 2n-1$, π contains $(2n-1) - n = n-1$ degree-3 vertices. The remaining vertex of degree 3 thus needs to be mapped to some point in either P_1 or P_2. That way it is possible to extend π to a Hamiltonian cycle, which contradicts our assumption of G being non-Hamiltonian.

Therefore, s and t must be original vertices of G having degree 3 and they belong to a Hamiltonian path that connects s and t in G. This path can be extended to a Hamiltonian cycle by an edge through the outer face of G. Since both s and t have one edge going up and one going down, they are incident to two faces on either side of the cycle in the embedding. Thus, G is a *yes*-instance of HCC, which concludes the proof. □

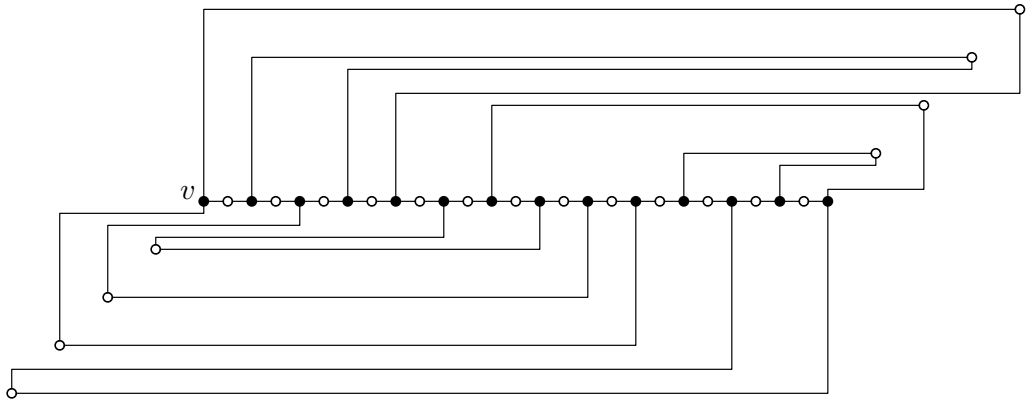


Figure 4.9: The geodesic embedding of G' on the sheared grid.

[2]We discussed earlier that at most two geodesics can go from any such vertex to points in $P(k_0, k_1, k_2)$.

4.3 Sparse Labeled Geodesic Point-Set Embeddability

In this section, we investigate the problem of Labeled Geodesic Point-Set Embeddability (Labeled Geodesic PSE), first introduced by Katz *et al.* [KKRW10]. Here, in addition to an input graph $G = (V, E)$, the correspondence of vertices and points on the grid is given as part of the problem instance. More formally, we have a bijection $\mu : V \to \mathbb{R}^2$ that maps the vertices of the graph to points on the grid. By reducing 3-Partition to Labeled Geodesic Matching (LGM), Katz *et al.* [KKRW10] show that Labeled Geodesic PSE, a special case of LGM, is NP-hard on the orthogonal grid and that it can be solved efficiently when the space limitation of the grid is loosened or dropped.

In the following, we will show that the problem is NP-hard on the hexagonal grid as well (using the sheared grid due to greater resemblance to the orthogonal grid) and that we can apply the same algorithm as proposed by Katz *et al.* [KKRW10] to solve certain problem instances, which we will call *sparse*, efficiently. For this, we will introduce the same notions, yet adapted to the hexagonal and sheared grids.

In order to prove the NP-hardness of Labeled Geodesic PSE, we will make use of the following two lemmas.

Lemma 4.3.1 (Shearing Lemma). Let p_1 and p_2 be two paths having the same lengths, $|p_1| = |p_2|$, on the orthogonal grid. Then p_1 and p_2 still have the same length after having sheared one or both axes of the grid.

Proof. Considering the line segments of the grid, shearing operations merely result in a scaling of the line segments, which again results in shorter or longer paths, which means that the lengths before and after the shearing do not match. However, since the number of line segments occupied by p_1 and p_2 does not change, the property that p_1 and p_2 have the same length is preserved when scaling line segments, i.e., still $|p_1| = |p_2|$. □

Lemma 4.3.2 Let g be an upward geodesic on the sheared grid. Then g does not consist of any diagonal line segments.

Proof. Regarding particularly upward geodesics, it can be seen in Figure 4.1b that it is categorically impossible for them to use diagonal line segments, since diagonals run from the upper left to the lower right, but not vice versa, as would be needed for g. Using such diagonal line segments anyway always introduces a violation of the triangle inequality. However, the triangle inequality is a critical property of a metric, and thus a property of geodesics as well. □

Corollary 4.3.1. Labeled Geodesic PSE *is* NP*-hard on the hexagonal grid.*

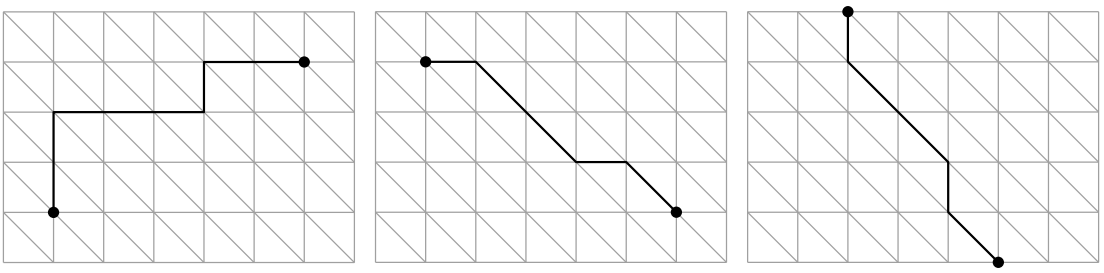
Proof. We essentially use the proof of NP-hardness of Labeled Geodesic PSE on the orthogonal grid given by Katz *et al.* [KKRW10]. Instead of considering downward geodesics, however, we consider upward geodesics on the sheared grid. Hence, we simply reflect the problem instance across the x-axis. This is possible, since upward geodesics do not use diagonals, in accordance with Lemma 4.3.2. The upward geodesics then run on an orthogonal grid. Finally, applying the Shearing Lemma 4.3.1 yields the NP-hardness on the hexagonal grid. □

In the following, we need to make a further distinction regarding geodesics on the sheared grid. So far, we distinguish only upward and downward geodesics. As noted further above, the sheared grid can be seen as an extension of the orthogonal grid by adding diagonals that go from the upper left to the lower right. Upward geodesics, as follows from Lemma 4.3.2, do not use those diagonals—Figure 4.10a shows an example. Downward geodesics, however, may use them. Let g_d be a downward geodesic. Depending on the type of downward geodesic, g_d, in addition to diagonal line segments, uses either horizontal or vertical line segments, which is essentially due to the triangle inequality. Henceforth, we distinguish two types of downward geodesics:

1. *Shallow downward geodesics* move only along horizontal and diagonal line segments of the grid. They do not use vertical line segments. Consider Figure 4.10b for an example.
2. *Steep downward geodesics* move only along vertical and diagonal line segments of the grid. They do not use horizontal line segments. Consider Figure 4.10c for an example.

From now on, unless stated otherwise, our observations refer only to the sheared grid. An instance (G, μ) of Labeled Geodesic PSE is called *sparse* if the minimum distance between any two occupied rows and any two occupied columns on the sheared grid—recall Figure 4.1b—is at least $3n - 6$. This requirement is due to the fact that, by Euler's formula [DBETT99], any planar graph consisting of n vertices has at most $3n - 6$ edges, and, in order to avoid space shortages, we consider sparse instances only. We call a vertex v of a 6-planar graph *admissible* if it is adjacent to at most one vertex on each ray starting at v, to at most two vertices in each (closed) sextant with respect to v and at most four vertices in each (closed) axis-aligned half-plane with respect to v. A 6-planar graph G can only have a geodesic embedding on the hexagonal grid if all of its vertices are admissible. In this case, we say that G is *admissible*.

Let γ be a geodesic embedding of $G = (V, E)$. For $e \in E$, we denote by $\gamma(e)$ the according embedding of e. We say that $\gamma(e)$ is *below* (*above*) $\gamma(f)$ if there is a vertical line intersecting $\gamma(e)$ below (above) $\gamma(f)$. We say that e is *strictly below* f if $\gamma(e)$ is below $\gamma(f)$ in every possible geodesic embedding of G. Although not formally exact, for convenience, we will mostly omit to explicitly mention the embedding $\gamma(e)$ and refer to the according geodesic embedding of e simply as e. In order to derive a combinatorial description of a geodesic embedding in terms of these "above-below" relations, we are going to introduce a relation $\prec$ on the set of edges. It should be noted, however, that not every such relation corresponds to a geodesic embedding of a graph.



(a) Upward geodesic. (b) Shallow downward geodesic. (c) Steep downward geodesic.

Figure 4.10: Geodesics on the sheared grid can be divided into three types: upward geodesics, shallow downward geodesics and steep downward geodesics.

Given some point $p \in \mathbb{R}^2$, we denote by $Q_k(p)$ the set of points (excluding p) in the (closed) k-th sextant of the coordinate system centered at p on the sheared grid. Since we will further below, in addition to the sheared grid, refer to the orthogonal grid as well, we also denote by $Q_k(p)$ the set of points in the (closed) k-th quadrant of the coordinate system centered at p on the orthogonal grid, again excluding p from $Q_k(p)$. It will be clear from the context when $Q_k(p)$ refers to orthogonal grid. Unless stated otherwise, we will refer to the sheared grid.

Again on the sheared grid, we say that $p \in \mathbb{R}^2$ *k-dominates* $q \in \mathbb{R}^2$ if $q \in Q_k(p)$. Given $e \in E$, we set $Q_k(e) := Q_k(e^-) \cup Q_k(e^+)$, where e^- and e^+ denote the lexicographically smaller and larger endpoint of e, respectively, according to the x-axis. We say that e k-dominates $q \in \mathbb{R}^2$ if $q \in Q_k(e)$. The *k-critical set of edges of* e is defined as $C_k(e) := \{f \in E \mid f \cap Q_k(e) \neq \varnothing\}$, that is, $C_k(e)$ contains all edges that have at least one of their endpoints in $Q_k(e)$. We define $Q_k(e)$ and $C_k(e)$ on the orthogonal grid analogously. Again, the context will clearly indicate when we refer to the orthogonal grid.

On the orthogonal grid, Katz *et al.* [KKRW10] present the following dualities that hold for all edges e and f:

$$f \in C_2(e) \Leftrightarrow e \in C_4(f) \text{ and } f \in C_1(e) \Leftrightarrow e \in C_3(f).$$

Similar to the orthogonal grid, there are dualities that hold for all edges e and f on the sheared grid:

$$f \in C_1(e) \Leftrightarrow e \in C_4(f) \text{ and } f \in C_2(e) \Leftrightarrow e \in C_5(f) \text{ and } f \in C_3(e) \Leftrightarrow e \in C_6(f).$$

Given a point $p = (p(x), p(y)) \in \mathbb{R}^2$, let $p^{\updownarrow} := \{q \in \mathbb{R} \mid q(x) = p(x)\}$, that is, the set of all points q, such that q and p are vertically aligned.

Furthermore, for $e \in E$, by $\mathcal{B}(e)$ we denote the *bounding box of* e, which is given by the rectangle determined as follows: If e is an upward edge, then e^- is the lower left corner point of the rectangle $\mathcal{B}(e)$, and e^+ is the upper right corner point of $\mathcal{B}(e)$. If e is downward, similarly, e^- is the upper left and e^+ the lower right corner point of $\mathcal{B}(e)$. Note that, if e^- and e^+ are horizontally or vertically aligned, then $\mathcal{B}(e)$ is given by a single horizontal or vertical line, respectively, connecting e^- and e^+.

For $e, f \in E$, we say that the e and f *overlap* if there is a common vertical line that intersects e and f. Note that, at this point, we do not know whether e is situated above or below f—we merely require a common vertical line to intersect the two edges.

A *combinatorial geodesic embedding of* G is given by a relation $\prec$ on the set of edges such that two edges e and f are comparable, denoted by $e \prec f$, *if and only if* there is a vertical line intersecting e below f in the embedding of G. Note that this implies that e and f overlap. For convenience, we shall write $f \succ e$ equivalently for $e \prec f$. In addition to this, we require two further properties. First, for any three edges e, f, g whose bounding boxes are intersected by a common vertical line, we have that $e \prec f$ and $f \prec g$ imply $e \prec g$. We call this property *local transitivity*. Secondly, we require certain *implication rules* to hold. Note that local transitivity is a necessary condition for the existence of a geodesic embedding. Unfortunately the implication rules given for the orthogonal grid by Katz *et al.* [KKRW10] do not hold as presented in their work. Their rules are too general, and thus wrong in certain cases, as we will see next.

The following implication rules for the orthogonal grid are the original rules presented by Katz *et al.* [KKRW10] in their work. Note that their rules were not given the names that we introduce below for easier later reference.

Implication Rule [KKRW10]**.** Let $e \in E$ be an upward edge and let $f \in E$.

(U1) If $f \in C_2(e)$, then $e \prec f$.
(U2) If $f \in C_4(e)$, then $f \prec e$.
(U3) If $f \prec e$, then $f' \prec e$ for all $f' \in C_4(f)$.
(U4) If $e \prec f$, then $e \prec f'$ for all $f' \in C_2(f)$.

Let $e \in E$ be a downward edge and let $f \in E$.

(D1) If $f \in C_1(e)$, then $e \prec f$.
(D2) If $f \in C_3(e)$, then $f \prec e$.
(D3) If $f \prec e$, then $f' \prec e$ for all $f' \in C_3(f)$.
(D4) If $e \prec f$, then $e \prec f'$ for all $f' \in C_1(f)$.

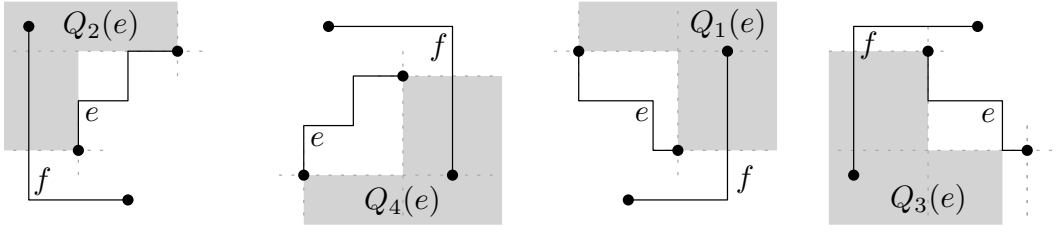
Without further restrictions, these implication rules are not correct, as we will prove by giving counterexamples. First, let us consider the rules (U1), (U2), (D1) and (D2) and the according Figures 4.11a-4.11d that depict our counterexamples. Suppose $e \in E$ is upward and let $f \in E$. Starting with rule (U1), Figure 4.11a shows an edge f and an upward edge e, such that $f \in C_2(e)$. According to rule (U1), we shall conclude that $e \prec f$. Clearly, this is not the case in our counterexample—here, we have $f \prec e$, and thus a contradiction to implication rule (U1).

Next, we consider rule (U2) and Figure 4.11b. In the figure, we have $f \in C_4(e)$ and according to rule (U2), it follows that $f \prec e$. In our case, however, we obviously have $e \prec f$, in contradiciton to rule (U2).

Let $e \in E$ be a downward edge and let $f \in E$. Figure 4.11c shows an example that disproves implication rule (D1). We have $f \in C_1(e)$ and in accordance with rule (D1) we conclude $e \prec f$. Obviously, the Figure shows $f \prec e$, which yields a contradiction rule (D1).

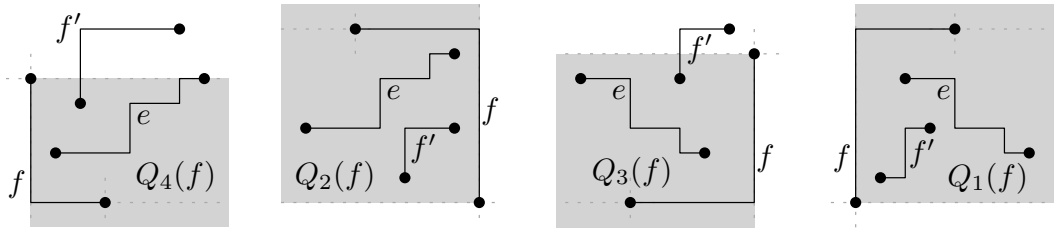
Considering implication rule (D2), Figure 4.11d shows that $f \in C_3(e)$ as required by rule (D2). We cannot conclude $f \prec e$, however, since the figure clearly shows that $e \prec f$, and thus a contradiction.

We now focus on the remaining rules (U3), (U4), (D3) and (D4). Consider Figures 4.12a-4.12d. According to Figure 4.12a, implication rule (U3) does not hold. Obviously, we have



(a) Counterexample to rule (U1).
(b) Counterexample to rule (U2).
(c) Counterexample to rule (D1).
(d) Counterexample to rule (D2).

Figure 4.11: Counterexamples to the implication rules (U1), (U2) and (D1), (D2) as presented by Katz *et al.* [KKRW10].



(a) Counterexample to rule (U3). (b) Counterexample to rule (U4). (c) Counterexample to rule (D3). (d) Counterexample to rule (D4).

Figure 4.12: Counterexamples to the implication rules (U3), (U4) and (D3), (D4) as presented by Katz *et al.* [KKRW10].

$f \prec e$ and $f' \in C_4(f)$. However, $f' \prec e$ does *not* hold, since $f' \succ e$. In other words, we have $e \prec f'$, which disproves implication rule (U3).

Similarly, consider Figure 4.12b. Clearly, it holds that $e \prec f$ and $f' \in C_2(f)$. However, we also have $e \succ f'$, which is equivalent to $f' \prec e$, and thus a contradiction to implication rule (U4).

Moving on to rule (D3), consider Figure 4.12c. Again, it can be seen that the premise $f \prec e$ and $f' \in C_3(f)$ is true. We cannot conclude $f' \prec e$ according to (D3), however, since $f' \succ e$, which is equivalent to $e \prec f'$.

Finally, let us take a look at rule (D4). Figure 4.12d shows that $e \prec f$ and $f' \in C_1(f)$. According to (D4), it then follows that $e \prec f$. However, since the figure shows that $f' \prec e$, we have a contradiction to rule (D4).

These implication rules can be corrected by making further assumptions. In the following, we present the set of fixed rules.

Implication Rules (Orthogonal Grid). Let $e \in E$ be an upward edge. Let $f, f' \in E$.

(U1) Let e and f overlap. If $f \in C_2(e)$ is upward, then $e \prec f$.
(U2) Let e and f overlap. If $f \in C_4(e)$ is upward, then $f \prec e$.
(U3) Let e and f' overlap. If $f \in C_2(e)$ and $f' \succ f$ are upward, then $f' \succ e$.
(U4) Let e and f' overlap. If $f \in C_2(e)$ is downward and $f' \succ f$ with $\mathcal{B}(f') \cap f^{-\updownarrow} \neq \varnothing$ is upward, then $f' \succ e$.
(U5) Let e and f' overlap. If $f \in C_4(e)$ and $f' \prec f$ are upward, then $f' \prec e$.
(U6) Let e and f' overlap. If $f \in C_4(e)$ is downward and $f' \prec f$ with $\mathcal{B}(f') \cap f^{+\updownarrow} \neq \varnothing$ is upward, then $f' \prec e$.

Let $e \in E$ be a downward edge. Let $f, f' \in E$.

(D1) Let e and f overlap. If $f \in C_1(e)$ is downward, then $e \prec f$.
(D2) Let e and f overlap. If $f \in C_3(e)$ is downward, then $f \prec e$.
(D3) Let e and f' overlap. If $f \in C_1(e)$ and $f' \succ f$ are downward, then $f' \succ e$.
(D4) Let e and f' overlap. If $f \in C_1(e)$ is upward and $f' \succ f$ with $\mathcal{B}(f') \cap f^{+\updownarrow} \neq \varnothing$ is downward, then $f' \succ e$.
(D5) Let e and f' overlap. If $f \in C_3(e)$ and $f' \prec f$ are downward, then $f' \prec e$.
(D6) Let e and f' overlap. If $f \in C_3(e)$ is upward and $f' \prec f$ with $\mathcal{B}(f') \cap f^{-\updownarrow} \neq \varnothing$ is downward, then $f' \prec e$.

Lemma 4.3.3 The implication rules are correct.

Proof. We will prove the rules not only with respect to the orthogonal grid, that is, we do not require edges to consist of horizontal and vertical line segments only. Instead, we only require edges $e \in E$ to be realized as monotone curves between e^- and e^+. An upward edge is now an edge whose slope is not negative. Likewise, the slope of a downward edge is not positive. Therefore, the implication rules are not restricted to the orthogonal grid, but we can also apply them with respect to the sheared grid later on.

For upward edges we will prove the according implication rules (U1)-(U6) one by one. Then we will argue that the correctness of the implication rules for downward edges follows by symmetry. Let $e \in E$ be an upward edge and let $f, f' \in E$.

(U1) Let $f \in C_2(e)$ be upward, and let e and f overlap. By definition of $C_2(e)$ we know that one or both endpoints of f are in $Q_2(e^-) \cup Q_2(e^+)$. Assume that f^+ is one of these two points, that is, $f^+ \in Q_2(e^-) \cup Q_2(e^+)$. Since f is upward, f^+ cannot be in $Q_2(e^-)$, and thus f^+ must be in $Q_2(e^+) \backslash Q_2(e^-)$. This in turn implies the existence of a point p_e of e situated vertically below f^+ (since f is upward). Hence, we can find a common vertical line that intersects e in p_e below f in f^+ and we conclude that $e \prec f$. In case f^+ coincides with e^+, we find a point p_f immediately left of f^+ and use p_f to find another point p_e located vertically below f on e. This again yields $e \prec f$. Note that p_f and p_e cannot coincide in this case, since we do not allow edges to cross. Conversely, if f^+ is not in $Q_2(e^-) \cup Q_2(e^+)$, then f^- must be in $Q_2(e^-) \cup Q_2(e^+)$. If $f^- \in Q_2(e^+)$, then we can argue as above and conclude $e \prec f$. Thus, suppose that $f^- \in Q_2(e^-)$. Since we assume e and f to overlap and since f is upward, we can find a point p_f on f that is situated vertically above e^-. This implies the existence of a vertical line that hits e^- below p_f, which means that e and f become comparable with respect to $\prec$, and we clearly have $e \prec f$.

(U2) Let $f \in C_4(e)$ be upward, and let e and f overlap. We use the duality from above stating that $f \in C_2(e) \Leftrightarrow e \in C_4(f)$. Since $f \in C_4(e)$, we know that $e \in C_2(f)$. Moreover, since e and f overlap, and since e is upward, we can apply rule (U1) to conclude that $f \prec e$.

(U3) Let e and f' overlap, let $f \in C_2(e)$ and $f' \succ f$ be upward edges. We distinguish two cases.

(i) Suppose that e and f overlap. Since e and f overlap and due to the fact that $f \in C_2(e)$, we can apply rule (U1), which implies $e \prec f$, and thus $f \succ e$. We now have $f' \succ f \succ e$. Moreover, since e, f and f' overlap pairwise, we can find three points on e, f and f'—that is, one situated on e, one situated on f and one situated on f'—that are intersected by a common vertical line. At this point, we can apply local transitivity, that is, transitivity with respect to the vertical line intersecting e, f and f', and we conclude that $f' \succ e$.

(ii) Conversely, suppose that e and f do not overlap. We use this fact to know that f^+ must be in $Q_2(e^-)$, and therefore $y(f^+) \geqslant y(e^-)$. Let γ_1 be the y-coordinate of $\gamma(f')$ at f^+, that is, vertically above f^+. Furthermore, let γ_2 be the y-coordinate of $\gamma(f')$ at e^-. We know that $\gamma_1 \geqslant y(f^+)$, since $f' \succ f$. We also know that $\gamma_2 \geqslant \gamma_1$, since f' is assumed to be upward. This suffices to conclude that $\gamma_2 \geqslant y(e^-)$. Hence, we find a vertical line that hits $e^- = (x(e^-), y(e^-))$ below some point $(x(e^-), \gamma_2)$ situated on f'. Therefore, e and f' are comparable with respect to $\prec$, and we conclude $e \prec f'$, that is, $f' \succ e$.

(U4) Let e and f' overlap, let $f \in C_2(e)$ be a downward edge, and let $f' \succ f$ be an upward edge with $\mathcal{B}(f') \cap f^{-\updownarrow} \neq \varnothing$. We know that f^- is in $Q_2(e)$. We distinguish two cases.

(i) Suppose that $f^- \in Q_2(e^+) \backslash Q_2(e^-)$. Consider point f^-. Clearly, f^- is above e^+ since $f^- \in Q_2(e^+)$. Moreover, again due to $f^- \in Q_2(e^+)$ and since e is upward, f^- is above all points on e. Since $\mathcal{B}(f') \cap f^{-\updownarrow} \neq \varnothing$, we know that there exists some point $p_{f'}$ on f' such that $y(p_{f'}) \geqslant y(f^-)$. Due to the fact that, for all $p_e \in \gamma(e)$, we have that $y(f^-) \geqslant y(p_e)$, we also know that $y(p_{f'}) \geqslant y(p_e)$. Since e and f' overlap, we can thus find a point $q_{f'}$ on f' (satisfying $y(q_{f'}) \geqslant y(p_{f'})$, because f' is upward), such that $q_{f'}$ is above all points on e, that is, $y(q_{f'}) \geqslant y(p_e)$, for all $p_e \in \gamma(e)$, and such that there is a point q_e on e situated vertically below $q_{f'}$. We can conclude that there exists a vertical line that hits q_e below $q_{f'}$, and thus this line intersects e below f'. By the definition of the $\prec$ relation, this in turn is equivalent to $e \prec f'$, and therefore we finally have $f' \succ e$.

(ii) Suppose that $f^- \in Q_2(e^-)$. We know that $y(f^-) \geqslant y(e^-)$, since $f \in C_2(e)$. Furthermore, due to the fact that $\mathcal{B}(f') \cap f^{-\updownarrow} \neq \varnothing$, there exists a point $p_{f'}$ on f' such that $y(p_{f'}) \geqslant y(f^-)$. Thus, we also know that $y(p_{f'}) \geqslant y(e^-)$. Moreover, since f' is upward, there exists another point $q_{f'}$ on f' satisfying $y(q_{f'}) \geqslant y(p_{f'})$. Since $y(p_{f'}) \geqslant y(e^-)$, we also have $y(q_{f'}) \geqslant y(e^-)$. Note that we assume e and f' to overlap, and therefore we can even constrain $q_{f'}$ to be situated vertically above e^-. This makes e and f' comparable with respect to $\prec$, since there is a vertical line that intersects e in e^- below f' in $q_{f'}$. Hence, we have $e \prec f'$, and thus $f' \succ e$.

(U5) Suppose e and f' overlap, let $f \in C_4(e)$ and $f' \prec f$ be upward edges. We reflect e, f and f' across the horizontal x-axis and the vertical y-axis according to the coordinate system originating in e^- (or e^+, either will do). (Equivalently, we rotate the edges by π radians using the same coordinate system.) We obtain from e, f and f' the new edges e_r, f_r and f'_r, respectively. All edges remain upward, and we now have $f_r \in C_2(e_r)$ as well as $f'_r \succ f_r$. The latter is due to the reflection across the y-axis, which intuitively exchanges top and bottom. Note that we now have all prerequisites needed for rule (U3), and we can conclude that $f'_r \succ e_r$ holds for the reflected edges. Therefore, before the reflection, we must have had $f' \prec e$. Put another way, knowing that $f'_r \succ e_r$ by rule (U3), we reflect again as described above and obtain $f' \prec e$.

(U6) Suppose e and f' overlap, let $f \in C_4(e)$ be a downward edge, and let $f' \prec f$ be an upward edge with $\mathcal{B}(f') \cap f^{+\updownarrow} \neq \varnothing$. We use the same idea as we did above for rule (U5). The reflection yields upward edges e_r, f'_r and a downward edge f_r, such that $f_r \in C_2(e_r)$ and $\mathcal{B}(f'_r) \cap f_r^{-\updownarrow} \neq \varnothing$. Hence, we can apply rule (U4) and conclude that $f'_r \succ e_r$ holds for the reflected edges, and therefore we know that $f' \prec e$ held initially.

Regarding the implication rules for downward edges, that is, rules (D1)-(D6), we argue as follows. We will see that rule (D1) holds since it is the dual counterpart of rule (U2): Let $f \in C_1(e)$ be a downward edge, and let e and f overlap. We reflect e and f across the (horizontal) x-axis of the coordinate system originating in e^- (or e^+, either will do). This yields two upward edges e_r and f_r. Moreover, we now have $f_r \in C_4(e_r)$. Both e_r and f_r are upward and they need to overlap as well, and thus rule (U2) is applicable, implying that $f_r \prec e_r$. We can conclude that f_r is below e_r. Moreover, since e_r and f_r are reflected versions of e and f with respect to the x-axis of the coordinate system centered at e^- (or

e^+), we know that f must be above e. Finally, using the fact that f is above e, we can conclude $f \succ e$, and thus $e \prec f$.

Essentially, we took advantage of symmetry to prove rule (D1). The same arguments can be used to prove the remaining rules (D2)-(D6). □

Similarly, there are implication rules that hold on the sheared grid. Note that the second and third sextants of the sheared grid correspond to the second quadrant of the orthogonal grid, and that the fifth and sixth sextants of the sheared grid correspond to the fourth quadrant on the orthogonal grid. Therefore, and also due to the fact that correctness of the implication rules was proven independently of the orthogonal grid, we can transfer the implication rules for the orthogonal grid to the sheared grid. Moreover, since we can distinguish two types of downward edges, that is, shallow downward and steep downward edges, we gain additional information that implies further implication rules. In the following, let $C_{2\cup3} := C_2 \cup C_3$ and $C_{5\cup6} := C_5 \cup C_6$. The implication rules for the sheared grid are as follows.

Implication Rules (Sheared Grid). Let $e \in E$ be an upward edge. Let $f, f' \in E$.

(U1) Let e and f overlap. If $f \in C_{2\cup3}(e)$ is upward, then $e \prec f$.

(U2) Let e and f overlap. If $f \in C_{5\cup6}(e)$ is upward, then $f \prec e$.

(U3) Let e and f' overlap. If $f \in C_{2\cup3}(e)$ and $f' \succ f$ are upward, then $f' \succ e$.

(U4) Let e and f' overlap. If $f \in C_{2\cup3}(e)$ is downward and $f' \succ f$ with $\mathcal{B}(f') \cap f^{-\updownarrow} \neq \emptyset$ is upward, then $f' \succ e$.

(U5) Let e and f' overlap. If $f \in C_{5\cup6}(e)$ and $f' \prec f$ are upward, then $f' \prec e$.

(U6) Let e and f' overlap. If $f \in C_{5\cup6}(e)$ is downward and $f' \prec f$ with $\mathcal{B}(f') \cap f^{+\updownarrow} \neq \emptyset$ is upward, then $f' \prec e$.

Let $e \in E$ be a downward edge. Let $f, f' \in E$.

(D1) Let e and f' overlap. If $f \in C_1(e)$ and $f' \succ f$ are downward, then $f' \succ e$.

(D2) Let e and f' overlap. If $f \in C_1(e)$ is upward and $f' \succ f$ with $\mathcal{B}(f') \cap f^{+\updownarrow} \neq \emptyset$ is downward, then $f' \succ e$.

(D3) Let e and f' overlap. If $f \in C_4(e)$ and $f' \prec f$ are downward, then $f' \prec e$.

(D4) Let e and f' overlap. If $f \in C_4(e)$ is upward and $f' \prec f$ with $\mathcal{B}(f') \cap f^{-\updownarrow} \neq \emptyset$ is downward, then $f' \prec e$.

Let $e \in E$ be a shallow downward edge. Let $f, f' \in E$.

(D5) Let e and f overlap. If $f \in C_1(e)$ is downward, then $e \prec f$.

(D6) Let e and f overlap. If $f \in C_4(e)$ is downward, then $f \prec e$.

(D7) Let e and f overlap. If $f \in C_2(e)$ is shallow downward, then $e \prec f$.

(D8) Let e and f overlap. If $f \in C_5(e)$ is shallow downward, then $f \prec e$.

Let $e \in E$ be a steep downward edge. Let $f, f' \in E$.

(D9) Let e and f overlap. If $f \in C_1(e)$ is downward, then $e \prec f$.

(D10) Let e and f overlap. If $f \in C_5(e)$ is downward, then $f \prec e$.

(D11) Let e and f overlap. If $f \in C_3(e)$ is steep downward, then $f \prec e$.

(D12) Let e and f overlap. If $f \in C_6(e)$ is steep downward, then $e \prec f$.

These implication rules apply to the sheared grid as well as the hexagonal grid. However, on the hexagonal grid, the relation $\prec$ is no longer defined by a common *vertical* line intersecting the bounding boxes of two edges, but by a common *diagonal* line going from the lower left to the upper right. As soon as any such implication rule is violated given a concrete relation $\prec$ for an edge e of G, this means that the corresponding order cannot be translated into a geodesic chain, which in turn implies that the relation does not correspond to a partial order that we can derive from a given geodesic embedding of G.

Next, we show that we can efficiently compute a geodesic embedding of an admissible graph G according to a given combinatorial embedding $\prec$ of G.

Given a combinatorial embedding $\prec$ of a graph $G = (V, E)$, let $G_\prec = (E, A)$ denote the corresponding directed graph on the set of edges, where $(e, f) \in A$ whenever $e \prec f$ in the embedding of G. We call $G_\prec$ the *representation (graph) of* $\prec$.

Theorem 4.3.1. *Let $G = (V, E)$ be admissible, and let $\prec$ be a combinatorial embedding of G. Given a representation $G_\prec$ of $\prec$, we can compute a geodesic embedding of G according to $\prec$ in $\mathcal{O}(n^2)$ time, which is worst-case optimal.*

Proof. The proof is similar to the one on the orthogonal grid by Katz *et al.* [KKRW10]. For the proof, we use the sheared grid due to greater similarity to the orthogonal grid—recall Figure 4.1b. We use a vertical sweep line moving from left to right in order to compute the embedding of G. Events, that is, decisions being made regarding further edge embeddings, occur at the vertices of G, sorted in lexicographical order from left to right and from bottom to top. While the line is swept, we divide the edges E of G into three groups. *Completed edges* are edges that have both their endpoints situated left of the sweep line, which means their geodesic embeddings were already entirely computed by the algorithm. *Partial edges* have one endpoint to the left and one endpoint to the right of the sweep line. A partial edge is embedded as a partial geodesic up to the current location of the sweep line. The remaining part of the edge, right of the sweep line, has yet to be computed. *Untouched edges* are those that have both their endpoints located right of the sweep line, which means that computation of their embeddings has not yet started.

Let c and c' be two consecutive occupied grid columns, c left of c'. We assume that we have already computed a partial geodesic embedding up to column c, that is, all partially or entirely embedded edges left of c have already been computed, and thus are fixed in the remaining computation. The main idea is as follows: We extend all edges not ending at c iteratively from bottom to top such that each newly embedded geodesic chain is situated *directly* above all previously embedded chains.

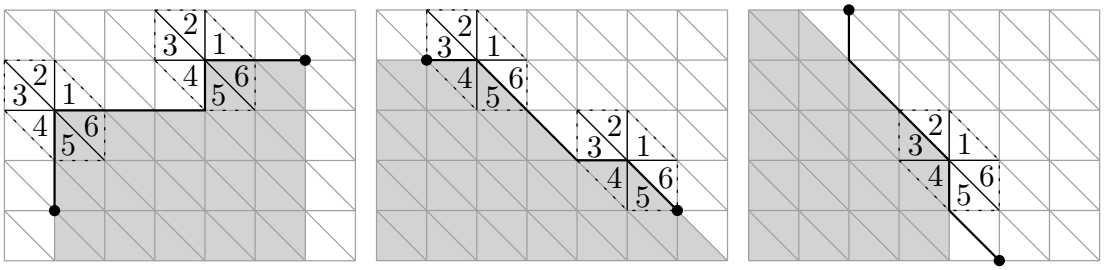
For this, we first sort the events in lexicographical order, i.e., from left to right and bottom to top, in $\mathcal{O}(n \log n)$ time. Let the order of events be given by $v_1, v_2, \ldots, v_n$. Then, we topologically sort the edges according to $\prec$ in $\mathcal{O}(n^2)$ time. Let the order of edges be given by $e_1, e_2, \ldots, e_m$. To see that this can be done in quadratic time, consider the following: We can sort the edges topologically in time proportional to the number of edges in $G_\prec$, which is in $\mathcal{O}(n^2)$, since $G_\prec$ is a simple graph whose vertices correspond to edges in G, and G is planar and thus has at most $3n - 6$ edges, in accordance with Euler's formula. Since each edge may be compared to all other edges, this results in $\mathcal{O}(n^2)$ time for the topological sort. The idea is to merge the order of events with the order of edges in order to compute the embedding. We store a list of partial edges at the sweep line sorted according to $\prec$. For instance, let that list be given by $e_{\alpha(1)}, e_{\alpha(2)}, \ldots, e_{\alpha(k)}$, sorted from bottom to top, when the sweep line is located at column c. Note that this list must be a sequence of

$e_1, e_2, \ldots, e_m$, since the edges are totally ordered by $\prec$. Let v be the next event on column c' and $e_{\alpha(i)}$, for some i, be the next edge to be embedded. We must decide whether to embed $e_{\alpha(i)}$ above or below v.[3] Whenever $e_{\alpha(i)}$ is the first edge on c that is above v, we insert the edges starting at v into the list of partial edges stored at the sweep line *before* $e_{\alpha(i)}$ is embedded, i.e., the newly added edges starting at v will be considered in the following steps of the algorithm before another geodesic chain of $e_{\alpha(i)}$ is computed. For the merging to succeed, we require the following. Suppose $e := e_{\alpha(i)}$ is an upward edge[4] that must be embedded above v, and let v be the endpoint of some edge e'. Then:

1. None of the following edges $e_{\alpha(j)}$, for $j > i$, may be embedded below v. To see that this holds, assume that there exists some $j > i$ such that $f := e_{\alpha(j)}$ needs to be embedded below v. This implies $f \prec e'$. Furthermore, we have $e' \prec e$, since e must be embedded above v. Local transitivity of $\prec$ implies $f \prec e$. However, we also have $e \prec f$, since the sweep line intersects the bounding boxes of e and f at c, and thus e and f are comparable according to $\prec$, which in turn means that $\alpha(i) < \alpha(j)$ implies $e \prec f$. This yields a contradiction to the acyclicity of $\prec$.
2. e must be embedded above all edges in $C_5(e')$ and $C_6(e')$ and below all edges in $C_2(e')$ and $C_3(e')$. This is immediately fulfilled since $\prec$ is defined to respect the implication rules.

We now describe how to embed edges. First, we observe that an upward geodesic is uniquely determined by the 5-dominant and 6-dominant points corresponding to the right bends along the geodesic. This is easy to see, since, first, the fifth and sixth sextants of the sheared grid correspond to the fourth quadrant of the orthogonal grid, and, second, upward geodesics on the sheared grid do not run diagonally according to Lemma 4.3.2; they run on an orthogonal grid as a subset of the sheared grid. Figures 4.13a and 4.14 illustrate this. For downward geodesics, the situation gets more complicated and we have to distungish two cases. In the following, we will see that a shallow downward geodesic is uniquely determined by the 4-dominant and 5-dominant points, and that a steep downward geodesic is uniquely determined by the 3-dominant and 4-dominant points corresponding to the right bends along the geodesic. Hence, we distunguish:

1. Shallow downward geodesics. On the orthogonal grid, it is the 3-dominant points corresponding to the right bends along a downward geodesic that uniquely determine the geodesic. By rotating the vertical y-axis of the orthogonal grid by $\pi/4$ radians counterclockwise, we obtain a sheared version, the *shallow* (*orthogonal*) *grid*, on



(a) Upward geodesic. (b) Shallow downward geodesic. (c) Steep downward geodesic.

Figure 4.13: Geodesics on the sheared grid can be uniquely determined using the sextants of points corresponding to right bends of the geodesics.

[3] In case we need to embed $e_{\alpha(i)}$ such that it joins v, there is nothing to decide.
[4] If e is a downward edge, we can proceed analogously.

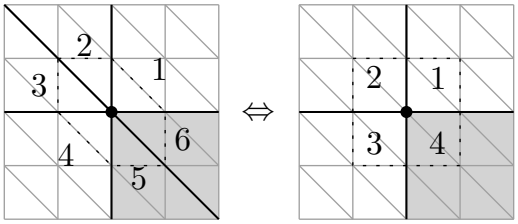


Figure 4.14: Sheared hexagonal grid and orthogonal grid.

which shallow downward geodesics run exclusively.[5] Consider Figure 4.13b as an example. This shallow grid is clearly a subset of the sheared hexagonal grid, as can be seen in Figure 4.15. We can identify the fourth and fifth sextants of the sheared hexagonal grid with the third quadrant of the shallow grid, and vice versa. Thus, on the sheared hexagonal grid, it is the 4-dominant and 5-dominant points corresponding to the right bends of a shallow downward geodesic that uniquely determine the geodesic. All shallow downward geodesics run exclusively on the shallow orthogonal grid. This is an immediate consequence of extending Lemma 4.3.2 to shallow downward geodesics.

2. Steep downward geodesics. The situation is similar to the one above. Again, recall that, on the orthogonal grid, a downward geodesic is uniquely determined by the 3-dominant points corresponding to the right bends along the geodesic. Rotating the horizontal x-axis of the orthogonal grid by $3\pi/4$ radians counterclockwise yields a sheared version, the *steep* (*orthogonal*) *grid*, on which steep downward geodesics run exclusively. Figure 4.13c illustrates this. Since the steep grid is obviously a subset of the sheared hexagonal grid—see Figure 4.16—, the third and fourth sextants of the sheared hexagonal grid correspond to the third quadrant of the steep grid, and vice versa. Hence, on the sheared hexagonal grid, the 3-dominant and 4-dominant points corresponding to the right bends of a steep downward geodesic uniquely determine the geodesic. All steep downward geodesics run exclusively on the steep grid.

We will use these facts to describe the geodesic chain corresponding to a single edge. Let R be a set of points. By $R^{\nwarrow} := \{(x-1, y+1) \mid (x, y) \in R\}$ we denote the set of points resulting from a translation of R by one unit to the left and one unit to the top. Similarly, $R^{\uparrow} := \{(x, y+1) \mid (x, y) \in R\}$ denotes the set of points resulting from translating R one unit to the top, and $R^{\rightarrow} := \{(x+1, y) \mid (x, y) \in R\}$ denotes the set of points resulting from R one unit to the right. By $H_{l,k}(R) := \bigcup_{p \in R}(Q_l(p) \cup Q_k(p))$ we denote the (l,k)-*hull of* R, that is, $H_{l,k}(R)$ is the boundary of the set of points that are l-dominated or k-dominated by R. Now, in order to compute a new geodesic chain corresponding to an edge $e = e_{\alpha(i)}$, we first compute the (l,k)-hull of the set of points corresponding to the geodesic chain bends of the previously embedded edges, and then, depending on whether

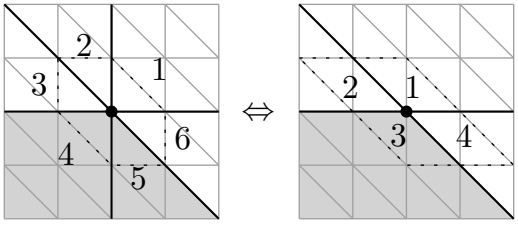


Figure 4.15: Sheared hexagonal grid and shallow orthogonal grid.

[5] Essentially, this is yet again due to the triangle inequality.

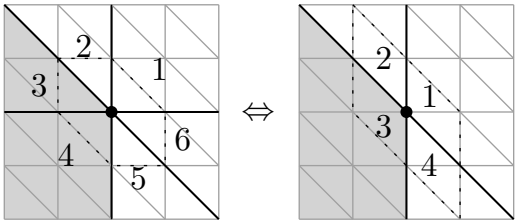


Figure 4.16: Sheared hexagonal grid and steep orthogonal grid.

we have an upward edge, a shallow downward edge or a steep downward edge, we translate this (l, k)-hull one unit to the top and one unit to the left, in case of an upward edge, or we translate the (l, k)-hull one unit to the top, in case of a shallow downward edge, or we translate the (l, k)-hull one unit to the right, in case of a steep downward edge.

More formally, we proceed as follows. Let B_i be the set of endpoints of all straight-line segments used between the columns c and c' in the embedding computed so far, and consider the edge $e = e_{\alpha(i)}$. Let $p_{\alpha(i)}$ be the current endpoint of $e_{\alpha(i)}$ on c, that is, the rightmost point of the geodesic chain corresponding to $e_{\alpha(i)}$ embedded so far. Furthermore, let $v \in V$ be the lastly considered event on c', which means that edges left of c' having v as their endpoint have already been embedded, and thus are completed edges. We would like to embed the fraction of e between c and c'. We distinguish two cases:

1. Suppose e does not end on c'. If e is an upward edge, we embed it on the (5,6)-hull of the point set $(B_i \cup \{v\} \setminus \{e^-\})^{\nwarrow} \cup \{p_{\alpha(i)}\}$ between c and c'. If, however, e is a downward edge not ending on c', we embed it on the (4,5)-hull of the point set $(B_i \cup \{v\} \setminus \{e^-\})^{\uparrow} \cup \{p_{\alpha(i)}\}$, if it is shallow, and on the (3,4)-hull of the point set $(B_i \cup \{v\} \setminus \{e^-\})^{\rightarrow} \cup \{p_{\alpha(i)}\}$, if it is steep. Intuitively, we embed the fraction of $e = e_{\alpha(i)}$ between c and c' such that it is *directly* above all previously embedded edges $e_{\alpha(j)}$ between c and c', for $j < i$.

2. Suppose e ends on c'. We call an edge $uv \in E$ *strictly shallow*, if it is a straight-line edge which leaves u diagonally to the right, that is, it exclusively runs along the grid line that connects the fifth and sixth sextant corresponding to u. If e is an upward edge, and if e^+ is the left endpoint of two steep downward or three downward edges starting in e^+, or if e^+ is the left endpoint of one steep and one strictly shallow downward edge starting in e^+, or if e is the second of two upward edges ending in e^+, then we embed the fraction of e on the (5,6)-hull of the point set $(B_i \setminus \{e^-\})^{\nwarrow} \cup \{p_{\alpha(i)}, q_e, e^+, \}$, where q_e is the grid point one unit left of e^+. Otherwise, we embed it on the (5,6)-hull of the point set $(B_i \setminus \{e^-\})^{\nwarrow} \cup \{p_{\alpha(i)}, e^+\}$. This special treatment is necessary to reserve sufficient space required for the embeddings of (yet untouched) edges that start in e^+. Similarly, if e is a steep downward edge, and if e is the only downward edge ending in e^+, then we embed it on the (3,4)-hull of the point set $(B_i \setminus \{e^-\})^{\rightarrow} \cup \{p_{\alpha(i)}, q_{ne}, e^+\}$, where q_{ne} is the grid point top left of e^+. Otherwise, we embed it on the (3,4)-hull of the point set $(B_i \setminus \{e^-\})^{\rightarrow} \cup \{p_{\alpha(i)}, e^+\}$. If e is a shallow downward edge, we embed it on the (4,5)-hull of the point set $(B_i \setminus \{e^-\})^{\uparrow} \cup \{p_{\alpha(i)}, e^+\}$. Here, no further case differentiation is required, since shallow downward edges may not occupy vertical grid-line segments that are possibly needed for edges starting in e^+. As already noted further above, this is due to shallow downward geodesics running exclusively on the shallow (orthogonal) grid, as a consequence of extending Lemma 4.3.2, and thus they cannot possibly occupy vertical grid-line segments. As a result, shallow downward edges ending in e^+ do not interfere with any edges starting in e^+, and thus impose no further restrictions on the embedding.

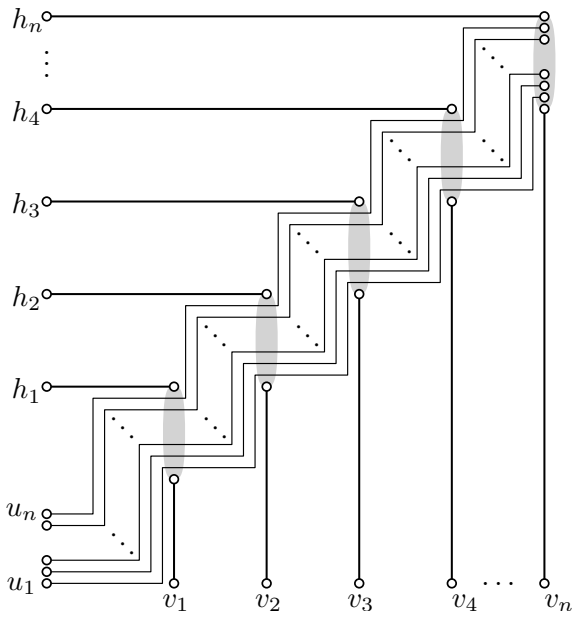


Figure 4.17: An instance of Labeled Geodesic PSE consisting of $3n$ edges and $2n^2$ edge bends.

By embedding edges this way, it is also guaranteed that they are embedded as geodesics. Additionally, the resulting embedding does not introduce any edge crossings. To see this, recall that G admissible, and thus we can embed the edges as described above, which results in no grid-line segment being used more than once. Moreover, since the instance is sparse and since any planar graph has at most $3n - 6$ edges, we do not run out of space when embedding the edges based on a translation of the k-hull of previously embedded edges by one unit to the top and one unit to the left (upward geodesics), or one unit to the top (shallow downward geodesics), or one unit to the right (steep downward geodesics). Finally, at all times, we embed edges respecting the order given by $\prec$.

In order to compute the k-hull for the current step, we walk along the k-hull of the previous step in linear time (instead of computing the k-hull from scratch every time, that is, instead of computing the k-hull by using the endpoints of all edges embedded so far) and extend it where needed. The total time complexity for this is proportional to the number of bends in the embedding. Note that each bend can be attributed to a vertex of the graph. Thus, since there are n vertices, each edge bends at most n times, which results in a total of at most $(3n - 6)n \in \mathcal{O}(n^2)$ bends.

Another way to describe the embedding of edges is as follows. By l_c we denote the list of partial edges $e_{\alpha(1)}, e_{\alpha(2)}, \ldots, e_{\alpha(k)}$ stored when the sweep line is located at column c. Recall that we have sorted this list according to $\prec$. Suppose furthermore that there are vertices situated on c that are incident to a total of k' edges starting at c to the right. Then, due to the local transitivity of $\prec$, the order of partial edges stored at column c' is preserved, since we merged k' untouched edges according to $\prec$ into the list of partial edges l_c, obtaining a new list, say, $l_{c'}$. That way, as a first step, we can straight-line embed all edges in $l_{c'}$ between c and c'. Note that this of course may easily result in edge segments that are no longer situated on the grid. As a second step, from bottom to top, we then "pull" those straight-line segments between c and c' onto the grid, introducing edge bends, while preserving the order given by $\prec$.

It remains to show that the algorithm is worst-case optimal. Consider the graph embedding shown in Figure 4.17. It shows an instance consisting of n horizontal edges $h_1, \ldots, h_n$ and n vertical edges $v_1, \ldots, v_n$, and an additional n upward edges $u_1, \ldots, u_n$. Each upward edge is strictly below every horizontal edge, and each upward edge is strictly above every vertical edge. Too see that as many as $2n^2$ edge bends are needed, consider the following. For $1 \leqslant i \leqslant n$, each of the n upward edges u_i needs to pass through all n corridors that are shown in gray color in the figure. This is due to the arrangement of the n pairs (h_i, v_i). Every such corridor introduces each edge to bend twice. Since there are n upward edges and n corridors, this results in $2n^2 \in \Omega(n^2)$ overall edge bends. Thus, the algorithm is worst-case optimal. □

At this point of the thesis, our initial plan was to show how to efficiently compute a combinatorial geodesic embedding of a given graph $G = (V, E)$. The idea is to first compute a kind of preliminary embedding and, based on this embedding, we can compute a full geodesic embedding using the algorithm from above. More precisely, we compute a *pre-embedding* of G, which is a directed graph $\Pi = (E, A)$ on the set of edges of G that only contains edges (e, f) such that e is strictly below f. Furthermore, the acyclicity of Π is equivalent to the existence of a combinatorial embedding of G.

Katz *et al.* [KKRW10] present an algorithm to compute a combinatorial geodesic embedding of a given graph which crucially relies on the implication rules. However, as we have seen above, the implication rules presented in their work do not hold, and this renders the algorithm flawed as well. We cannot simply use their approach to a compute a combinatorial embedding of a given graph with respect to the hexagonal grid. Since this difficulty had not come to our attention until the very end of this thesis, we are unable to present such an algorithm in detail. Nevertheless, we outline an idea.

Suppose we are given a graph $G = (V, E)$. Note that the implication rules (U3)-(U6) and (D3)-(D6) require certain geometrical information to be given, for example information regarding the overlapping of edges, information regarding the containment of edges in the k-critical set of other edges, and so on. Given that information, we can express these implication rules as simple implications, for instance $f' \succ f \;\Rightarrow\; f' \succ e$ in the case of rule (U3). Using this observation, we model the requirement that $\prec$ satisfies the implication rules as an instance φ of 2-SAT, by defining for any pair of overlapping edges i, j a variable x_{ij} such that

$$x_{ij} := \begin{cases} 1, & \text{if } i \prec j, \\ 0, & \text{if } j \prec i. \end{cases}$$

Suppose we are given a satisfying assignment A for φ. We would like to conclude that the relation $\prec$ corresponding to A is a combinatorial geodesic embedding of G. By construction we have that any pair of overlapping edges is comparable with respect to $\prec$ and the implication rules hold. However, the definition of a combinatorial geodesic embedding also requires local transitivity to hold, and this is not obviously guaranteed by $\prec$. Nevertheless, we claim that local transitivity is also fulfilled by $\prec$. To prove this, one would need to show that the implication rules enforce local transitivity. The basic idea is described by Katz *et al.* [KKRW10]. However, their approach needs to be adapted according to corrected set of implication rules. Due to time constraints, this is not part of this thesis.

Next, assuming we are given a combinatorial geodesic embedding of G, we can apply the algorithm from Theorem 4.3.1 to find a geodesic embedding of G on the hexagonal grid. Finally, since 2-SAT can be solved efficiently and since the algorithm from Theorem 4.3.1 works in $\mathcal{O}(n^2)$ time, we can efficiently find a geodesic embedding of a given graph G.

Theorem 4.3.2. *Given a 6-planar graph G, we can efficiently compute a geodesic embedding of G on the hexagonal grid, if one exists.*

Again, to prove this, one would need to show that local transitivity holds for the corrected set of implication rules.

5. Conclusion

In this thesis, we studied the drawing or embedding of graphs on non-standard grids. One motivating fact was that angles formed by two edges incident to the same vertex may not become arbitrarily small, which is beneficial in terms of visual impression or attractiveness of a drawing. While there is a huge variety of results related to the orthogonal grid—the standard grid—, only few results are known regarding the grids that we considered.

In particular, we first considered the honeycomb grid and presented algorithms that produce h-v drawings of complete binary trees and sector drawings of trees having degree at most 3. We used an iterative approach based on an idea due to Crescenzi *et al.* [CP97] to produce drawings of complete binary trees according to the h-v convention. The construction of sector drawings was based on an inductive approach. For both drawing styles we analyzed the area occupied by a corresponding drawing. Our h-v drawings require an area that is in $\mathcal{O}(n \log n)$, and we have seen that an h-v drawing which requires an area of A on the orthogonal grid induces an h-v drawing on the honeycomb grid occupying an area of $8A$. The algorithm to compute sector drawings requires $\mathcal{O}(n^2)$ area. More precisely, the area occupied by sector drawings is bounded by $0.731n^2$, for $n \geqslant 3$.

Afterwards, we studied fundamentals of the hexagonal grid, also referred to as the triangular grid, and analyzed the computational complexity of two problems that were first introduced by Katz *et al.* [KKRW10] with respect to the orthonal grid. Katz *et al.* showed that the problem of Geodesic Point-Set Embeddability (Geodesic PSE) is NP-hard on the orthogonal grid. By shearing the hexagonal grid we were able to obtain a grid structure—which we referred to as the sheared grid—that resembled the orthogonal grid. This allowed us to use almost the same approach as Katz *et al.* to prove the problem to be NP-hard on the hexagonal grid as well. Then, again using ideas due to Katz *et al.*, we were able to easily prove the problem of Labeled Geodesic Point-Set Embeddability (Labeled Geodesic PSE) to be NP-hard on the hexagonal grid, as is the case on the orthogonal grid. Finally, we tried to adapt their approach to solving sparse instances efficiently to the hexagonal grid. Unfortunately, this did not work out quite so well, since the implication rules in their original form [KKRW10] were erroneous, and thus the algorithm to compute combinatorial embeddings, which crucially rests upon the implication rules, did not work. It is also unfortunate that this first came to our attention in a very late stage of this thesis. Nevertheless, we were able to correct the implication rules and

formally prove their correctness. While we were not able to fix the aforesaid algorithm due to time constraints, we could adapt an algorithm that, given a combinatorial geodesic embedding of a graph, finds a geodesic embedding of that graph in $\mathcal{O}(n^2)$ time, which is worst-case optimal.

As for the future, a certainly interesting task will be to fix the algorithm for computing combinatorial embeddings by considering the corrected set of implication rules. Furthermore, while the implication rules have been corrected, it is questionable whether they are—in a sense—complete, that is, whether the current set of implication rules covers all possible arrangements of all types of edges.

Further future work may concern the minimization of edge bends introduced in the drawings we presented. While on the honeycomb grid every edge inherently bends with every additional line segment it occupies, the number of bends is clearly tied to the length of the respective edge, and thus minimizing edge lengths will also reduce the number of bends introduced in a drawing. Katz *et al.* [KKRW10] also comment on bend minimization as an optimization criterion, and particularly regarding sparse instances of Labeled Geodesic PSE, they conclude by stating that they do not know whether an efficient algorithm for bend minimization is within reach. They note that their sweepline algorithm does not even provide an embedding with the minimum number of edge bends for the given combinatorial embedding. This obviously extends to the hexagonal and sheared hexagonal grids, since the algorithm presented in this work bases on the same essentials.

Bibliography

[AB04] Shabnam Aziza and Therese C. Biedl. Hexagonal grid drawings: Algorithms and lower bounds, 2004.

[AP61] L. Auslander and S.V. Parter. On embedding graphs in the plane. *Journal of Applied Mathematics and Mechanics*, pages 517–523, 1961.

[BBB+09] Christian Bachmaier, Franz J. Brandenburg, Wolfgang Brunner, Andreas Hofmeier, Marco Matzeder, and Thomas Unfried. Graph drawing. chapter Tree Drawings on the Hexagonal Grid, pages 372–383. Springer-Verlag, Berlin, Heidelberg, 2009.

[BBC11] Melanie Badent, Ulrik Brandes, and Sabine Cornelsen. More canonical ordering. *Journal of Graph Algorithms and Applications*, 15(1):97–126, 2011.

[BDHS97] Prosenjit Bose, Alice Dean, Joan Hutchinson, and Thomas Shermer. On rectangle visibility graphs, 1997.

[Bie96] Therese C. Biedl. Optimal orthogonal drawings of connected plane graphs. In *Proceedings of the 8th Canadian Conference on Computational Geometry*, pages 306–311. Carleton University Press, 1996.

[BK97] Therese C. Biedl and Michael Kaufmann. Area-efficient static and incremental graph drawings, 1997.

[BK98] Therese C. Biedl and Goos Kant. A better heuristic for orthogonal graph drawings. In *Computational Geometry: Theory and Applications*, pages 24–35. Springer-Verlag, 1998.

[BLV93] Giuseppe Di Battista, Giuseppe Liotta, and Francesco Vargiu. Spirality of orthogonal representations and optimal drawings of series-parallel graphs and 3-planar graphs (extended abstract). In *Proceedings of the 3rd Workshop on Algorithms and Data Structures*, Workshop on Algorithms and Data Structures 1993, pages 151–162, London, UK, 1993. Springer-Verlag.

[Cha85] Bernard Chazelle. Slimming down search structures: A functional approach to algorithm design. In *26th Annual Symposium on Foundations of Computer Science*, pages 165–174, 1985.

[CK97] Marek Chrobak and Goos Kant. Convex grid drawings of 3-connected planar graphs. *International Journal of Computational Geometry and Applications*, 7:211–223, 1997.

[CK12] Sabine Cornelsen and Andreas Karrenbauer. Accelerated bend minimization. In *Proceedings of the 19th International Conference on Graph Drawing*, Graph Drawing 2011, pages 111–122, Berlin, Heidelberg, 2012. Springer-Verlag.

[CN95] Marek Chrobak and Shin-ichi Nakano. Minimum-width grid drawings of plane graphs. *Proceedings of Graph Drawing 1996, Lecture Notes in Computer Science*, 10:104–110, 1995.

[CON85] Norishige Chiba, Kazunori Onoguchi, and Takao Nishizeki. Drawing planar graphs nicely. *Acta Informatica*, 1985.

[CP95] Marek Chrobak and T.H. Payne. A linear-time algorithm for drawing a planar graph on a grid. *Information Processing Letters*, 54(4):241–246, 1995.

[CP97] Pilu Crescenzi and Paolo Penna. Minimum-area h-v drawings of complete binary trees. In *Proceedings of the 5th Symposium on Graph Drawing*, volume 1353 of *Lecture Notes in Computer Science*, pages 371–382, 1997.

[DBETT99] Giuseppe Di Battista, Peter Eades, Roberto Tamassia, and Ioannis G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs.* Prentice Hall, Upper Saddle River, New Jersey, 1999.

[DBTT92] Giuseppe Di Battista, Roberto Tamassia, and Ioannis G. Tollis. Constrained visibility representations of graphs. *Information Processing Letters*, 41(1):1–7, 1992.

[dFPP88] Hubert de Fraysseix, János Pach, and Richard Pollack. Small sets supporting fáry embeddings of planar graphs. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing*, STOC '88, pages 426–433, New York, NY, USA, 1988. ACM.

[dFPP90] Hubert de Fraysseix, János Pach, and Richard Pollack. How to draw a planar graph on a grid. *Combinatorica*, 10:41–51, 1990.

[EG94] Shimon Even and Gilad Granot. Rectilinear planar drawings with few bends in each edge. Technical report, Computer Science Department, Technion, Israel Institute of Technology, 1994.

[Fár48] István Fáry. On straight line representation of planar graphs. *Acta Universitatis Szegediensis. Acta Scientiarum Mathematicarum*, 11:229–233, 1948.

[FHKR92] Martin Fürer, Xin He, Ming-Yang Kao, and Balaji Raghavachari. $\mathcal{O}(n \log \log n)$-work parallel algorithms for straight-line grid embeddings of planar graphs. In *Proceedings of the 4th Annual ACM Symposium on Parallel Algorithms & Architectures*, Symposium on Parallel Algorithms & Architectures 1992, pages 410–419, New York, NY, USA, 1992. ACM.

[FKK97] Ulrich Fößmeier, Goos Kant, and Michael Kaufmann. 2-visibility drawings of planar graphs. In Stephen North, editor, *Graph Drawing*, volume 1190 of *Lecture Notes in Computer Science*, pages 155–168. Springer Berlin / Heidelberg, 1997.

[GM98] Carsten Gutwenger and Petra Mutzel. Planar polyline drawings with good angular resolution. In *Proceedings of Graph Drawing 1998, Lecture Notes in Computer Science*, pages 167–182. Springer-Verlag, 1998.

[GT94] Ashim Garg and Roberto Tamassia. On the computational complexity of upward and rectilinear planarity testing (extended abstract), 1994.

[GT97] Ashim Garg and Roberto Tamassia. A new minimum cost flow algorithm with applications to graph drawing. In *Proceedings of Graph Drawing 1996, Lecture Notes in Computer Science*, pages 201–216. Springer-Verlag, 1997.

[GW98] Michael T. Goodrich and Christopher G. Wagner. A framework for drawing planar graphs with curves and polylines. In *Proceedings of the 6th International Symposium on Graph Drawing*, Graph Drawing 1998, pages 153–166, London, UK, 1998. Springer-Verlag.

[He95] Xin He. Grid embedding of internally triangulated plane graphs without non-empty triangles. Technical report, 1995.

[He97] Xin He. Grid embedding of 4-connected plane graphs. *Discrete & Computational Geometry*, 17:339–358, 1997.

[Kan92a] Goos Kant. Drawing planar graphs using the *lmc*-ordering. In *Proceedings of the 33rd Annual Symposium on Foundations of Computer Science*, pages 101–110, Washington, DC, USA, 1992. IEEE Computer Society.

[Kan92b] Goos Kant. Hexagonal grid drawings, 1992.

[Kan94] Goos Kant. A more compact visibility representation. In *Proceedings of the 19th International Workshop on Graph-Theoretical Concepts in Computer Science*, pages 411–424. Springer-Verlag, 1994.

[Kan96] Goos Kant. Drawing planar graphs using the canonical ordering. *Algorithmica*, 16:4–32, 1996.

[KH94] Goos Kant and Xin He. Two algorithms for finding rectangular duals of planar graphs. In Jan van Leeuwen, editor, *Graph-Theoretic Concepts in Computer Science*, volume 790 of *Lecture Notes in Computer Science*, pages 396–410. Springer Berlin / Heidelberg, 1994.

[KH97] Goos Kant and Xin He. Regular edge labeling of 4-connected plane graphs and its applications in graph drawing problems. *Theoretical Computer Science*, 172(1–2):175–193, 1997.

[KKRW10] Bastian Katz, Marcus Krug, Ignaz Rutter, and Alexander Wolff. Manhattan-geodesic embedding of planar graphs. In David Eppstein and Emden Gansner, editors, *Graph Drawing*, volume 5849 of *Lecture Notes in Computer Science*, pages 207–218. 2010.

[Knu63] Donald E. Knuth. Computer-drawn flowcharts. *Communications of the ACM*, 6(9):555–563, 1963.

[LEC75] Abraham Lempel, Shimon Even, and I. Cederbaum. An algorithm for planarity testing of graphs. *Theory of Graphs: International Symposium, Rome 1966*, C-24(2):113–121, 1975.

[LLS05] Ching-Chi Lin, Hsueh-I Lu, and I-Fan Sun. Improved compact visibility representation of planar graph via schnyder's realizer. *SIAM Journal on Discrete Mathematics*, 18(1):19–29, 2005.

[LRT79] Richard J. Lipton, Donald J. Rose, and Robert E. Tarjan. Generalized nested dissection. *SIAM Journal on Numerical Analysis*, 16(2):346–358, 1979.

[OVW78] R. M. Otten and J. G. Van Wijk. *Graph Representations in Interactive Layout Design*, pages 914–918. 1978.

[PT95] Achilleas Papakostas and Ioannis G. Tollis. Improved algorithms and bounds for orthogonal drawings. In *Proceedings of the DIMACS International Workshop on Graph Drawing*, Graph Drawing 1994, pages 40–51, London, UK, 1995. Springer-Verlag.

[PT96] Achilleas Papakostas and Ioannis G. Tollis. Algorithms for area-efficient orthogonal drawings. *Computational Geometry: Theory and Applications*, 9:83–110, 1996.

[RNN96] Md. Saidur Rahman, Shin-ichi Nakano, and Takao Nishizeki. Rectangular grid drawings of plane graphs. In *Computing and Combinatorics: Second Annual International Conference, COCOON 1996, Hong Kong*, 1996.

[RT86] Pierre Rosenstiehl and Robert E. Tarjan. Rectilinear planar layouts and bipolar orientations of planar graphs. *Discrete & Computational Geometry*, 1:343–353, 1986.

[Sch89] Walter Schnyder. Planar graphs and poset dimension. *Order*, 5:323–343, 1989.

[Sch90] Walter Schnyder. Embedding planar graphs on the grid. In *Proceedings of the 1st Annual ACM-SIAM Symposium on Discrete Algorithms*, Symposium on Discrete Algorithms 1990, pages 138–148, Philadelphia, PA, USA, 1990. Society for Industrial and Applied Mathematics.

[Sch95] Markus Schäffter. Drawing graphs on rectangular grids. *Discrete Applied Mathematics*, 63(1):75–89, 1995.

[SR83] Kenneth J. Supowit and Edward M. Reingold. The complexity of drawing rrees nicely. *Acta Informatica*, 18:377–392, 1983.

[ST92] Walter Schnyder and William T. Trotter. Convex embeddings of 3-connected plane graphs. *Abstracts of the American Mathematical Society*, 13:502, 1992.

[Ste51] S. K. Stein. Convex maps. *Proceedings of the American Mathematical Society*, 2:464–466, 1951.

[Sto84] James A. Storer. On minimal-node-cost planar embeddings. *Networks*, 14:181–212, 1984.

[Tam87] Roberto Tamassia. On embedding a graph in the grid with the minimum number of bends. *SIAM Journal on Computing*, 16:421–444, 1987.

[Tam90] Roberto Tamassia. Planar orthogonal drawings of graphs, 1990.

[TDBB88] Roberto Tamassia, Giuseppe Di Battista, and Carlo Batini. Automatic graph drawing and readability of diagrams. *IEEE Transactions on Systems, Man, and Cybernetics*, 18:61–79, 1988.

[Tho80] Carsten Thomassen. Planarity and duality of finite and infinite planar graphs. *Journal of Combinatorial Theory*, 1980.

[TT86] Roberto Tamassia and Ioannis G. Tollis. A unified approach to visibility representations of planar graphs. *Discrete & Computational Geometry*, 1:321–341, 1986.

[TT87] Roberto Tamassia and Ioannis G. Tollis. Efficient embedding of planar graphs in linear time. In *Proceedings of the IEEE International Symposium on Circuits and Systems*, pages 495–498, 1987.

[TT89] Roberto Tamassia and Ioannis G. Tollis. Planar grid embedding in linear time. *IEEE Transactions on Circuits and Systems*, 36(9):1230–1234, 1989.

[Tut60] William T. Tutte. Convex representations of graphs. *Proceedings of the London Mathematical Society*, 10:304–320, 1960.

[Tut63] William T. Tutte. How to draw a graph. *Proceedings of the London Mathematical Society*, s3-13:743–767, 1963.

[Val81] Leslie G. Valiant. Universality considerations in vlsi circuits. *IEEE Transactions on Computers*, 30(2):135–140, 1981.

[Wag36] Klaus Wagner. Bemerkungen zum vierfarbenproblem. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 46:26–33, 1936.

[Woo82] D. Woods. *Drawing Planar Graphs*. PhD thesis, Computer Science Department, Stanford University, Stanford, CA, 1982.

[ZH05] Huaming Zhang and Xin He. Improved visibility representation of plane graphs. *Computational Geometry: Theory and Applications*, 30(1):29–39, 2005.