

GRAPH DRAWING

Roberto Tamassia

INTRODUCTION

Graph drawing addresses the problem of constructing geometric representations of graphs, and has important applications to key computer technologies such as software engineering, database systems, visual interfaces, and computer-aided-design. Research on graph drawing has been conducted within several diverse areas, including discrete mathematics (topological graph theory, geometric graph theory, order theory), algorithmics (graph algorithms, data structures, computational geometry, VLSI), and human-computer interaction (visual languages, graphical user interfaces, software visualization). This chapter overviews aspects of graph drawing that are especially relevant to computational geometry. Basic definitions on drawings and their properties are given in Section 1.1. Bounds on geometric and topological properties of drawings (e.g., area and crossings) are presented in Section 1.2. Section 1.3 deals with the time complexity of fundamental graph drawing problems. General techniques for drawing graphs are surveyed in Section 1.4. Section 1.5 covers selected topics that have recently attracted considerable research interest.

1.1 DRAWINGS AND THEIR PROPERTIES

TYPES OF GRAPHS

First, we define some terminology on graphs pertinent to graph drawing.

GLOSSARY

n: number of vertices.

m: number of edges.

d: maximum vertex degree (i.e., number of incident edges).

degree-*k* graph: graph with maximum degree $d \leq k$.

digraph: directed graph, i.e., graph with directed edges (drawn as arrows).

acyclic digraph: without directed cycles.

transitive edge: edge (u, v) of a digraph is transitive if there is a directed path from u to v not containing edge (u, v) .

reduced digraph: without transitive edges.

- source:** vertex of a digraph without incoming edges.
- sink:** vertex of a digraph without outgoing edges.
- st-digraph:** acyclic digraph with exactly one source and one sink, joined by an edge (also called bipolar digraph).
- connected graph:** any two vertices are joined by a path.
- biconnected graph:** any two vertices are joined by two vertex-disjoint paths.
- triconnected graph:** any two vertices are joined by three vertex-disjoint paths.
- tree:** connected graph without cycles.
- rooted tree:** directed tree with a distinguished vertex, called the root, such that all the edges are directed towards the root.
- binary tree:** rooted tree where each vertex has at most two incoming edges.
- layered (di)graph:** the vertices are partitioned into sets, called layers. A rooted tree can be viewed as a layered digraph where the layers are sets of vertices at the same distance from the root.
- k-layered (di)graph:** layered (di)graph with k layers.

TYPES OF DRAWINGS

In a drawing of a graph, vertices are represented by points (or by geometric figures such as circles or rectangles) and edges are represented by curves. Except for Section 1.5, which covers three-dimensional drawings, we consider drawings in the plane.

GLOSSARY

- polyline drawing:** each edge is a polygonal chain (see Fig. 1.1(a)).
- straight-line drawing:** each edge is a straight-line segment (see Fig. 1.1(b)).
- orthogonal drawing:** each edge is a chain of horizontal and vertical segments (see Fig. 1.1(c)).
- bend:** in a polyline drawing, point where two segments forming the same edge meet (see Fig. 1.1(a)).
- crossing:** point where two edges intersect (see Fig. 1.1(b)).
- grid drawing:** polyline drawing such that vertices, crossings and bends have integer coordinates.
- planar drawing:** no two edges cross (see Fig. 1.1(d)).
- planar (di)graph:** admits a planar drawing.
- embedded (di)graph:** planar (di)graph with a prespecified topological embedding (i.e., set of faces), which must be preserved in the drawing.
- upward drawing:** drawing of a digraph where each edge is monotonically non-decreasing in the vertical direction (see Fig. 1.1(d)).
- upward planar digraph:** admits an upward planar drawing.
- layered drawing:** drawing of a layered graph such that vertices in the same layer are horizontally aligned (also called hierarchical drawing).

face: a region of the plane bounded by vertices and edges of a planar drawing.

convex drawing: planar straight-line drawing such that the boundary of each face is a convex polygon.

visibility drawing: drawing of a graph based on a geometric visibility relation. E.g., the vertices are drawn as horizontal segments, and the edges are associated with vertically visible segments.

proximity drawing: drawing of a graph based on a geometric proximity relation. E.g., a tree is drawn as the Euclidean minimum spanning tree of a set of points.

dominance drawing: upward drawing of an acyclic digraph such that there exists a directed path from vertex u to vertex v if and only if $x(u) \leq x(v)$ and $y(u) \leq y(v)$.

hv-drawing: upward orthogonal straight-line drawing of a binary tree such that the drawings of the subtrees of each node are separated by a horizontal or vertical line.

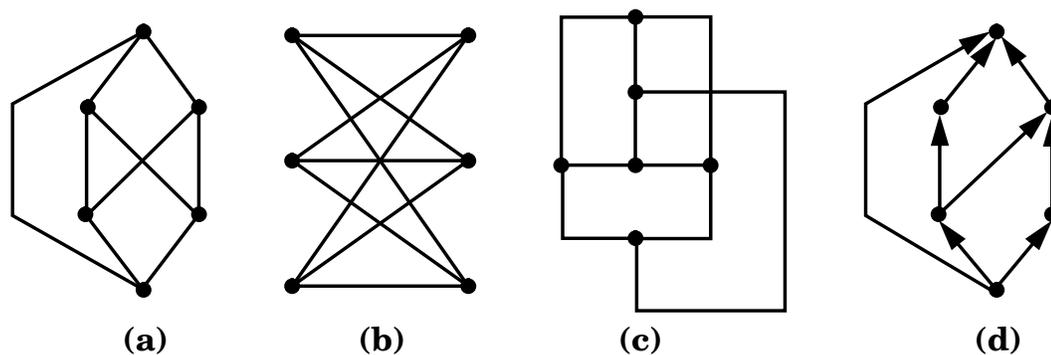


Figure 1.1: Types of drawings: (a) polyline drawing of $K_{3,3}$; (b) straight-line drawing of $K_{3,3}$; (c) orthogonal drawing of $K_{3,3}$; (d) planar upward drawing of an acyclic digraph.

Straight-line and orthogonal drawings are special cases of polyline drawings. Polylines provide great flexibility since they can approximate drawings with curved edges. However, edges with more than two or three bends may be difficult to “follow” for the eye. Also, a system that supports editing of polyline drawings is more complicated than one limited to straight-line drawings. Hence, depending on the application, polyline or straight-line drawings may be preferred. If vertices are represented by points, orthogonal drawings exist only for graphs of maximum vertex degree 4.

PROPERTIES OF DRAWINGS

GLOSSARY

- crossings* (χ):** total number of crossings.
- area*:** area of the smallest convex polygon covering the drawing.
- total edge length*:** sum of the lengths of the edges.
- maximum edge length*:** maximum length of an edge.
- total number of bends*:** total number of bends on the edges of a polyline drawing.
- maximum number of bends*:** maximum number of bends on an edge of a polyline drawing.
- angular resolution* (ρ):** smallest angle formed by two edges, or segments of edges, incident on the same vertex or bend, in a polyline drawing.
- aspect-ratio*:** ratio of the longest to the shortest side of the smallest rectangle with horizontal and vertical sides covering the drawing.

There are infinitely many drawings for a graph. In drawing a graph, we would like to take into account a variety of properties. For example, planarity and the display of symmetries are highly desirable in visualization applications. Also, it is customary to display trees and acyclic digraphs with upward drawings. In general, to avoid wasting valuable space on a page or a computer screen, it is important to keep the area of the drawing small. In this scenario, many graph drawing problems can be formalized as multi-objective optimization problems (e.g., construct a drawing with minimum area and minimum number of crossings), so that trade-offs are inherent in solving them. Typically, it is desirable to maximize the angular resolution and to minimize the other measures.

The following examples illustrate two typical trade-offs in graph drawing problems. Figure 1.2(a–b) shows two drawings of K_4 , the complete graph on four vertices. The drawing of part (a) is planar, while the drawing of part (b) maximizes symmetries. It can be shown that no drawing of K_4 is optimal with respect to both criteria, i.e., the maximum number of symmetries cannot be achieved by a planar drawing. Figure 1.2(c–d), shows two drawing of the same acyclic digraph G . The drawing of part (c) is upward, while the drawing of part (d) is planar. It can be shown that there is no drawing of G which is both planar and upward.

1.2 BOUNDS ON DRAWING PROPERTIES

For various classes of graphs and drawing types, many universal/existential upper and lower bounds for specific drawing properties have been discovered. Such bounds typically exhibit trade-offs between drawing properties. A universal bound applies to all the graphs of a given class. An existential bound applies to infinitely many graphs of the class.

Whenever we give bounds on the area or edge length, we assume that the drawing is constrained by some resolution rule that prevents it from being arbitrarily scaled down (e.g., requiring a grid drawing, or a minimum unit distance between any two vertices).

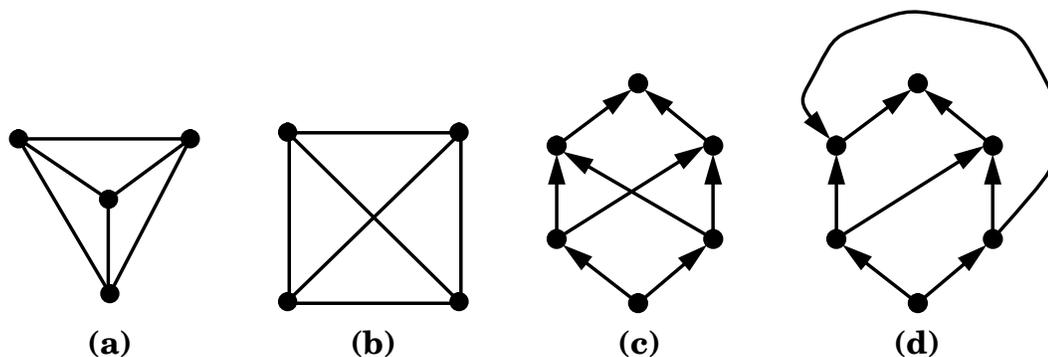


Figure 1.2: (a–b) Tradeoff between planarity and symmetry in drawing K_4 . (c–d) Tradeoff between planarity and upwardness in drawing an acyclic digraph G .

BOUNDS ON THE AREA

Table 1.1 summarizes selected universal upper bounds and existential lower bounds on the area of drawings of graphs.

Table 1.1: Universal upper bounds and existential lower bounds on the area of drawings of graphs. We denote with a an arbitrary constant such that $0 \leq a < 1$. We denote with b and c fixed constants such that $1 < b < c$.

Class of Graphs	Drawing Type	Area		Source
rooted tree	upward planar straight-line grid	$O(n \log n)$	$\Omega(n)$	[17, 98]
rooted tree	strictly upward planar straight-line grid	$O(n \log n)$	$\Omega(n \log n)$	[17]
degree- $O(n^a)$ rooted tree	upward planar polyline grid	$O(n)$	$\Omega(n)$	[56]
binary tree	upward planar orthogonal grid	$O(n \log \log n)$	$\Omega(n \log \log n)$	[56]
tree	planar straight-line grid	$O(n \log n)$	$\Omega(n)$	[17, 98]
degree- $O(n^a)$ tree	planar polyline grid	$O(n)$	$\Omega(n)$	[56]
degree-4 tree	planar orthogonal grid	$O(n)$	$\Omega(n)$	[114, 81]
planar graph	planar polyline grid	$O(n^2)$	$\Omega(n^2)$	[31, 33, 73]
planar graph	planar straight-line		$\Omega(c^{\rho n})$	[59]
planar graph	planar straight-line grid	$O(n^2)$	$\Omega(n^2)$	[21, 96]
triconnected planar graph	planar straight-line convex grid	$O(n^2)$	$\Omega(n^2)$	[73]
planar graph	planar orthogonal grid	$O(n^2)$	$\Omega(n^2)$	[7, 73, 103, 108]
planar degree-4 graph	orthogonal grid	$O(n \log^2 n)$	$\Omega(n \log n)$	[114, 81, 6]
upward planar digraph	upward planar grid straight-line	$\Omega(c^n)$	$O(b^n)$	[2, 33, 58]
reduced planar st -digraph	upward planar grid straight-line dominance	$O(n^2)$	$\Omega(n^2)$	[33]
upward planar digraph	upward planar grid polyline	$O(n^2)$	$\Omega(n^2)$	[31, 33]
general graph	polyline grid	$O((n + \chi)^2)$	$\Omega(n + \chi)$	

The effect of bends on the area requirement is dual. On one hand, bends occupy space and hence negatively affect the area. On the other hand, bends may help in routing edges without using additional space. Linear or almost-linear bounds

on the area can be achieved for trees. See Table 1.4 for trade-offs between area and aspect-ratio in drawings of trees. Planar graphs admit planar drawings with quadratic area. However, the area requirement of planar straight-line drawings may be exponential if high angular resolution is also desired. Almost linear area can be instead achieved in nonplanar drawings of planar graphs, which have applications to VLSI circuits. Upward planar drawings provide an interesting trade-off between area and the total number of bends. Indeed, unless the digraph is reduced, the area can become exponential if a straight-line drawing is required. A quadratic area bound is achieved only at the expense of a linear number of bends.

BOUNDS ON THE ANGULAR RESOLUTION

Table 1.2 summarizes selected universal lower bounds and existential upper bounds on the angular resolution of drawings of graphs.

Table 1.2: Universal lower bounds and existential upper bounds on the angular resolution of drawings of graphs. We denote with c a fixed constant such that $c > 1$.

Class of Graphs	Drawing Type	Angular Resolution		Source
general graph	straight-line	$\Omega(\frac{1}{d^2})$	$O(\frac{\log d}{d^2})$	[48]
planar graph	straight-line	$\Omega(\frac{1}{d})$	$O(\frac{1}{d})$	[48]
planar graph	planar straight-line	$\Omega(\frac{1}{cd})$	$O(\sqrt{\frac{\log d}{d^3}})$	[59, 87]

BOUNDS ON THE NUMBER OF BENDS

Table 1.3 summarizes selected universal upper bounds and existential lower bounds on the total and maximum number of bends in orthogonal drawings. Some bounds are stated for $n \geq 5$ or $n \geq 7$ because the maximum number of bends is at least 2 for K_4 and at least 3 for the skeleton graph of an octahedron, in any planar orthogonal drawing

TRADE-OFF BETWEEN AREA AND ASPECT-RATIO

The ability to construct area-efficient drawings is essential in practical visualization applications, where screen space is at a premium. However, achieving small area is not enough: e.g., it is easy to see that a drawing with high aspect-ratio may not be conveniently placed on a workstation screen, even if it has modest area. Hence, it is important to keep the aspect-ratio small. Ideally, one would like to obtain small area for any given aspect-ratio in a wide range. This would provide graphical user interfaces with the flexibility of fitting drawings in arbitrarily shaped windows.

Table 1.3: Orthogonal drawings: universal upper bounds and existential lower bounds on the total and maximum number of bends. Notes: † $n \geq 7$; ‡ $n \geq 5$.

Class of Graphs	Drawing Type	Total No. Bends		Max No. Bends		Source
		\leq	\geq	\leq	\geq	
degree-4 graph†	orthogonal	$\leq 2n + 2$	$\geq n$	≤ 2	≥ 2	[7]
planar degree-4 graph†	orthogonal planar	$\leq 2n + 2$	$\geq 2n - 2$	≤ 2	≥ 2	[7, 111]
embedded degree-4 graph	orthogonal planar	$\leq \frac{12}{5}n + 2$	$\geq 2n - 2$	≤ 3	≥ 3	[46, 86, 108, 111]
biconnected embedded degree-4 graph	orthogonal planar	$\leq 2n + 2$	$\geq 2n - 2$	≤ 3	≥ 3	[46, 86, 108, 111]
triconnected embedded degree-4 graph	orthogonal planar	$\leq \frac{7}{2}n + 4$	$\geq \frac{4}{3}(n - 1) + 2$	≤ 2	≥ 2	[73]
embedded degree-3 graph‡	orthogonal planar	$\leq \frac{1}{2}n + 1$	$\geq \frac{1}{2}n + 1$	≤ 1	≥ 1	[73, 85]

A variety of trade-offs for the area and aspect-ratio arise even when drawing graphs with a simple structure, such as trees. Table 1.4 summarizes selected universal bounds that can be simultaneously achieved on the area and the aspect-ratio of various types of drawings of trees.

Table 1.4: Universal upper bounds that can be simultaneously achieved for the area and aspect-ratio in drawings of trees. We denote with a an arbitrary constant such that $0 \leq a < 1$.

Class of Graphs	Drawing Type	Area	Aspect-Ratio	Source
rooted tree	upward planar straight-line layered grid	$O(n^2)$	$O(1)$	[90]
rooted tree	upward planar straight-line grid	$O(n \log n)$	$O(n / \log n)$	[17, 98]
rooted degree- $O(1)$ tree	upward planar polyline grid	$O(n)$	$O(n^a)$	[56]
binary tree	upward planar orthogonal grid	$O(n \log \log n)$	$O(n \log \log n / \log^2 n)$	[56]
degree-4 tree	orthogonal grid	$O(n)$	$O(1)$	[114, 81]
degree-4 tree	orthogonal grid with leaves on convex hull	$O(n \log n)$	$O(1)$	[11]

While upward planar straight-line drawings are the most natural way of visualizing rooted trees, the existing drawing techniques are unsatisfactory with respect to either the area requirement or the aspect ratio. The situation is similar for orthogonal drawings. Regarding polyline drawings, linear area can be achieved with a prescribed aspect ratio [56]. However, experiments show that this is done at the expense of a somehow aesthetically unappealing drawing.

For nonupward drawings of trees, linear area and optimal aspect ratio are possible for planar orthogonal drawings, and a small (logarithmic) amount of extra area is needed if the leaves are constrained to be on the convex hull of the drawing (e.g., pins on the boundary of a VLSI circuit). However, the above nonupward drawing methods do not seem to yield aesthetically pleasing drawings, and are suited more for VLSI layout than for visualization applications.

TRADE-OFF BETWEEN AREA AND ANGULAR RESOLUTION

Table 1.5 summarizes selected universal bounds that can be simultaneously achieved

on the area and the angular resolution of drawings of graphs.

Table 1.5: Universal asymptotic upper bounds for the area and lower bounds for the angular resolution that can be simultaneously achieved in drawings of graphs. We denote with b and c fixed constants such that $b > 1$ and $c > 1$.

Class of Graphs	Drawing Type	Area	Angular Resolution	Source
planar graph	straight-line	$O(d^6 n)$ **	$\Omega(\frac{1}{d^2})$	[48]
planar graph	straight-line	$O(d^3 n)$ **	$\Omega(\frac{1}{d})$	[48]
planar graph	planar straight-line grid	$O(n^2)$	$\Omega(\frac{1}{n^2})$	[21, 96]
planar graph	planar straight-line	$O(b^n)$	$\Omega(\frac{1}{c^d})$	[87]
planar graph	planar polyline grid	$O(n^2)$	$\Omega(\frac{1}{d})$	[73]

Universal lower bounds on the angular resolution exist that depend only on the degree of the graph. Also, substantially better bounds can be achieved by drawing a planar graph with bends or in a nonplanar way.

OPEN PROBLEMS

- Determine the area requirement of (upward) planar straight-line drawings of trees. There is currently an $O(\log n)$ gap between the known upper and lower bounds.
- Determine the area requirement of orthogonal (or, more generally, polyline) nonplanar drawings of planar graphs. There is currently an $O(\log n)$ gap between the known upper and lower bounds.
- Close the gap between the $\Omega(\frac{1}{d^2})$ universal lower bound and the $O(\frac{\log d}{d^2})$ existential upper bound on the angular resolution of straight-line drawings of general graphs.
- Close the gap between the $\Omega(\frac{1}{c^d})$ universal lower bound and the $O(\sqrt{\frac{\log d}{d^3}})$ existential upper bound on the angular resolution of planar straight-line drawings of planar graphs.
- Determine the best possible aspect-ratio and area that can be simultaneously achieved for (upward) planar straight-line and orthogonal drawings of trees.

1.3 COMPLEXITY OF GRAPH DRAWING PROBLEMS

Tables 1.6–1.8 summarize selected results on the time complexity of some fundamental graph drawing problems.

Table 1.6: The time complexity of some fundamental graph drawing problems: general graphs and digraphs.

Class of Graphs	Problem	Time Complexity		Source
general graph	minimize crossings		NP-hard	[54]
2-layered graph	minimize crossings in layered drawing with preassigned order on one layer		NP-hard	[43]
general graph	compute maximum planar subgraph		NP-hard	[53]
general graph	planarity testing and computing a planar embedding	$O(n)$	$\Omega(n)$	[8, 13, 47, 22, 68, 82]
general graph	compute maximal planar subgraph	$O(n + m)$	$\Omega(n + m)$	[32, 62, 80, 36]
general digraph	upward planarity testing		NP-hard	[60]
embedded digraph	upward planarity testing	$O(n^2)$	$\Omega(n)$	[3]
single-source digraph	upward planarity testing	$O(n)$	$\Omega(n)$	[4, 69]
general graph	draw as the intersection graph of a set of unit diameter disks in the plane		NP-hard	[12]

It is interesting that apparently similar problems exhibit very different time complexities. For example, while planarity testing can be done in linear time, upward planarity testing is NP-hard. Note that, as illustrated in Figure 1.2(c–d), planarity and acyclicity are necessary but not sufficient conditions for upward planarity.

While many efficient algorithms exist for constructing drawings of trees and planar graphs with good universal area bounds, exact area minimization for most types of drawings is NP-hard, even for trees.

OPEN PROBLEMS

- Reduce the time complexity of upward planarity testing for embedded digraphs, or prove a superlinear lower bound.
- Reduce the time complexity of bend minimization for planar orthogonal drawings of embedded graphs, or prove a superlinear lower bound.

1.4 TECHNIQUES FOR DRAWING GRAPHS

In this section we outline some of the most successful techniques that have been devised for drawing general graphs.

PLANARIZATION

The planarization approach is motivated by the availability of many efficient and well-analyzed drawing algorithms for planar graphs (See Table 1.7). If the graph is nonplanar, it is transformed into a planar graph by means of a preliminary planarization step that replaces each crossing with a fictitious vertex. Finding the minimum number of crossings or a maximum planar subgraph are NP-hard

Table 1.7: The time complexity of some fundamental graph drawing problems: planar graphs and digraphs.

Class of Graphs	Problem	Time Complexity		Source
planar graph	compute planar straight-line drawing with prescribed edge lengths		NP-hard	[44]
planar graph	compute planar straight-line drawing with maximum angular resolution		NP-hard	[55, 73]
embedded graph	test the existence of a planar straight-line drawing with prescribed angles between pairs of consecutive edges incident on a vertex		NP-hard	[55]
maximal planar graph	test the existence of a planar straight-line drawing with prescribed angles between pairs of consecutive edges incident on a vertex	$O(n)$	$\Omega(n)$	[35]
planar graph	compute planar straight-line grid drawing with $O(n^2)$ area and $O(1/n^2)$ angular resolution	$O(n)$	$\Omega(n)$	[21, 96]
planar graph	compute planar polyline drawing with $O(n^2)$ area, $O(n)$ bends, and $O(1/d)$ angular resolutions	$O(n)$	$\Omega(n)$	[73]
triconnected planar graph	compute planar straight-line convex grid drawing with $O(n^2)$ area and $O(1/n^2)$ angular resolution	$O(n)$	$\Omega(n)$	[73]
triconnected planar graph	compute planar straight-line strictly convex drawing	$O(n)$	$\Omega(n)$	[14, 112, 113]
reduced planar st -digraph	compute upward planar grid straight-line dominance drawing with minimum area	$O(n)$	$\Omega(n)$	[33]
upward planar digraph	compute upward planar polyline grid drawing with $O(n^2)$ area and $O(n)$ bends	$O(n)$	$\Omega(n)$	[31, 33]
planar degree-4 graph	compute planar orthogonal grid drawing with minimum number of bends		NP-hard	[60]
planar degree-3 graph	compute planar orthogonal grid drawing with minimum number of bends and $O(n^2)$ area	$O(n^5 \log n)$	$\Omega(n)$	[29]
embedded degree-4 graph	compute planar orthogonal grid drawing with minimum number of bends and $O(n^2)$ area	$O(n^2 \log n)$	$\Omega(n)$	[103]
planar degree-4 graph	compute planar orthogonal grid drawing with $O(n^2)$ area and $O(n)$ bends	$O(n)$	$\Omega(n)$	[7, 73, 108]

Table 1.8: The time complexity of some fundamental graph drawing problems: trees. We denote with k a fixed constant such that $k \geq 1$.

Class of Graphs	Problem	Time Complexity		Source
tree	draw as the Euclidean minimum spanning tree of a set of points in the plane		NP-hard	[42]
degree-4 tree	minimize area in planar orthogonal grid drawing		NP-hard	[37, 77, 99, 9]
degree-4 tree	minimize total/maximum edge length in planar orthogonal grid drawing		NP-hard	[5, 9, 61]
rooted tree	minimize area in a planar straight-line upward layered grid drawing that displays symmetries and isomorphisms of subtrees		NP-hard	[102]
rooted tree	minimize area in a planar straight-line upward layered drawing that displays symmetries and isomorphisms of subtrees	$O(n^k)$	$\Omega(n)$	[102]
binary tree	minimize area in hv-drawing	$O(n \sqrt{n \log n})$	$\Omega(n)$	[40]
rooted tree	compute planar straight-line upward layered grid drawing with $O(n^2)$ area	$O(n)$	$\Omega(n)$	[90]
rooted tree	compute planar polyline upward grid drawing with $O(n)$ area	$O(n)$	$\Omega(n)$	[56]

problems. Hence, existing planarization algorithms use heuristics.

The best available heuristic for the maximum planar subgraph problem is described in [71]. This method has a solid theoretical foundation in polyhedral combinatorics, and achieves good results in practice.

A successful drawing algorithm based on the planarization approach and a bend-minimization method [103] is described in [106]. It has been widely used in software visualization systems,

LAYERING

The layering approach for constructing polyline drawings of directed graphs transforms the digraph into a layered digraph and then constructs a layered drawing. A typical algorithm based on the layering approach consists of the following main steps:

1. Assign each vertex to a layer, with the goal of maximizing the number of edges oriented upward.
2. Insert fictitious vertices along the edges crossing layers, such that each edge in the resulting digraph connects vertices in consecutive layers. (The fictitious vertices will be displayed as bends in the final drawing.)
3. Permute the vertices on each layer with the goal of minimizing crossings.
4. Adjust the position of the vertices in each layer with the goal of distributing the vertices uniformly and minimizing the number of bends.

Most of the subproblems involved in the various steps are NP-hard, hence heuristics must be used. The layering approach was pioneered by Sugiyama et al. [101]. The most notable developments of this technique are due to Gansner et al. [52, 51]

PHYSICAL SIMULATION

This approach uses a physical model where the vertices and edges of the graph are viewed as objects subject to various forces. Starting from an initial random configuration, the physical system evolves into a final configuration of minimum energy, which yields the drawing. Rather than solving a system of differential equations, the evolution of the system is usually simulated using numerical methods (e.g., at each step, the forces are computed and corresponding incremental displacements of the vertices are performed). Drawing algorithms based on the physical simulation approach are often able to detect and display symmetries in the graph. However, their running time is typically high.

The physical simulation approach was pioneered in [38, 79]. Recent sophisticated developments include [20, 49, 50, 63, 72, 100].

1.5 RECENT RESEARCH TRENDS

In this section, we overview selected areas of graph drawing that have recently attracted increasing attention.

THREE-DIMENSIONAL DRAWINGS

Recent advances in hardware and software technology for computer graphics open the possibility of displaying three-dimensional (3D) visualizations on a variety of low-cost workstations. Previous research on 3D graph drawing has focused on the development of visualization systems (see, e.g., [91, 94]).

Much work needs to be done on the theoretical foundations of 3D graph drawing. Recent progress has been reported in [16, 19, 64, 84].

PROXIMITY AND VISIBILITY DRAWINGS

Geometric representations of graphs by means of the vertical visibility relation among horizontal segments are well understood (see, e.g., [1, 34, 95, 107, 109, 115]). They have applications to motion planning and VLSI layout. Recent progress on visibility drawings with a fixed number of directions of visibility has been reported in [23, 74, 78]. Other types of visibility drawings are covered in Chapter **.

Proximity drawings include Gabriel, relative neighborhood, Delaunay, sphere of influence, and minimum spanning drawings. Increasing attention has been devoted to the problem of characterizing the classes of graphs that admit various types of proximity drawings. A survey of this area appears in [28].

DECLARATIVE METHODS

Research in graph drawing has traditionally focused on algorithmic methods, where the drawing of the graph is generated according to a prespecified set of aesthetic criteria (such as planarity or area minimization) that are embodied in an algorithm. The algorithmic approach is computationally efficient, however, it does not naturally support constraints, i.e., requirements that the user may want to impose on the drawing of a specific graph (e.g., clustering or aligning a given set of vertices). Previous work has shown that only a rather limited constraint satisfaction capability can be added to existing drawing algorithms (see, e.g., [34, 106]).

Recently, several attempts have been made at developing languages for the specification of constraints and at devising techniques for graph drawing based on the resolution of systems of constraints (see, e.g., [24, 72, 88]). A visual approach to graph drawing, where the layout of a graph is pictorially specified “by example,” is proposed by Cruz and Garg [18]. Recent work by Eades and Lin [83] also attempts at combining algorithmic and declarative methods. Brandenburg presents a comprehensive approach to graph drawing based on graph grammars [10].

Related work includes the physical simulation approach (Section 1.4) and genetic algorithms [75, 76].

DYNAMIC GRAPH DRAWING

Many graphic interfaces provide graph editors that allow the user to interactively modify the drawing of a graph. After an editing session, it would be useful to have an algorithm that “beautifies” the drawing with smooth modifications that preserve the user’s “mental map” of the diagram [39].

This scenario motivates the development of dynamic algorithms that perform incremental changes to the drawing of a graph subject to a sequence of update operations, such as insertion and deletions of vertices and edges. Trade-offs between running time, optimization of the drawing properties, and preservation of the mental map are typical issues to be addressed in dynamic graph drawing.

Most of the existing results on dynamic graph drawing are limited to planar graphs [15, 32, 45, 80, 89].

EXPERIMENTATION

Many graph drawing algorithms have been implemented and used in practical applications. Most papers show sample outputs, and some also provide limited experimental results on small test suites. However, in order to evaluate the practical performance of a graph drawing algorithm in visualization applications, it is essential to perform extensive experimentations with input graphs derived from the application domain.

The performance of four planar straight-line drawing algorithms on 10,000 randomly generated maximal planar graphs is compared by Jones et al. [70].

Himsolt [66] presents a comparative study of twelve graph drawings algorithms based on various approaches. The experiments are conducted on 100 sample graphs with the graph drawing system *GraphEd* [67]. Many examples of drawings constructed by the algorithms are shown, and various objective and subjective evaluations on the aesthetic quality of the drawings produced are given.

Di Battista et al. [26] report on an extensive experimental study comparing three orthogonal drawing algorithms based on the planarization approach. The test data are 11,582 graphs, ranging from 10 to 100 vertices, which have been generated from a core set of 112 graphs used in “real-life” software engineering and database applications. The experiments are conducted with *Diagram Server* [27, 30] (a graph drawing system based on a client-server architecture) and provide a detailed quantitative evaluation of the performance of the three algorithms.

1.6 SOURCES AND RELATED MATERIAL

A comprehensive bibliography on graph drawing algorithms [25] cites more than 300 papers written before 1993. Most papers on graph drawing are cited in `geom.bib`, the computational geometry BibTeX bibliography available on the internet from `ftp://cs.usask.ca/pub/geometry/` (search for keyword “graph drawing”). Surveys on various aspects of graph drawing appear in [28, 41, 57, 65, 92, 93, 97, 104, 105].

The proceedings of the annual Symposium on Graph Drawing are published by Springer-Verlag in the LNCS series [110]. Special issues dedicated to graph drawing will appear between 1995 and 1996 in *Algorithmica* (edited by G. Di Battista and R. Tamassia), *Computational Geometry: Theory and Applications* (edited by G. Di Battista and R. Tamassia), and the *Journal of Visual Languages and Computing* (edited by I. F. Cruz and P. Eades).

RELATED CHAPTERS

Chapter **: Visibility

Chapter **: Geometric Graph Theory

References

- [1] T. Andreae. Some results on visibility graphs. *Discrete Appl. Math.*, 40:5–17, 1992.
- [2] P. Bertolazzi, R. F. Cohen, G. Di Battista, R. Tamassia, and I. G. Tollis. How to draw a series-parallel digraph. *Internat. J. Comput. Geom. Appl.*, 4:385–402, 1994.
- [3] P. Bertolazzi, G. Di Battista, G. Liotta, and C. Mannino. Upward drawings of tri-connected digraphs. *Algorithmica*, to appear.
- [4] P. Bertolazzi, G. Di Battista, C. Mannino, and R. Tamassia. Optimal upward planarity testing of single-source digraphs. In *1st Annual European Symposium on Algorithms (ESA '93)*, volume 726 of *Lecture Notes in Computer Science*, pages 37–48. Springer-Verlag, 1993.
- [5] S. Bhatt and S. Cosmadakis. The complexity of minimizing wire lengths in VLSI layouts. *Inform. Process. Lett.*, 25:263–267, 1987.
- [6] S. N. Bhatt and F. T. Leighton. A framework for solving VLSI graph layout problems. *J. Comput. Syst. Sci.*, 28:300–343, 1984.
- [7] T. Biedl and G. Kant. A better heuristic for orthogonal graph drawings. In *Proc. 2nd Annu. European Sympos. Algorithms (ESA '94)*, volume 855 of *Lecture Notes in Computer Science*, pages 24–35. Springer-Verlag, 1994.
- [8] K. Booth and G. Lueker. Testing for the consecutive ones property interval graphs and graph planarity using PQ-tree algorithms. *J. Comput. Syst. Sci.*, 13:335–379, 1976.
- [9] F. J. Brandenburg. Nice drawings of graphs and trees are computationally hard. Technical Report MIP-8820, Fakultat für Mathematik und Informatik, Univ. Passau, 1988.
- [10] F. J. Brandenburg. Designing graph drawings by layout graph grammars. In R. Tamassia and I. G. Tollis, editors, *Graph Drawing (Proc. GD '94)*, volume 894 of *Lecture Notes in Computer Science*, pages 416–427. Springer-Verlag, 1995.
- [11] R. P. Brent and H. T. Kung. On the area of binary tree layouts. *Inform. Process. Lett.*, 11:521–534, 1980.
- [12] H. Breu and D. G. Kirkpatrick. Unit disk graph recognition is NP-hard. *Comput. Geom. Theory Appl.* (special issue on Geometric Representations of Graphs, edited by G. Di Battista and R. Tamassia). (to appear).

-
- [13] N. Chiba, T. Nishizeki, S. Abe, and T. Ozawa. A linear algorithm for embedding planar graphs using PQ-trees. *J. Comput. Syst. Sci.*, 30(1):54–76, 1985.
- [14] N. Chiba, K. Onoguchi, and T. Nishizeki. Drawing planar graphs nicely. *Acta Inform.*, 22:187–201, 1985.
- [15] R. F. Cohen, G. Di Battista, R. Tamassia, and I. G. Tollis. A framework for dynamic graph drawing. *SIAM J. Comput.*, to appear.
- [16] R. F. Cohen, P. Eades, T. Lin, and F. Ruskey. Three-dimensional graph drawing. In R. Tamassia and I. G. Tollis, editors, *Graph Drawing (Proc. GD '94)*, volume 894 of *Lecture Notes in Computer Science*, pages 1–11. Springer-Verlag, 1995.
- [17] P. Crescenzi, G. Di Battista, and A. Piperno. A note on optimal area algorithms for upward drawings of binary trees. *Comput. Geom. Theory Appl.*, 2:187–200, 1992.
- [18] I. F. Cruz and A. Garg. Drawing graphs by example efficiently: Trees and planar acyclic digraphs. In R. Tamassia and I. G. Tollis, editors, *Graph Drawing (Proc. GD '94)*, volume 894 of *Lecture Notes in Computer Science*, pages 404–415. Springer-Verlag, 1995.
- [19] G. Das and M. T. Goodrich. On the complexity of approximating and illuminating three-dimensional convex polyhedra. In *Proc. 4th Workshop Algorithms Data Struct.*, Lecture Notes in Computer Science. Springer-Verlag, 1995.
- [20] R. Davidson and D. Harel. Drawing graphs nicely using simulated annealing. *Commun. ACM.* (to appear).
- [21] H. de Fraysseix, J. Pach, and R. Pollack. How to draw a planar graph on a grid. *Combinatorica*, 10:41–51, 1990.
- [22] H. de Fraysseix and P. Rosenstiehl. A depth-first-search characterization of planarity. *Annals of Discrete Mathematics*, 13:75–80, 1982.
- [23] A. M. Dean and J. P. Hutchinson. Rectangle-visibility representations of bipartite graphs. In R. Tamassia and I. G. Tollis, editors, *Graph Drawing (Proc. GD '94)*, volume 894 of *Lecture Notes in Computer Science*, pages 159–166. Springer-Verlag, 1995.
- [24] E. Dengler, M. Friedell, and J. Marks. Constraint-driven diagram layout. In *Proc. IEEE Symp. on Visual Languages (VL '93)*, pages 330–335, 1993.
- [25] G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. Algorithms for drawing graphs: an annotated bibliography. *Comput. Geom. Theory Appl.*, 4:235–282, 1994.
- [26] G. Di Battista, A. Garg, G. Liotta, R. Tamassia, E. Tassinari, and F. Vargiu. An experimental comparison of three graph drawing algorithms. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, 1995.
- [27] G. Di Battista, A. Giammarco, G. Santucci, and R. Tamassia. The architecture of Diagram Server. In *Proc. IEEE Workshop on Visual Languages (VL '90)*, pages 60–65, 1990.
- [28] G. Di Battista, W. Lenhart, and G. Liotta. Proximity drawability: a survey. In R. Tamassia and I. G. Tollis, editors, *Graph Drawing (Proc. GD '94)*, volume 894 of *Lecture Notes in Computer Science*, pages 328–339. Springer-Verlag, 1995.
- [29] G. Di Battista, G. Liotta, and F. Vargiu. Spirality of orthogonal representations and optimal drawings of series-parallel graphs and 3-planar graphs. In *Proc. Workshop*

- Algorithms Data Struct.*, volume 709 of *Lecture Notes in Computer Science*, pages 151–162. Springer-Verlag, 1993.
- [30] G. Di Battista, G. Liotta, and F. Vargiu. Diagram Server. *J. Visual Languages and Computing* (special issue on Graph Visualization, edited by I. F. Cruz and P. Eades), 6(3), 1995.
- [31] G. Di Battista and R. Tamassia. Algorithms for plane representations of acyclic digraphs. *Theoret. Comput. Sci.*, 61:175–198, 1988.
- [32] G. Di Battista and R. Tamassia. Incremental planarity testing. In *Proc. 30th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 436–441, 1989.
- [33] G. Di Battista, R. Tamassia, and I. G. Tollis. Area requirement and symmetry display of planar upward drawings. *Discrete Comput. Geom.*, 7:381–401, 1992.
- [34] G. Di Battista, R. Tamassia, and I. G. Tollis. Constrained visibility representations of graphs. *Inform. Process. Lett.*, 41:1–7, 1992.
- [35] G. Di Battista and L. Vismara. Angles of planar triangular graphs. In *Proc. 25th Annu. ACM Sympos. Theory Comput. (STOC 93)*, pages 431–437, 1993.
- [36] H. N. Djidjev. A linear algorithm for the maximal planar subgraph problem. In *Proc. 4th Workshop Algorithms Data Struct.*, Lecture Notes in Computer Science. Springer-Verlag, 1995.
- [37] D. Dolev, F. T. Leighton, and H. Trickey. Planar embedding of planar graphs. In F. P. Preparata, editor, *Advances in Computing Research*, volume 2, pages 147–161. JAI Press, Greenwich, Conn., 1985.
- [38] P. Eades. A heuristic for graph drawing. *Congressus Numerantium*, 42:149–160, 1984.
- [39] P. Eades, W. Lai, K. Misue, and K. Sugiyama. Preserving the mental map of a diagram. In *Proceedings of Compugraphics 91*, pages 24–33, 1991.
- [40] P. Eades, T. Lin, and X. Lin. Minimum size h-v drawings. In *Advanced Visual Interfaces (Proceedings of AVI '92)*, volume 36 of *World Scientific Series in Computer Science*, pages 386–394, 1992.
- [41] P. Eades and X. Lin. How to draw a directed graph. In *Proc. IEEE Workshop on Visual Languages (VL '89)*, pages 13–17, 1989.
- [42] P. Eades and S. Whitesides. The realization problem for minimum spanning trees is NP-hard. *Algorithmica* (special issue on Graph Drawing, edited by G. Di Battista and R. Tamassia). (to appear).
- [43] P. Eades and N. Wormald. Edge crossings in drawings of bipartite graphs. Technical Report 108, Department of Computer Science, University of Queensland. to appear in *Algorithmica*.
- [44] P. Eades and N. Wormald. Fixed edge length graph drawing is NP-hard. *Discrete Appl. Math.*, 28:111–134, 1990.
- [45] D. Eppstein, Z. Galil, G. Italiano, and T. Spencer. Separator based sparsification for dynamic planar graph algorithms. In *Proc. 25th Annu. ACM Sympos. Theory Comput.*, pages 208–217, 1993.
- [46] S. Even and G. Granot. Rectilinear planar drawings with few bends in each edge. Technical Report 797, Computer Science Dept., Technion, 1994.

-
- [47] S. Even and R. E. Tarjan. Computing an st-numbering. *Theoret. Comput. Sci.*, 2:339–344, 1976.
- [48] M. Formann, T. Hagerup, J. Haralambides, M. Kaufmann, F. T. Leighton, A. Simvonis, E. Welzl, and G. Woeginger. Drawing graphs in the plane with high resolution. *SIAM J. Comput.*, 22:1035–1052, 1993.
- [49] A. Frick, A. Ludwig, and H. Mehltau. A fast adaptive layout algorithm for undirected graphs. In R. Tamassia and I. G. Tollis, editors, *Graph Drawing (Proc. GD '94)*, volume 894 of *Lecture Notes in Computer Science*, pages 388–403. Springer-Verlag, 1995.
- [50] T. Fruchterman and E. Reingold. Graph drawing by force-directed placement. *Softw. - Pract. Exp.*, 21(11):1129–1164, 1991.
- [51] E. R. Gansner, E. Koutsofios, S. C. North, and K. P. Vo. A technique for drawing directed graphs. *IEEE Trans. Softw. Eng.*, 19:214–230, 1993.
- [52] E. R. Gansner, S. C. North, and K. P. Vo. DAG – A program that draws directed graphs. *Softw. - Pract. Exp.*, 18(11):1047–1062, 1988.
- [53] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, New York, NY, 1979.
- [54] M. R. Garey and D. S. Johnson. Crossing number is NP-complete. *SIAM J. Algebraic Discrete Methods*, 4(3):312–316, 1983.
- [55] A. Garg. On drawing angle graphs. In R. Tamassia and I. G. Tollis, editors, *Graph Drawing (Proc. GD '94)*, volume 894 of *Lecture Notes in Computer Science*, pages 84–95. Springer-Verlag, 1995.
- [56] A. Garg, M. T. Goodrich, and R. Tamassia. Area-efficient upward tree drawings. In *Proc. 9th Annu. ACM Sympos. Comput. Geom.*, pages 359–368, 1993.
- [57] A. Garg and R. Tamassia. Upward planarity testing. *Order*.
- [58] A. Garg and R. Tamassia. Efficient computation of planar straight-line upward drawings. In *Graph Drawing '93 (Proc. ALCOM Workshop on Graph Drawing)*, Paris, France, 1993.
- [59] A. Garg and R. Tamassia. Planar drawings and angular resolution: Algorithms and bounds. In *Proc. 2nd Annu. European Sympos. Algorithms (ESA '94)*, volume 855 of *Lecture Notes in Computer Science*, pages 12–23. Springer-Verlag, 1994.
- [60] A. Garg and R. Tamassia. On the computational complexity of upward and rectilinear planarity testing. In R. Tamassia and I. G. Tollis, editors, *Graph Drawing (Proc. GD '94)*, volume 894 of *Lecture Notes in Computer Science*, pages 286–297. Springer-Verlag, 1995.
- [61] A. Gregori. Unit length embedding of binary trees on a square grid. *Inform. Process. Lett.*, 31:167–172, 1989.
- [62] J. C. X. Han and R. E. Tarjan. An $O(m \log n)$ -time algorithm for the maximal subgraph problem. *SIAM J. Comput.*, 22:1142–1162, 1993.
- [63] D. Harel and M. Sardas. Randomized graph drawing with heavy-duty preprocessing. *J. Visual Languages and Computing* (special issue on Graph Visualization, edited by I. F. Cruz and P. Eades), 6(3), 1995.

- [64] S. M. Hashemi and I. Rival. Upward drawings to fit surfaces. In *Order, Algorithms, and Applications (Proc. ORDAL '94)*, volume 831 of *Lecture Notes in Computer Science*, pages 53–58. Springer-Verlag, 1994.
- [65] X. He and M.-Y. Kao. Regular edge labelings and drawings of planar graphs. In R. Tamassia and I. G. Tollis, editors, *Graph Drawing (Proc. GD '94)*, volume 894 of *Lecture Notes in Computer Science*, pages 96–103. Springer-Verlag, 1995.
- [66] M. Himsolt. Comparing and evaluating layout algorithms within GraphEd. *J. Visual Languages and Computing* (special issue on Graph Visualization, edited by I. F. Cruz and P. Eades), 6(3), 1995.
- [67] M. Himsolt. GraphEd: a graphical platform for the implementation of graph algorithms. In R. Tamassia and I. G. Tollis, editors, *Graph Drawing (Proc. GD '94)*, volume 894 of *Lecture Notes in Computer Science*, pages 182–193. Springer-Verlag, 1995.
- [68] J. Hopcroft and R. E. Tarjan. Efficient planarity testing. *J. ACM*, 21(4):549–568, 1974.
- [69] M. D. Hutton and A. Lubiw. Upward planar drawing of single source acyclic digraphs. In *Proc. 2nd ACM-SIAM Sympos. Discrete Algorithms*, pages 203–211, 1991.
- [70] S. Jones, P. Eades, A. Moran, N. Ward, G. Delott, and R. Tamassia. A note on planar graph drawing algorithms. Technical Report 216, Department of Computer Science, University of Queensland, 1991.
- [71] M. Juenger and P. Mutzel. Maximum planar subgraphs and nice embeddings: Practical layout tools. *Algorithmica* (special issue on Graph Drawing, edited by G. Di Battista and R. Tamassia). (to appear).
- [72] T. Kamada. *Visualizing Abstract Objects and Relations*. World Scientific Series in Computer Science, 1989.
- [73] G. Kant. Drawing planar graphs using the canonical ordering. *Algorithmica* (special issue on Graph Drawing, edited by G. Di Battista and R. Tamassia). (to appear).
- [74] G. Kant, G. Liotta, R. Tamassia, and I. Tollis. Area requirement of visibility representations of trees. In *Proc. 5th Canad. Conf. Comput. Geom.*, pages 192–197, Waterloo, Canada, 1993.
- [75] C. Kosak and J. Marks. A parallel genetic algorithm for network-diagram layout. In *Proc. 4th Internat. Conf. on Genetic Algorithms (ICGA91)*, 1991.
- [76] C. Kosak, J. Marks, and S. Shieber. Automating the layout of network diagrams with specified visual organization. *IEEE Trans. Syst. Man Cybern.*, 24(3):440–454, 1994.
- [77] M. R. Kramer and J. van Leeuwen. The complexity of wire-routing and finding minimum area layouts for arbitrary VLSI circuits. In F. P. Preparata, editor, *Advances in Computing Research*, volume 2, pages 129–146. JAI Press, Greenwich, Conn., 1985.
- [78] E. Kranakis, D. Krizanc, and J. Urrutia. On the number of directions in visibility representations of graphs. In R. Tamassia and I. G. Tollis, editors, *Graph Drawing (Proc. GD '94)*, volume 894 of *Lecture Notes in Computer Science*, pages 167–176. Springer-Verlag, 1995.
- [79] J. B. Kruskal and J. Seery. Designing network diagrams. In *Proc. First General Conference on Social Graphics*, pages 22–50. U. S. Department of the Census, 1980.

-
- [80] J. A. La Poutré. Alpha-algorithms for incremental planarity testing. In *Proc. 26th Annu. ACM Sympos. Theory Comput. (STOC 94)*, pages 706–715, 1994.
- [81] C. E. Leiserson. Area-efficient graph layouts (for VLSI). In *Proc. 21st Annu. IEEE Sympos. Found. Comput. Sci.*, pages 270–281, 1980.
- [82] A. Lempel, S. Even, and I. Cederbaum. An algorithm for planarity testing of graphs. In *Theory of Graphs: Internat. Symposium (Rome 1966)*, pages 215–232, New York, 1967. Gordon and Breach.
- [83] T. Lin and P. Eades. Integration of declarative and algorithmic approaches for layout creation. In R. Tamassia and I. G. Tollis, editors, *Graph Drawing (Proc. GD '94)*, volume 894 of *Lecture Notes in Computer Science*, pages 376–387. Springer-Verlag, 1995.
- [84] G. Liotta and G. Di Battista. Computing proximity drawings of trees in the 3-dimensional space. In *Proc. 4th Workshop Algorithms Data Struct.*, Lecture Notes in Computer Science. Springer-Verlag, 1995.
- [85] Y. Liu, P. Marchioro, R. Petreschi, and B. Simeone. Theoretical results on at most 1-bend embeddability of graphs. Technical report, Dipartimento di Statistica, Univ. di Roma “La Sapienza”, 1990.
- [86] Y. Liu, A. Morgana, and B. Simeone. General theoretical results on rectilinear embeddability of graphs. *Acta Math. Appl. Sinica*, 7:187–192, 1991.
- [87] S. Malitz and A. Papakostas. On the angular resolution of planar graphs. *SIAM J. Discrete Math.*, 7:172–183, 1994.
- [88] J. Marks. A formal specification for network diagrams that facilitates automated design. *Journal of Visual Languages and Computing*, 2:395–414, 1991.
- [89] K. Miriyala, S. W. Hornick, and R. Tamassia. An incremental approach to aesthetic graph layout. In *Proc. Internat. Workshop on Computer-Aided Software Engineering (CASE '93)*, 1993.
- [90] E. Reingold and J. Tilford. Tidier drawing of trees. *IEEE Trans. Softw. Eng.*, SE-7(2):223–228, 1981.
- [91] S. P. Reiss. An engine for the 3D visualization of program information. *J. Visual Languages and Computing* (special issue on Graph Visualization, edited by I. F. Cruz and P. Eades), 6(3), 1995.
- [92] I. Rival. Graphical data structures for ordered sets. In I. Rival, editor, *Algorithms and Order*, pages 3–31. Kluwer Academic Publishers, 1989.
- [93] I. Rival. Reading, drawing, and order. In I. G. Rosenberg and G. Sabidussi, editors, *Algebras and Orders*, pages 359–404. Kluwer Academic Publishers, 1993.
- [94] G. G. Robertson, J. D. Mackinlay, and S. K. Card. Cone trees: Animated 3d visualizations of hierarchical information. In *Proc. CHI*, pages 189–193, 1991.
- [95] P. Rosenstiehl and R. E. Tarjan. Rectilinear planar layouts and bipolar orientations of planar graphs. *Discrete Comput. Geom.*, 1(4):343–353, 1986.
- [96] W. Schnyder. Embedding planar graphs on the grid. In *Proc. 1st ACM-SIAM Sympos. Discrete Algorithms*, pages 138–148, 1990.
- [97] F. Shahrokhi, L. A. Székely, and I. Vrt'ó. Crossing numbers of graphs, lower bound techniques and algorithms: a survey. In R. Tamassia and I. G. Tollis, editors, *Graph*

- Drawing (Proc. GD '94)*, volume 894 of *Lecture Notes in Computer Science*, pages 131–142. Springer-Verlag, 1995.
- [98] Y. Shiloach. *Arrangements of Planar Graphs on the Planar Lattice*. PhD thesis, Weizmann Institute of Science, 1976.
- [99] J. A. Storer. On minimal node-cost planar embeddings. *Networks*, 14:181–212, 1984.
- [100] K. Sugiyama and K. Misue. Graph drawing by magnetic-spring model. *J. Visual Languages and Computing* (special issue on Graph Visualization, edited by I. F. Cruz and P. Eades), 6(3), 1995.
- [101] K. Sugiyama, S. Tagawa, and M. Toda. Methods for visual understanding of hierarchical systems. *IEEE Trans. Syst. Man Cybern.*, SMC-11(2):109–125, 1981.
- [102] K. J. Supowit and E. M. Reingold. The complexity of drawing trees nicely. *Acta Inform.*, 18:377–392, 1983.
- [103] R. Tamassia. On embedding a graph in the grid with the minimum number of bends. *SIAM J. Comput.*, 16(3):421–444, 1987.
- [104] R. Tamassia. Drawing algorithms for planar st-graphs. *Australasian Journal of Combinatorics*, 2:217–235, 1990.
- [105] R. Tamassia. Planar orthogonal drawings of graphs. In *Proc. IEEE Internat. Sympos. on Circuits and Systems*, 1990.
- [106] R. Tamassia, G. Di Battista, and C. Batini. Automatic graph drawing and readability of diagrams. *IEEE Trans. Syst. Man Cybern.*, SMC-18(1):61–79, 1988.
- [107] R. Tamassia and I. G. Tollis. A unified approach to visibility representations of planar graphs. *Discrete Comput. Geom.*, 1(4):321–341, 1986.
- [108] R. Tamassia and I. G. Tollis. Planar grid embedding in linear time. *IEEE Trans. on Circuits and Systems*, CAS-36(9):1230–1234, 1989.
- [109] R. Tamassia and I. G. Tollis. Representations of graphs on a cylinder. *SIAM J. Discrete Math.*, 4(1):139–149, 1991.
- [110] R. Tamassia and I. G. Tollis, editors. *Graph Drawing (Proc. GD '94)*, volume 894 of *Lecture Notes in Computer Science*. Springer-Verlag, 1995.
- [111] R. Tamassia, I. G. Tollis, and J. S. Vitter. Lower bounds for planar orthogonal drawings of graphs. *Inform. Process. Lett.*, 39:35–40, 1991.
- [112] W. T. Tutte. Convex representations of graphs. *Proceedings London Mathematical Society*, 10:304–320, 1960.
- [113] W. T. Tutte. How to draw a graph. *Proceedings London Mathematical Society*, 3(13):743–768, 1963.
- [114] L. Valiant. Universality considerations in VLSI circuits. *IEEE Trans. Comput.*, C-30(2):135–140, 1981.
- [115] S. K. Wismath. Characterizing bar line-of-sight graphs. In *Proc. 1st Annu. ACM Sympos. Comput. Geom.*, pages 147–152, 1985.