4. Drawing on Physical Analogies

Ulrik Brandes

Graph layout methods described in previous chapters were based on structural characteristics of the graph, or a preprocessed version of the graph. Often, such knowledge is not provided. In this chapter, we take a look at a class of methods applicable to general graphs, without prior knowledge of any structural properties. Their common denominator is that they liken the graph to a system of interacting physical objects, the underlying assumption being that relaxed (energy-minimal) states of suitably defined systems correspond to readable and informative layouts.

Methods based on physical analogies are quite popular, mainly for three reasons. First of all, they are very intuitive, because layout is related to the everyday experience of the surrounding physical world. Secondly, their basic instances are comparatively easy to understand and to program. The threshold to get started is thus very low. And finally, they often yield fairly satisfactory results on medium-sized graphs up to around 50 vertices. In general, these methods consist of two components,

- a *model* consisting of physical objects (representing the elements of the graph) and interactions between these objects, and
- an *algorithm* that (approximately) computes an equilibrium configuration of the system.

The specifications of a model fully represent the intuition behind what is considered a good layout, ideally depending on the specific context of the graph. Its associated algorithm merely serves as an optimization routine for the objective function explicated in the model.

First we introduce the fundamental concept behind physical modeling, that is the basis for all methods presented in this chapter. Then we describe a number of models and associated algorithms in Sections 4.2 and 4.3. An asset of physical modeling that is often overlooked is its inherent flexibility. For this reason, we conclude this chapter by listing examples of model specifications tailored to specific layout objectives.

4.1 The Springs

Given a connected undirected graph with no particular background information, the following two criteria of readable layout seem to be generally agreed upon for the conventional two-dimensional straight-line representation.

1. Vertices should spread well on the page.
2. Adjacent vertices should be close.

Only intuitive explanations can be offered. While uniform vertex distribution reduces clutter, the implied uniform edge lengths leave an undistorted impression of the graph. Since "clutter" and "distortion" already have physical connotations, it seems fairly natural to start thinking of a more specific physical analogy.

We are used to observing even spacing between repelling objects. This makes it natural to imagine vertices behaving like charged balls to satisfy the first criterion. A physical analogy for the second criterion is also easy to find, since it states that we should not allow adjacent balls to drift too far apart. Springs replacing edges will do the job. Springs are better suited than, for example, sticks or ropes, because they can be both extended and compressed to allow moderate distortion, but exert increasingly strong forces when deviating further from their natural length. Moderate distortion may be inevitable, since it is impossible to represent every graph with straight edges of equal length. Note that it is $\mathcal{NP}$-hard to decide whether an arbitrary graph has a straight-line embedding with equal edge lengths in any number of dimensions (Johnson, 1982) or just any planar straight-line embedding (Eades and Wormald, 1990).

Figure 4.1 illustrates the imaginary substitution of vertices and edges with charged balls and connecting springs, respectively. If the system is let go, it attains an equilibrium state in which all forces cancel each other, and the substitution can be reversed to obtain a straight-line drawing that satifies the criteria from above at least approximately.

Formally, such a model can be expressed either in terms of forces acting on the physical objects, or in terms of a potential energy reflecting the internal stress of the system and thus describing how well a configuration matches the design goals modeled in the system. Algorithms to simulate a system's relaxation typically try to move the objects iteratively into stable states in which all forces cancel each other, or to minimize the energy directly. Prominent examples from each kind of formalization are described in the following two sections. In Section 4.4, we give examples for creative use of these analogies in more advanced layout models that include various criteria for good layout.

4.2 Force-Directed Placement

The seminal paper for physical modeling in graph drawing is a short text of Eades (1984), though closely related methods had already been described in the context of VLSI design (Fisk et al., 1967; Quinn and Breuer, 1979). Given a connected undirected graph $G = (V, E)$, let $p = (p_v)_{v \in V}$ be a vector of vertex positions $p_v = (x_v, y_v)$ in the plane. We denote by $\|p_v - p_u\|$ the length of the difference vector $p_v - p_u$, which is the Euclidean distance between positions p_u and p_v. Furthermore, we denote by $\overrightarrow{p_u p_v}$ the unit length vector $\frac{p_v - p_u}{\|p_v - p_u\|}$ pointing from p_u to p_v. The model of Eades (1984), now known as the

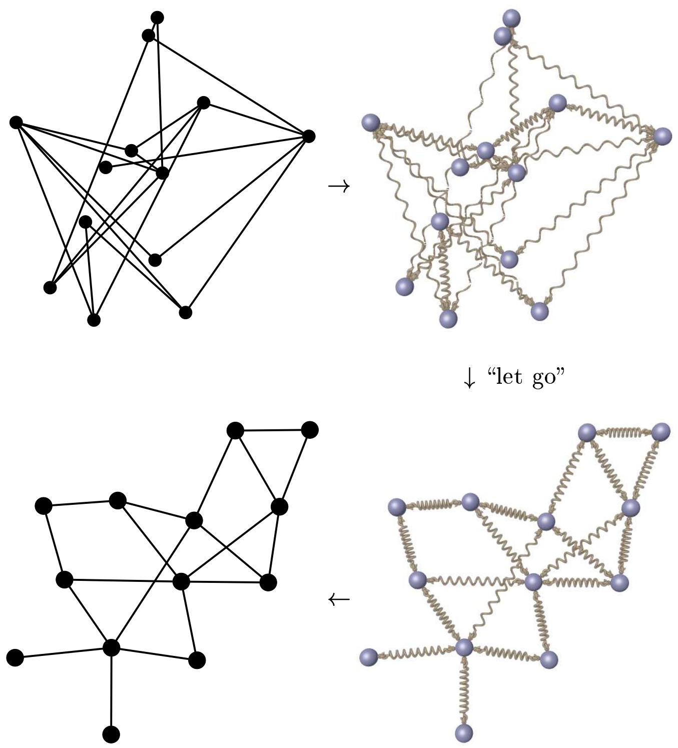


Fig. 4.1. The spring analogy.

spring embedder, implements the analogy described in the previous section. It is defined using repelling forces

$$f_{\mathsf{rep}}(p_u, p_v) = \frac{c_\varrho}{\|p_v - p_u\|^2} \cdot \overrightarrow{p_u p_v}$$

between every pair of non-adjacent vertices $u, v \in V$, where c_ϱ is a repulsion constant. Complementary spring forces between adjacent vertices $u, v \in V$ shall keep these sufficiently apart, yet close to each other. However, instead of more realistic forces according to Hooke's law, (imaginary) logarithmic springs which exert weaker forces on far apart vertices are employed. They yield forces

$$f_{\mathsf{spring}}(p_u, p_v) = c_\sigma \cdot \log \frac{\|p_u - p_v\|}{l} \cdot \overrightarrow{p_v p_u},$$

so that the direction depends on whether the actual distance is less or greater than a natural length l of the spring. Constant c_σ is a parameter controlling the strength of the spring. Figure 4.2 gives a qualitative impression of the forces a vertex u exerts on vertex v, depending on the distance between the two. The solid line shows the force in case u is adjacent to v (f_{spring}), while the dotted line indicates the force in case u is not adjacent to v (f_{rep}). Positive values signify a force dragging v towards u, whereas negative values signify a force repelling v from u.

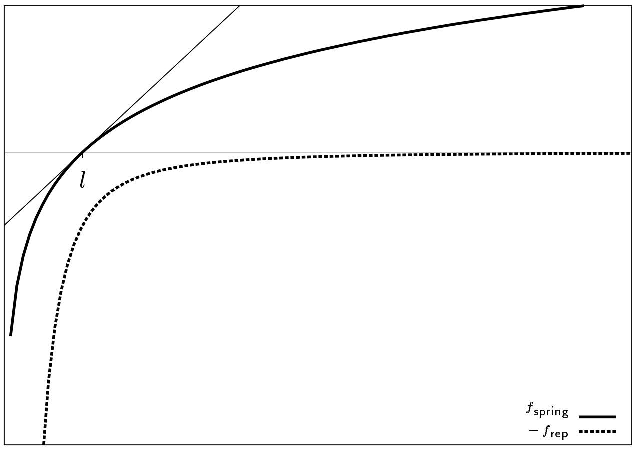


Fig. 4.2. Magnitude of spring embedder forces.

Next we address the question of how to obtain an equilibrium configuration. Vertex positions not corresponding to a system at equilibrium imply positive internal stress. To relax a stressed system, vertices are iteratively moved, at time t, according to a *net force vector* $F_v(t)$, which is the sum of all repulsion and spring forces acting on v. After computing $F_v(t)$ for all $v \in V$, each vertex is moved a constant δ times this vector. This constant is used to prevent excessive movement due to synchronous update. By iteratively computing the forces on all vertices and updating positions accordingly, the system approaches a stable state, in which no local improvement is possible. See Algorithm 6 for a concise description.

Algorithm 6: Spring embedder

Input: connected undirected graph $G = (V, E)$
initial placement $p = (p_v)_{v \in V}$

Output: placement p with low internal stress

for $t \leftarrow 1$ ***to*** ITERATIONS **do**
  for $v \in V$ **do**
    $F_v(t) \leftarrow \sum_{u : \{u,v\} \notin E} f_{\mathsf{rep}}(p_u, p_v) + \sum_{u : \{u,v\} \in E} f_{\mathsf{spring}}(p_u, p_v)$
  for $v \in V$ **do** $p_v \leftarrow p_v + \delta \cdot F_v(t)$

Despite its simplicity, the spring embedder produces satisfactory output in many cases. To even out some shortcomings of the method, several refinements have been developed. These refinements mainly aim at faster computation, but sometimes also at improved quality of the layout.

A number of heuristics is used by Fruchterman and Reingold (1991) to speed up many aspects of layout computation. Firstly, the forces are modified to allow faster evaluation. Repelling forces

$$f_{\mathsf{rep}}(p_u, p_v) = \frac{l^2}{\|p_u - p_v\|} \cdot \overrightarrow{p_u p_v}$$

are used between every pair of vertices, and additional attracting forces

$$f_{\mathsf{attr}}(p_u, p_v) = \frac{\|p_u - p_v\|^2}{l} \cdot \overrightarrow{p_v p_u}$$

are used between adjacent vertices. The combination of attraction and repulsion between adjacent vertices yields a spring-like force $f_{\mathsf{spring}}(p_u, p_v) = f_{\mathsf{attr}}(p_u, p_v) + f_{\mathsf{rep}}(p_u, p_v)$, similar in effect to the force used by Eades (1984). Since its magnitude increases more than proportionally with the distance (see Figure 4.3 for a comparison), one may also hope for faster convergence.

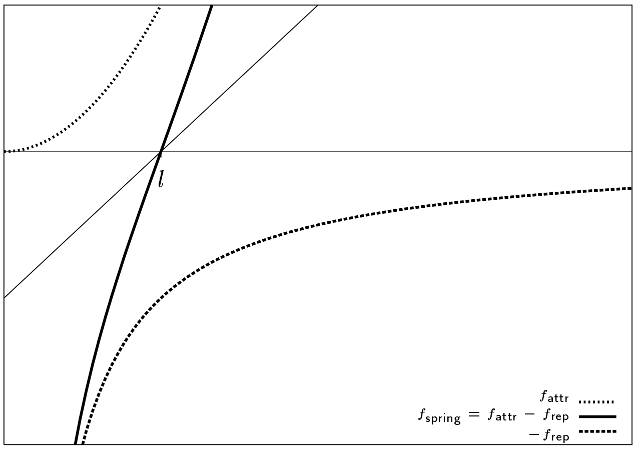
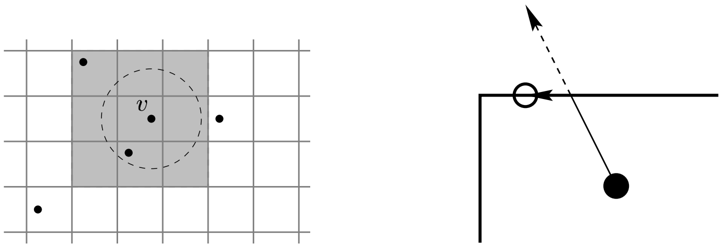


Fig. 4.3. Modified forces by Fruchterman and Reingold (1991).

A second heuristic to speed up computation does not change the objective function, but the precision of evaluation. Since repulsion from far away vertices does not contribute much to the displacement vector, such irrelevant vertices are omitted in the sum of repulsive forces using a grid technique. Only vertices lying in grid cells close to the cell of v are considered, and only if their distance is below a fixed threshold, a repulsive force is calculated and included in the sum of forces. See Figure 4.4(a).

Two other modifications with respect to Algorithm 6 are concerned with the displacement vector. Instead of applying a constant damping factor δ to the net force vector, the net force vector is clipped at a time-dependent maximum displacement $\delta(t)$ to prevent excessive changes, especially in later stages of the iteration when the placement is close to a stable state. The second modification to the displacement ensures that the graph is laid out inside of a given rectangular area, like a screen or a sheet of paper. If the displacement would position a vertex beyond a fixed boundary, the corresponding coordinate of the displacement vector is clipped.



(a) neglecting weak repulsive forces (b) coordinate clipping

Fig. 4.4. Spring embedder modifications of Fruchterman and Reingold (1991).

Another notable refinement of the basic spring embedder is described in Frick et al. (1995). Again, both forces and iteration scheme are modified to speed up the algorithm and to improve layout quality (under the same criteria). Repulsive and attractive forces are defined so that no square root has to be taken,

$$f_{\mathsf{rep}}(p_u, p_v) = \frac{l^2}{||p_u - p_v||^2} \cdot (p_u - p_v),$$
$$f_{\mathsf{attr}}(p_u, p_v) = \frac{||p_u - p_v||^2}{l^2 \cdot \Phi(v)} \cdot (p_v - p_u),$$

and all computations are performed using integer arithmetic. The denominator in the attractive force is defined as $\Phi(v) = 1 + \frac{d_G(v)}{2}$ and effectively slows down high-degree vertices. However, a new gravitational force is introduced, dragging each vertex towards the barycenter $\zeta = \sum_{w \in V} p_w$ of all vertices by

$$f_{\mathsf{grav}}(p_u, p_v) = \Phi(v) \cdot \gamma \cdot \left(\frac{\zeta}{|V|} - p_v \right),$$

which is stronger for high degree vertices. The contribution of this force is controlled via a gravitational constant γ. Similar to layout area restriction, gravitational forces prevent components of a disconnected graph from drifting arbitrarily far apart. In addition to the above forces, a small random force is added to the net force to make the algorithm more robust against poor equilibrium states.

To reduce the number of iterations, the net force vector of each vertex during the previous iteration, $F_v(t-1)$, is stored and compared to the current one. Each vertex has its own adaptive absolute displacement distance $\delta_v(t)$ that is modified according to the angular difference $\alpha = \angle(F_v(t-1), F_v(t))$ of the current and the previous net force vector. If a vertex is to be moved into roughly the same direction as before ($\sin\alpha \approx 1$), $\delta_v(t)$ is chosen larger, if it is to be moved in the opposite direction (*oscillation*: $\sin\alpha \approx -1$), $\delta_v(t)$ is chosen smaller. Like oscillation, *rotation* is an indicator of ineffective movement of a vertex. A *skew gauge* is updated, if the current and the previous displacement vector are almost perpendicular ($\cos\alpha \approx 1$), and $\delta_v(t)$ is lowered, if a large skew suggests that a vertex rotates around some position. During each iteration, vertices are visited in random order, and each position is updated immediately by $p_v \leftarrow p_v + \max\{\delta_{\mathsf{max}}, \delta_v(t)\} \cdot \frac{F_v(t)}{\|F_v(t)\|}$, where δ_{max} is a fixed maximum displacement. Our own experience confirms that these heuristics substantially reduce the number of iterations needed to reach a stable state.

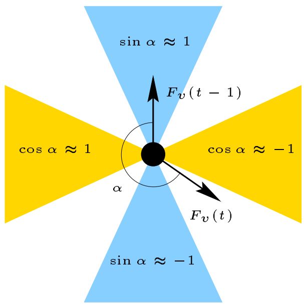


Fig. 4.5. Detection of oscillation and rotation (Frick et al., 1995).

The modifications of Fruchterman and Reingold (1991) and Frick et al. (1995) demonstrate that the spring embedder can be varied in many ways without changing its principle behavior. Clearly, many other heuristics are conceivable, but the two examples presented seem to cover sufficiently many aspects relevant to faster computation, faster convergence, and robust results.

In the next section, we turn to methods that try to satisfy the same criteria, but take a formally different approach by defining an explicit objective function.

4.3 Energy-Based Placement

Forces defined in the spring embedder variants described above indicate in which direction a vertex can be moved to reduce the forces acting on it, and thus an implicit internal energy of the physical system. Instead of displacing vertices according to these forces, one might as well attempt to minimize this energy directly. A spring of natural length l and of strength c_σ with actual length d (assumed to be within reasonable limits) has a *potential energy* of

$$U_{\mathsf{spring}}(d) = c_\sigma \cdot (d - l)^2,$$

Kamada and Kawai (1989) avoid a second potential for repulsion by using springs of different length and strength between every pair of vertices. Their specific choice of springs is governed by the assumption that the ideal distance between two vertices is the length of a shortest path between them, multiplied by the ideal length of a single edge, i.e. every path in the graph is best represented by a straight line. The natural length of the spring connecting vertices $u, v \in V$ is therefore chosen proportional to $d_G(u, v)$, which denotes the length of a shortest path between them. Clearly, perfect relaxation of all springs is impossible for most graphs, so local distances are rendered more important by using springs of strength inverse to their length. The resulting objective function is the sum over the potential energies of all $n \cdot (n-1)/2$ springs,

$$U_{\mathsf{KK}}(p) = \sum_{u,v \in V} \frac{c}{d_G(p_u, p_v)^2} \cdot \left(\|p_u - p_v\| - l \cdot d_G(u,v)\right)^2 ,$$

where c is a scaling constant, and l is the ideal length of a single edge.

To obtain a local minimum of this objective function, a modified Newton-Raphson method is applied. In a local minimum, all partial derivatives of U_{KK} are zero. This condition can be expressed in a system of dependent non-linear equations. Similar to Quinn and Breuer (1979), the Newton-Raphson method is modified in that the coordinates of a single vertex are updated while all others are fixed. In each iteration, the vertex with the longest gradient is picked and moved several times until its gradient falls below a given threshold.

It is interesting to note that the physically inspired objective function U_{KK} is closely related to the objective function

$$U_{\mathsf{MDS}}(p) = \frac{1}{\sum\limits_{u,v \in V} d_G(u,v)^2} \cdot \sum_{u,v \in V} \left(\|p_u - p_v\| - l \cdot d_G(u,v)\right)^2$$

of multidimensional scaling defined in (Kruskal and Wish, 1978). The family

$$S_k(p) = \frac{1}{\sum\limits_{u,v \in V} l_{u,v}^{2-k}} \cdot \sum_{u,v \in V} \frac{1}{l_{u,v}^k} \cdot \left(\|p_u - p_v\| - l_{u,v}\right)^2$$

of objective functions, where $l_{u,v}$ is the desired distance between vertices u and v, and $k \in \{0, 1, 2\}$, is discussed by Cohen (1997). While S_0 corresponds to multidimensional scaling, S_2 corresponds to the above layout objective function. As early as in the 1960s, multidimensional scaling was the first technique for automatic layout of social networks,[1] and it is still in use (Krackhardt et al., 1994). It seems, though, that the bias of U_{KK} towards exact representation of short distances, due to stronger short springs, results in layouts that display less clutter and have fewer small angles.

While the optimization method used by Kamada and Kawai (1989) does not differ notably from the ones in the previous section, the following simplification of the objective function admits exact optimization in time that is polynomial in the number of vertices. Instead of springs of some varying length, Tutte (1963) uses springs of ideal length zero. Setting the partial derivatives of the resulting objective function

$$U_{\mathsf{center}}(p) = \sum_{\{u,v\} \in E} \|p_u - p_v\|^2$$

equal to zero yields two independent systems of linear equations (one for each coordinate), as opposed to the one non-linear system obtained from U_{KK}. These linear systems of equations can conveniently be written in the form

$$(D - A) \cdot x = \mathbf{0},$$
$$(D - A) \cdot y = \mathbf{0},$$

where A is the adjacency matrix of G, D is the diagonal matrix of vertex degrees, and x and y are the vectors of x- and y-coordinates. The famous Matrix Tree Theorem (Kirchhoff, 1847) then states that the determinant of any submatrix obtained from $D - A$ by deleting a positive number of rows and their corresponding columns equals the number of spanning trees of the graph obtained from G by contracting the vertices corresponding to these rows into a new one. See Chaiken and Kleitman (1978) for several variants of this theorem. Since this number clearly is positive for connected graphs, so is the determinant, which implies that there is a unique layout minimizing U_{center}, if only the position of at least one vertex in each connected component of the input graph is fixed. This optimal layout can be computed by solving the smaller system of linear equations obtained by adjusting the right hand sides using the coordinates of fixed vertices.

Optimal layouts with respect to U_{center} are called *barycentric*, because the optimality conditions imply that every vertex not fixed in advance is placed in the barycenter of its neighbors. The main theorem of Tutte (1963) assures even more: barycentric layouts of 3-connected planar graphs are planar with

[1] Charles Kadushin, personal communication (1999).

all internal faces convex, if the vertices of a single face in the unique planar embedding are fixed to lie on a convex polygon (in appropriate order). Since the adjacency matrices of planar graphs are sparse, such layouts can be obtained in time $\mathcal{O}(n \log n)$ (Lipton et al., 1979). See Figure 4.6 for an example.

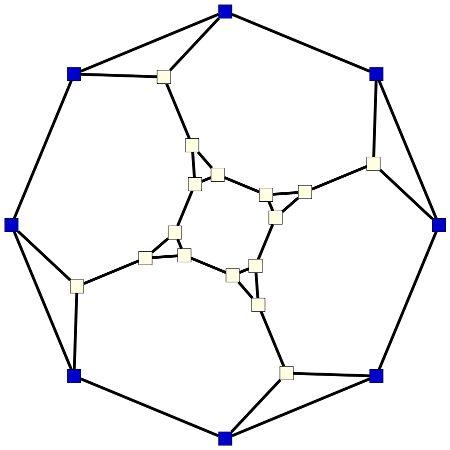


Fig. 4.6. Barycentric layout (darker vertices have been fixed in advance).

Davidson and Harel (1996) deviate even further from models of physical reality. They define attraction and repulsion potentials

$$U_{\mathsf{attr}}(p_u, p_v) = c_{\mathsf{attr}} \cdot \|p_u - p_v\|^2$$
$$U_{\mathsf{rep}}(p_u, p_v) = \frac{c_{\mathsf{rep}}}{\|p_u - p_v\|^2}$$

that are similar to the respective forces of Fruchterman and Reingold (1991). Note that the attraction potential is a scaled version of the zero-length spring potential of Tutte (1963). Combining attraction and repulsion yields a spring potential $U_{\mathsf{spring}}(p_u, p_v) = U_{\mathsf{attr}}(p_u, p_v) + U_{\mathsf{rep}}(p_u, p_v)$ that is again computed for pairs of adjacent vertices. Note that this definition results in an ideal edge length of $\sqrt[4]{c_{\mathsf{rep}}/c_{\mathsf{attr}}}$. Generalizing the repulsion analogy, they also define potentials penalizing vertices that lie close to the boundary of the layout area, and potentials penalizing short distances between a vertex and an edge.

A distinct ingredient of the objective function used in Davidson and Harel (1996) is a potential weighting of the number of crossings in the current layout. Crossings tend to be displeasing and hindering to a viewer (Purchase et al., 1997), and their minization is thus a reasonable criterion for good layout. However, counting crossings leads to a discrete objective function that can no longer be treated by an algorithm based on gradient methods. Moreover, minimizing the number of crossings is also an $\mathcal{NP}$-hard problem (Garey and Johnson, 1983). To obtain at least a local minimum of U_{DH}, *simulated annealing* (see Reeves 1995 for a more recent textbook), a general method for minimizing objective functions of combinatorial problems, is used. It is only by coincidence that this method also has a physical analogy. Given a candidate solution, a new solution is proposed by slightly modifying the current one. If the new solution reduces the value of the objective function, it becomes the new candidate solution. Otherwise, it becomes the new candidate solution only with probability $e^{\frac{-\Delta U}{T}}$, where ΔU is the increase of the objective function, and $T > 0$ is the *temperature* parameter controlling the algorithms ability to climb up hills in the energy landscape. Convergence is enforced by slowly lowering T to zero. A more precise description of a typical simulated annealing algorithm used for graph layout is given in Algorithm 7.

Algorithm 7: Simulated annealing

```
Input: graph G = (V, E)
       initial placement p = (p_v)_{v∈V}
Output: placement p with locally optimal value U(p)

while T > THRESHOLD do
    for v ∈ V do
        p^old ← p
        p_v ← p_v + Δ_random
        if U(p^old) < U(p) then
            with probability 1 − e^((U(p^old) − U(p))/T) reset p ← p^old
    reduce T
```

A number of heuristics to speed up layout computation with the notoriously slow simulated annealing is introduced by Tunkelang (1994). Most notably, there is no initial placement, but vertices are introduced in a breadth-first-search order starting in the graph-theoretic center, and positions are restricted to a coarse grid with only few types of displacements allowed. A parallel implementation is described in Monien et al. (1996).

An experimental comparison (Brandenburg et al., 1996) of (sequential) implementations revealed that the approaches presented in these last two sections (excluding the barycentric model) yield comparable layouts. As a very general rule-of-thumb, energy-based placement approaches tend to produce better results for small to medium-sized graphs (around 30 vertices), while force-directed placement approaches are considerably faster (experience suggest that they need only one tenth of the number of iterations).

Besides gradient methods and simulated annealing, genetic algorithms have been applied to physically motivated objective functions (Kosak et al., 1994; Masui, 1992). Branke et al. (1997) use force-directed methods as a local fine-tuning step of their genetic algorithm.

4.4 Modeling with Forces and Energies

Forces and potential energies have been found to model elementary criteria for readable layouts of straight-line representations of graphs. While the results are usually satisfactory with respect to the two criteria mentioned in the introductory section of this chapter (uniform vertex distribution, uniform edge lengths, and, as a consequence, symmetry), such drawings may not be useful for graphs representing specific structural information.

An important aspect that has not been covered yet is the immense expressive power of force-directed or energy-based placement for formulating layout design goals. This section contains a number of examples, showing how forces can be used to formulate criteria for good layout far beyond vertex distribution and edge length.

The following paragraphs sketch modeling ideas from the literature, comprising a toolbox for the adjustment of force-directed or energy-based layout methods to account for a fairly broad range of requirements.

3D Layouts. For some purposes it may be necessary or desirable to represent a graph in three-dimensional space. In combination with interactive (preferably fly-through) browsers, these appear to be particularly useful for exploring large graphs. The methods outlined so far make no particular assumptions on the number of dimensions and are easily modified to produce three-dimensional (or one-dimensional, for that matter) layouts. For example, a straightforward generalization of the method by Frick et al. (1995) is given in Bruß and Frick (1996). The approach of Davidson and Harel (1996) is adjusted by Cruz and Twarog (1996), where the crossing count, superfluous in three dimensions, is substituted with edge-edge repulsion. See Chapter 7 for more on three-dimensional graph layout.

Clustering. Important information may be represented by intrinsic or extrinsic clusterings of vertices. Since experimental work suggests that users tend to group geometrically close vertices (McGrath et al., 1996), it seems desirable to keep vertices of the same cluster close to each other, while those in different clusters should be further apart.

By their very nature, spring-type methods geometrically cluster dense subgraphs. While this is not at all true for other definitions of clusters, these can be reduced to dense subgraphs by introducing dummy vertices that represent the clusters, and connecting them appropriately (Eades et al., 1997a; Huang and Eades, 1998a). For each cluster, a new vertex is introduced, attracting its contained vertices, while repelling vertices of other clusters. To achieve larger distances between clusters, additional repelling forces are introduced between dummy vertices and all other vertices not contained in their respective cluster. Figure 4.8(a) gives a small example.

Instead of introducing dummy vertices to represent cluster centroids, a *meta-graph* representing connections between clusters is introduced by Wang and Miyamoto (1996). It consists of a vertex for each cluster, and has an

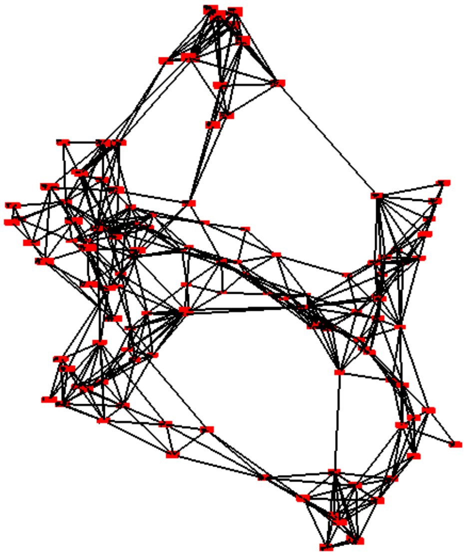


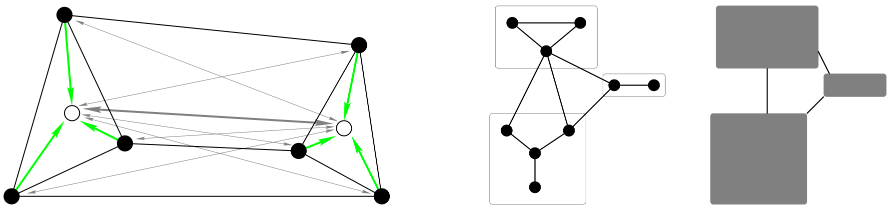
Fig. 4.7. 3D layout of travel distances between 31 U.S. cities (Bruß and Frick, 1996).

edge between each pair of distinct meta-vertices containing adjacent vertices of the original graph. To provide the necessary area for clusters, meta-vertices are not assumed to have point size, but to occupy rectangular areas. Consequently, first a technique is introduced that accounts for the fact that actual edge lengths depend on the relative positions of rectangularly shaped vertices. In the placement iteration, contribution weights are used to gradually shift from an emphasis of forces in the meta-graph (for cluster spacing) to forces in the input graph. More detail on force-directed methods to represent clusters is provided in Chapter 8.

Directed edges. For directed graphs such as data-flow or time-dependency graphs, it is often desirable to have the directed edges point into roughly the same direction. Using forces exerted on edges by external fields of a chosen orientation, edges can be pushed to point in any given direction (Sugiyama and Misue, 1995). Let θ be the angle between an edge's prescribed direction and its current direction, and let $\overrightarrow{p_u p_v}^{\perp}$ be the unit length vector perpendicular to $\overrightarrow{p_u p_v}$ and pointing towards a decrease of θ, then rotative forces

$$f_{\mathsf{rot}}(p_u, p_v) = b \cdot ||p_u - p_v||^{c_1} \cdot \theta^{c_2} \cdot \overrightarrow{p_u p_v}^{\perp}$$

can be added to the net force vector, thus pressing the edge to reduce θ (see Figure 4.9). While b is a constant that controls the strength of the magnetic field acting on an edge between u and v with respect to the other forces, c_1 and c_2 are parameters that control the relative dependency of rotative forces on vertex distance and angle deviation, respectively.



(a) Dummy vertices represent the centroids of clusters (Eades et al., 1997a; Huang and Eades, 1998a).

(b) A clustering with its corresponding meta-graph (Wang and Miyamoto, 1996).

Fig. 4.8. Force-directed models for clusters.

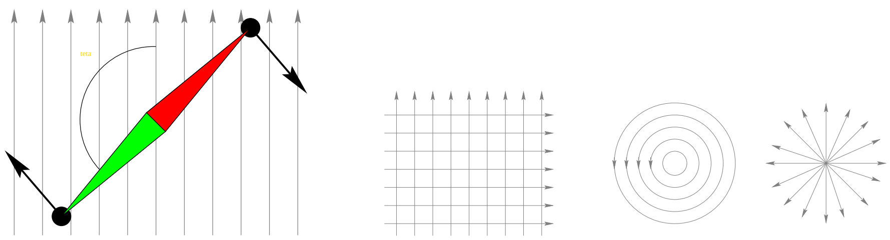


Fig. 4.9. Magnetic springs represent directed edges subject to an arbitrary force field (Sugiyama and Misue, 1995).

Curved edges. So far, all representations considered here use straight lines to depict edges. It is shown in Brandes and Wagner (1998b) how the layout of curved line representations can be reduced to the straight-line case by placing control points instead of vertices. More precisely, edges that are to be depicted by Bézier curves are replaced with vertices representing their control points. Then an auxiliary graph is constructed by introducing a dummy edge of appropriate desired length for each control segment, and additional attractive and repulsive forces to ensure certain properties of the resulting layout. Figure 4.10 shows a graph of train connections with curved representation of edges that are classified as "transitive" (top left), the auxiliary graph constructed for this instance (bottom left), and a larger example (right). The same technique obviously works for polyline representations.

Dynamic layout. When users look at the drawing of a graph, they familiarize themselves with this drawing by building a cognitive representation called the *mental map*. Dynamic graph layout is concerned with the trade-off between static layout quality and the user's mental map.

A framework to extend static graph layout methods to dynamic ones by incorporating a difference metric into the objective function is advocated

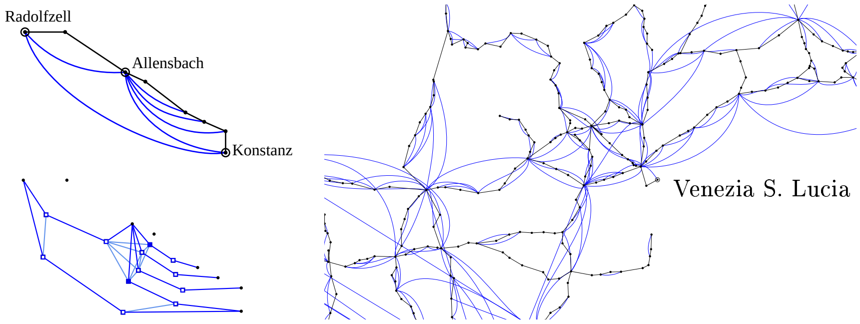


Fig. 4.10. Curved edge layout using an auxiliary graph (Brandes and Wagner, 1998b).

in Brandes and Wagner (1997). For energy-based methods, the difference metric summing over the Euclidean distances of vertex positions translates nicely into an attraction potential for imaginary springs of natural length zero, which can be used in force-directed approaches as well. An overview of approaches for drawing dynamic and interactive graphs is provided in Chapter 9.

Constraints. Constraints can be integrated into force-directed or energy-based placement methods in several ways, depending on the type of constraint. Unary constraints, i.e., restrictions on the admissible positions of single vertices, are easily satisfied if feasible positions form a connected region in the layout area or space (see Figure 4.11). In this case, every configuration remains reachable, and position updates can be modified to ensure that each vertex is at a feasible position at all times.

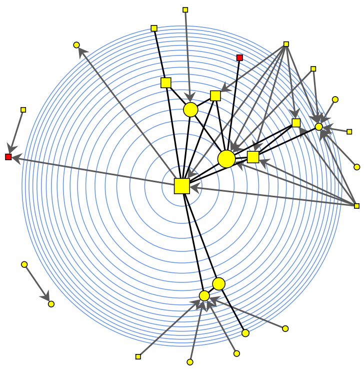


Fig. 4.11. Unary constraints. In this drawing of a social network, each vertex is restricted to lie on the circumference of a circle, signifying its centrality (Brandes, 1999).

A second way to satisfy unary constraints that also handles disconnected feasible regions is to allow vertices to temporarily occupy infeasible positions, but to use additional forces tht drag vertices to feasible positions or penalties pushing them away from infeasible positions, respectively. The penalty approach is easily extended to arbitrary constraints, but rarely guarantees satisfaction of all constraints in practice.

To satisfy pairwise linear (in)equalities, the placement iteration is interleaved with conflict detection in Wang and Miyamoto (1996). Every pair of vertices that would not satisfy a constraint if a displacement is carried out, is joined by a "rigid stick" that forces them to move together while maintaining the constraint. Note that criteria corresponding to weak linear pairwise constraints can be modeled using the magnetic springs of Sugiyama and Misue (1995).

Arbitrary constraints can be incorporated in methods based on physical analogies by applying more elaborate constraint satisfaction techniques, as in He and Marriott (1998). A survey of the use of constraints in graph layout is given by Tamassia (1998)

From a more general point of view, the above modeling techniques can be seen as instances of a generic framework for layout models, in which arbitrary layout variables are assigned values subject to some constraints and criteria that describe feasible and desirable configurations. Graph layout is thus cast as a general constraint optimization problem, for which objective functions can be devised by combining criteria for small configurations like pairs of adjacent or non-adjacent vertices, triples of connected vertices, pairs of edges, and so on. Force-directed and energy-based methods fit well into this framework, because they are intuitive and incorporate an implicit notion of compromise between the various criteria.

Bibliography

Abbott, K. R., and Sarin, S. K. (1994). Experiences with workflow management: Issues for the next generation. In *Proceedings of ACM Conference on Computer-Supported Cooperative Work, Workflow and Information Sharing (CSCW'94)*, pages 113–120.

Agarwal, P. K., and Erickson, J. (1997). Geometric range searching and its relatives. Technical Report CS 1997-11, Department of Computer Science, Duke.

Agarwal, P. K., van Kreveld, M., and Suri, S. (1998). Label placement by maximum independent set in rectangles. *Computational Geometry: Theory and Applications*, 11(3-4):209–218.

Aho, A., Hopcroft, J., and Ullman, J. (1974). *The Design and Analysis of Computer Algorithms*. Addison-Wesley.

Ahuja, R. K., Magnanti, T. L., and Orlin, J. B. (1993). *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall.

Alpert, C. J., and Kahng, A. B. (1995). Recent directions in netlist partitioning: a survey. *INTEGRATION, the VLSI Journal*, 19:1–81.

Amtrup, H. H. J., and Jost, U. (1996). What's in a word graph? Evaluation and enhancement of word lattices. Technical Report Verbmobil-Report 186, University Hamburg, Germany.

Andreev, E. M. (1970a). On convex polyhedra in Lobacevskii spaces. *Math. USSR-Sb.*, 10:413–440.

Andreev, E. M. (1970b). On convex polyhedra of finite volume in Lobacevskii space. *Math. USSR-Sb.*, 12:255–259.

Auslander, L., and Parter, S. V. (1961). On imbedding graphs in the plane. *Journal of Mathematics and Mechanics*, 10(3):517–523.

Baker, K. A., Fishburn, P. C., and Roberts, F. S. (1971). Partial orders of dimension 2. *Networks*, 2:11–28.

Batagelj, V., Kerzic, D., and Pisanski, T. (1992). Automatic clustering of languages. *Computational Linguistics*, 18(3):339–352.

Berge, C. (1993). *Graphs*. North Holland, Amsterdam, 3rd edition.

Berger, B., and Shor, P. (1990). Approximation algorithms for the maximum acyclic subgraph problem. In *Proceedings of the 1st ACM-SIAM Symposium on Discrete Algorithms (SODA'90)*, pages 236–243.

Bertolazzi, P., Cohen, R. F., Di Battista, G., Tamassia, R., and Tollis, I. G. (1994a). How to draw a series-parallel digraph. *International Journal of Computational Geometry and Applications*, 4:385–402.

Bertolazzi, P., Di Battista, G., and Didimo, W. (1997). Computing orthogonal drawings with the minimum number of bends. In *Proceedings of the 5th Workshop on Algorithms and Data Structures (WADS'97)*, Spinger LNCS 1272, pages 331–344.

Bertolazzi, P., Di Battista, G., Liotta, G., and Mannino, C. (1994b). Upward drawings of triconnected digraphs. *Algorithmica*, 6(12):476–497.

Bertolazzi, P., Di Battista, G., Mannino, C., and Tamassia, R. (1993). Optimal upward planarity testing of single-source digraphs. In *Proceedings of the 1st European Symposium on Algorithms (ESA'93)*, Springer LNCS 726, pages 37–48.

Bertsekas, D. P. (1998). *Network Optimization: Continuous and Discrete Models.* Athena Scientific.

Biedl, T., Shermer, T., Whitesides, S., and Wismath, S. (1999). Bounds for orthogonal 3D graph drawing. *Journal of Graph Algorithms and Applications*, 3(4):63–79.

Biedl, T. C. (1997). *Orthogonal Graph Visualization: The Three-Phase Method with Applications.* PhD thesis, Rutgers University.

Biedl, T. C. (1998). Three approaches to 3D-orthogonal box-drawings. In *Proceedings of the 6th International Symposium on Graph Drawing (GD'98).* Springer LNCS 1547, pages 30–43.

Biedl, T. C., and Kant, G. (1994). A better heuristic for orthogonal graph drawing. In *Proceedings of the 2nd European Symposium on Algorithms (ESA'94)*, Springer LNCS 855, pages 24–35.

Biedl, T. C., and Kaufmann, M. (1997). Area-efficient static and incremental graph drawings. In *Proceedings of the 5th European Symposium on Algorithms (ESA'97)*, Springer LNCS 1284, pages 37–52.

Biedl, T. C., Madden, B. P., and Tollis, I. G. (1997a). The three-phase method: A unified approach to orthogonal graph drawing. In *Proceedings of the 5th International Symposium on Graph Drawing (GD'97).* Springer LNCS 1353, pages 391–402.

Biedl, T. C., Shermer, T., Whitesides, S., and Wismath, S. (1997b). Orthogonal 3D graph drawing. In *Proceedings of the 5th International Symposium on Graph Drawing (GD'97).* Springer LNCS 1353, pages 76–86.

Blythe, J., McGrath, C., and Krackhardt, D. (1996). The effect of graph layout on inference from social network data. In *Proceedings of the 3rd International Symposiom on Graph Drawing (GD'95).* Springer LNCS 1027, pages 40–51.

Böhringer, K.-F., and Paulisch, F. N. (1990). Using constraints to achieve stability in automatic graph layout algorithms. In *Proceedings of the ACM Human Factors in Computing Systems Conference (CHI'90)*, pages 43–51.

Booth, K. S., and Lueker, G. S. (1976). Testing for the consecutive ones property, interval graphs and graph planarity using PQ-tree algorithms. *Journal of Computer and System Sciences*, 13:335–379.

Borgida, A., Brachman, R., McGuinness, D., and Resnick, L. (1989). CLASSIC: A structural data model for objects. In *Proceedings of the 1989 ACM-SIGMOD International Conference on Management of Data*, pages 59–67.

Bose, P., Gomez, F., Ramos, P., and Toussaint, G. (1996). Drawings nice projections of objects in space. In *Proceedings of the 3rd International Symposium on Graph Drawing (GD'95)*, Springer LNCS 1027, pages 52–63.

Brandenburg, F. J., Himsolt, M., and Rohrer, C. (1996). An experimental comparison of force-directed and randomized graph drawing algorithms. In *Proceedings of the 3rd International Symposium on Graph Drawing (GD'95)*, Springer LNCS 1027, pages 76–87.

Brandes, U. (1999). *Layout of Graph Visualizations*. PhD thesis, University of Konstanz. `http://www.ub.uni-konstanz/kops/volltexte/1999/255/`.

Brandes, U., Kenis, P., Raab, J., Schneider, V., and Wagner, D. (1999). Explorations into the visualization of policy networks. *Journal of Theoretical Politics*, 11(1):75–106.

Brandes, U., and Wagner, D. (1997). A Bayesian paradigm for dynamic graph layout. In *Proceedings of the 5th International Symposium on Graph Drawing (GD'97)*, Springer LNCS 1353, pages 236–247.

Brandes, U., and Wagner, D. (1998a). Dynamic grid embedding with few bends and changes. In *Proceedings of the 9th Annual International Symposium on Algorithms and Computation (ISAAC'98)*, Springer LNCS 1533, pages 89–98.

Brandes, U., and Wagner, D. (1998b). Using graph layout to visualize train interconnection data. In *Proceedings of the 6th International Symposium on Graph Drawing (GD'98)*, Springer LNCS 1547, pages 44–56.

Branke, J., Bucher, F., and Schmeck, H. (1997). A genetic algorithm for drawing undirected graphs. In *Proceedings of the 3rd Nordic Workshop on Genetic Algorithms and their Applications*, pages 193–206.

Bridgeman, S., Di Battista, G., Didimo, W., Liotta, G., Tamassia, R., and Vismara, L. (2000). Turn-regularity and optimal area drawings for orthogonal representations. *Computational Geometry: Theory and Applications*, 16(1):53–93.

Bridgeman, S., Fanto, J., Garg, A., Tamassia, R., and Vismara, L. (1997). INTERACTIVEGIOTTO: An algorithm for interactive orthogonal graph drawing. In *Proceedings of the 5th International Symposium on Graph Drawing (GD'97)*, Springer LNCS 1353, pages 303–308.

Bridgeman, S., and Tamassia, R. (1998). Difference metrics for interactive orthogonal graph drawing algorithms. In *Proceedings of the 6th International Symposium on Graph Drawing (GD'98)*, Springer LNCS 1457, pages 57–71.

Bruß, I., and Frick, A. (1996). Fast interactive 3-D graph visualization. In *Proceedings of the 3rd International Symposium on Graph Drawing (GD'95)*, Springer LNCS 1027, pages 99–110.

Cai, J., Han, X., and Tarjan, R. E. (1993). An $O(m \log n)$-time algorithm for the maximal planar subgraph problem. *SIAM Journal on Computing*, 22:1142–1162.

Carpano, M. J. (1980b). Automatic display of hierarchized graphs for computer aided decision analysis. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-10(11):705–715.

Catarci, T. (1995). The assignment heuristic for crossing reduction. *IEEE Trans. Syst. Man Cybern.*, 25(3):515–521.

Chaiken, S., and Kleitman, D. J. (1978). Matrix tree theorems. *Journal of Combinatorial Theory, Series A*, 24:377–381.

Chan, T., Goodrich, M. T., Kosaraju, S. R., and Tamassia, R. (1996). Optimizing area and aspect ration in straight-line orthogonal tree drawings. In *Proceedings of the 4th International Symposium on Graph Drawing (GD'96)*. Springer LNCS 1190, pages 63–75.

Chan, T. M. (1999). A near-linear area bound for drawing binary trees. In *Proceedings of the 10th ACM-SIAM Symposium on Discrete Algorithms (SODA'99)*, pages 161–168.

Chiba, N., Nishizeki, T., Abe, S., and Ozawa, T. (1985). A linear time algorithm for embedding planar graphs using PQ-trees. *Journal of Computer and System Sciences*, 30:54–76.

Christensen, J., Friedman, S., Marks, J., and Shieber, S. (1997). Empirical testing of algorithms for variable-sized label placement. In *Proceedings of the 13th Annual ACM Symposium on Computational Geometry*, pages 415–417.

Christensen, J., Marks, J., and Shieber, S. (1993). Algorithms for cartographic label placement. In *Proceedings of the American Congress on Surveying and Mapping 1*, pages 75–89.

Christensen, J., Marks, J., and Shieber, S. (1995). An empirical study of algorithms for point-feature label placement. *ACM Transactions on Graphics*, 14(3):203–232.

Chrobak, M., and Kant, G. (1997). Convex grid drawings of 3-connected planar graphs. *International Journal of Computational Geometry and Applications*, 7(3):211–224.

Chvátal, V. (1983a). *Linear Programming*. W. H. Freeman.

Closson, M., Everett, H., Gartshore, S., and Wismath, S. (1998). Arrangepak, orthopak and vispak 2.0. Technical Report TR-CS-98, University of Lethbridge.

Coffman, E. G., and Graham, R. L. (1972). Optimal scheduling for two processor systems. *Acta Informatica*, 1:200–213.

Cohen, J. D. (1997). Drawing graphs to convey proximity: An incremental arrangement method. *ACM Transactions on Computer-Human Interaction*, 4(3):197–229.

Cohen, R. F., Di Battista, G., Tamassia, R., and Tollis, I. G. (1995). Dynamic graph drawings: Trees, series-parallel digraphs, and planar *st*-digraphs. *SIAM Journal on Computing*, 24(5):970–1001.

Cohen, R. F., Di Battista, G., Tamassia, R., Tollis, I. G., and Bertolazzi, P. (1992). A framework for dynamic graph drawing. In *Proceedings of the 8th ACM Annual Symposium on Computational Geometry (SCG'92)*, pages 261–270.

Colin de Verdière, Y. (1989). Empilements de cercles: convergence d'une methode de point fixe. *Forum Mathematicum*, 1:395–402.

Cormen, T., Leiserson, C., and Rivest, R. (1990). *Introduction to Algorithms.* The MIT Electrical Engineering and Computer Science Series. The MIT Press and McGraw-Hill Book Company.

Crescenzi, P., Di Battista, G., and Piperno, A. (1992). A note on optimal area algorithms for upward drawings of binary trees. *Computational Geometry: Theory and Applications*, 2:187–200.

Crescenzi, P., and Piperno, A. (1995). Optimal-area upward drawings of AVL-trees. In *Proceedings of the DIMACS International Workshop on Graph Drawing (GD'94)*. Springer LNCS 894, pages 307–317.

Cruz, I. F., and Twarog, J. P. (1996). 3D graph drawing with simulated annealing. In *Proceedings of the 3rd International Symposium on Graph Drawing (GD'95)*, Springer LNCS 1027, pages 162–165.

Cunningham, W. H. (1976). A network simplex method. *Mathematical Programming*, 11:105–116.

Czyzowicz, J. (1991). Lattice diagrams with few slopes. *Journal of Combinatorial Theory, Series A*, 56:96–108.

Czyzowicz, J., Pelc, A., and Rival, I. (1990). Drawing orders with few slopes. *Discrete Mathematics*, 82:233–250.

Dai, W. W.-M., and Kuh, E. S. (1987). Global spacing of building-block layout. In *Proceedings of the IFIP International Conference on Very Large Scale Integration (VLSI'87)*, pages 193–205.

Datta, A., Lenhof, H.-P., Schwarz, C., and Smid, M. H. M. (1993). Static and dynamic algorithms for *k*-point clustering problems. In *Proceedings of the 3rd Workshop on Algorithms and Data Structures (WADS'93)*, Springer LNCS 709, pages 265–276.

Davidson, R., and Harel, D. (1996). Drawing graphs nicely using simulated annealing. *ACM Transactions on Graphics*, 15(4):301–331.

de Fraysseix, H., Pach, J., and Pollack, R. (1990). How to draw a planar graph on a grid. *Combinatorica*, 10:41–51.

Di Battista, G., Eades, P., Tamassia, R., and Tollis, I. G. (1994). Algorithms for drawing graphs: An annotated bibliography. *Computational Geometry*, 4:235–282.

Di Battista, G., Eades, P., Tamassia, R., and Tollis, I. G. (1999). *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall.

Di Battista, G., Liotta, G., and Vargiu, F. (1998a). Spirality and optimal orthogonal drawings. *SIAM Journal on Computing*, 27(6):1764–1811.

Di Battista, G., Liu, W. P., and Rival, I. (1990). Bipartite graphs, upward drawings, and planarity. *Information Processing Letters*, 36(6):317–322.

Di Battista, G., Patrignani, M., and Vargiu, F. (1998b). A split & push approach to 3-D orthogonal drawing. In *Proceedings of the 6th International Symposium on Graph Drawing (GD'98)*. Springer LNCS 1547, pages 87–101.

Di Battista, G., and Tamassia, R. (1988). Algorithms for plane representations of acyclic digraphs. *Theoretical Computer Science*, 61(2-3):175–198.

Di Battista, G., and Tamassia, R. (1989). Incremental planarity testing. In *Proceedings of the 30th Symposium on the Foundations of Computer Science (FOCS'89)*, pages 436–441.

Di Battista, G., and Tamassia, R. (1990). On-line graph algorithms with SPQR-trees. In *Proceedings of the 17th International Colloqium on Automata, Languages and Programming (ICALP'90)*, Springer LNCS 443, pages 598–611.

Di Battista, G., and Tamassia, R. (1996). On-line planarity testing. *SIAM Journal on Computing*, 25(5):956–997.

Di Battista, G., and Vismara, L. (1993). Angles of planar triangulated graphs. In *Proceedings of the 25th Annual ACM Symposium on the Theory of Computing (STOC'93)*, pages 431–437.

Di Battista, G., and Vismara, L. (1996). Angles of planar triangulated graphs. *SIAM Journal on Discrete Mathematics*, 9(3):349–359.

Didimo, W., and Liotta, G. (1998). Computing orthogonal drawings in a variable embedding setting. In *Proceedings of the 9th Annual International Symposium on Algorithms and Computation (ISAAC'98)*, Springer LNCS 1533, pages 79–88.

Dietz, P. F., and Sleator, D. D. (1987). Two algorithms for maintaining order in a list. In *Proceedings of the 19th Annual ACM Symposium of Theory of Computing (STOC'87)*, pages 365–372.

Djidjev, H. N. (1995). A linear algorithm for the maximal planar subgraph problem. In *Proceedings of the 4th Workshop on Algorithms and Data Structures (WADS'95)*. Springer LNCS 955, pages 369–380.

Doddi, S., Mararthe, M. V., Mirzaian, A., Moret, B. M. E., and Zhu, B. (1999). Map labeling and its generalizations. Technical Report LA-UR-96-2411, Los Alamos National Labatory.

Doddi, S., Marathe, M. V., Mirzaian, A., Moret, B. M. E., and Zhu, B. (1997). Map labeling and its generalizations. In *Proceedings of the 8th ACM-SIAM Symposium on Discrete Algorithms (SODA'97)*, pages 148–157.

Dresbach, S. (1995). A new heuristic layout algorithm for directed acyclic graphs. In *Operations Research Proceedings 1994*, pages 121–126.

Duncan, C. A., Goodrich, M. T., and Kobourov, S. G. (1998). Balanced aspect ratio trees and their use for drawing very large graphs. In *Proceedings of the 6th International Symposium on Graph Drawing (GD'98).* Springer LNCS 1547, pages 111–124.

Eades, P. (1984). A heuristic for graph drawing. *Congressus Numerantium*, 42:149–160.

Eades, P., Cohen, R. F., and Huang, M. L. (1997a). Online animated graph drawing for Web navigation. In *Proceedings of the 5th International Symposium on Graph Drawing (GD'97)*, Springer LNCS 1353, pages 330–335.

Eades, P., and Feng, Q. W. (1996). Multilevel visualization of clustered graphs. In *Proceedings of the 4th International Symposium on Graph Drawing (GD'96).* Springer LNCS 1190, pages 101–112.

Eades, P., and Feng, Q. W. (1997). Drawing clustered graphs on an orthogonal grid. In *Proceedings of the 5th International Symposium on Graph Drawing (GD'97).* Springer LNCS 1353, pages 146–157.

Eades, P., Feng, Q. W., and Lin, X. (1996a). Straight-line drawing algorithms for hierarchical graphs and clustered graphs. In *Proceedings of the 4th International Symposium on Graph Drawing (GD'96).* Springer LNCS 1190, pages 113–128.

Eades, P., Feng, Q., and Nagamochi, H. (1999). Drawing clustered graphs on an orthogonal grid. *Journal on Graph Algorithms and Applications*, 3(4):3–29.

Eades, P., Huang, M. L., and Wang, J. (1997b). Online animated graph drawing using a modified spring algorithm. Technical Report 97–05, Department of Computer Science and Software Engineering, University of Newcastle.

Eades, P., and Kelly, D. (1986). Heuristics for reducing crossings in 2-layered networks. *Ars Combinatorica*, 21.A:89–98.

Eades, P., Lai, W., Misue, K., and Sugiyama, K. (1991). Preserving the mental map of a diagram. In *Proceedings of Compugraphics '91*, pages 24–33.

Eades, P., and Lin, X. (1995). A new heuristic for the feedback arc set problem. *Australian Journal of Combinatorics*, 12:15–26.

Eades, P., Lin, X., and Smyth, W. F. (1993). A fast and effective heuristic for the feedback arc set problem. *Information Processing Letters*, 47:319–323.

Eades, P., and Marks, J. (1995). Graph drawing contest report. In *Proceedings of the DIMACS International Workshop on Graph Drawing (GD'94)*, Springer LNCS 894, pages 143–146.

Eades, P., and Marks, J. (1996). Graph-drawing contest report. In *Proceedings of the 3rd International Symposium on Graph Drawing (GD'95)*, Springer LNCS 1027, pages 224–233.

Eades, P., Marks, J., and North, S. C. (1996). Graph-drawing contest report. In *Proceedings of the 4th International Symposium on Graph Drawing (GD'96)*, Springer LNCS 1190, pages 129–138.

Eades, P., Marks, J., and North, S. C. (1997c). Graph-drawing contest report. In *Proceedings of the 5th International Symposium on Graph Drawing (GD'97)*, Springer LNCS 1353, pages 438–445.

Eades, P., Marks, J., Mutzel, P., and North, S. C. (1998). Graph drawing contest report. In *Proceedings of the 6th International Symposium on Graph Drawing (GD'98)*, Springer LNCS 1547, pages 423–435.

Eades, P., Nagamochi, H., and Feng, Q. (1998). Straight-line drawing algorithms for hierarchical graphs and clustered graphs. Technical Report 98-03, Department of Computer Science and Software Engineering, University of Newcastle, Australia. Available at `ftp://ftp.cs.newcastle.edu.au/pub/techreports/tr98-03.ps.Z`.

Eades, P., Stirk, C., and Whitesides, S. (1996). The techniques of Kolmogorov and Bardzin for three-dimensional orthogonal graph drawing. *Information Processing Letters*, 60(2):97–103. University.

Eades, P., and Sugiyama, K. (1990). How to draw a directed graph. *Journal of Information Processing*, 13:424–437.

Eades, P., Symvonis, A., and Whitesides, S. (1996b). Two algorithms for three dimensional orthogonal graph drawing. In *Proceedings of the 4th International Symposium on Graph Drawing (GD'96)*. Springer LNCS 1190, pages 139–154.

Eades, P., Symvonis, A., and Whitesides, S. (2000). Three-dimensional orthogonal graph drawing. *Discrete Applied Mathematics*, 103(1-3):55–87.

Eades, P., and Whitesides, S. (1994). Drawing graphs in two layers. *Theoretical Computer Science*, 131(2):361–374.

Eades, P., and Wormald, N. C. (1990). Fixed edge-length graph drawing is NP-hard. *Discrete Applied Mathematics*, 28:111–134.

Eades, P., and Wormald, N. C. (1994). Edge crossings in drawings of bipartite graphs. *Algorithmica*, 11(4):379–403.

Edmondson, S., Christensen, J., Marks, J., and Shieber, S. (1997). A general cartographic labeling algorithm. *Cartographica*, 33(4):13–23.

Eiglsperger, M., Fößmeier, U., and Kaufmann, M. (2000). Orthogonal graph drawing with constraints. In *Proceedings of the 11th ACM-SIAM Symposium on Discrete Algorithms (SODA 2000)*, pages 3–11.

Eppstein, D., and Erickson, J. (1994). Iterated nearest neighbors and finding minimal polytopes. *Discrete Computational Geometry*, 11:321–350.

Even, S. (1979). *Graph Algorithms.* Pitman.

Even, S., and Tarjan, R. E. (1976). Computing an st-numbering. *Theoretical Computer Science*, 2:436–441.

Faria, L., De Figueiredo, C. M. H., and Mendonca, C. F. X. (1998). Splitting number is $\mathcal{NP}$-complete. *Proceedings of the 24th International Workshop on Graph-Theoretic Concepts in Computer Science (WG'98)*, Springer LNCS 1517, pages 285–297.

Fekete, S. P., and Meijer, H. (1999). Rectangle and box visibility graphs in 3D. *International Journal of Computational Geometry and Applications*, 9(1):1–27.

Feng, Q. (1997). *Algorithms for Drawing Clustered Graphs*. PhD thesis, University of Newcastle. `http://www.cs.newcastle.edu.au/Dept/theses.html`.

Feng, Q.-W., Cohen, R. F., and Eades, P. (1995). Planarity for clustered graphs. In *Proceedings of the 3rd European Symposium on Algorithms (ESA'95)*. Springer LNCS 979, pages 213–226.

Fialko, S., and Mutzel, P. (1998). A new approximation algorithm for the planar augmentation problem. In *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'98)*, pages 260–269.

Fisk, C. J., Caskey, D. L., and West, L. E. (1967). ACCEL: Automated circuit card etching layout. *Proceedings of the IEEE*, 55(11):1971–1982.

Foley, J. D., van Dam, A., Feiner, S. K., and Hughes, J. F. (1990). *Computer Graphics*, 2nd edition. Addison-Wesley.

Force, A. C. G. I. T. (1996). Application challenges to computational geometry. Technical Report TR-521-96, Princeton University.

Formann, M., Hagerup, T., Haralambides, J., Kaufmann, M., Leighton, F. T., Simvonis, A., Welzl, E., and Woeginger, G. (1990). Drawing graphs in the plane with high resolution. In *Proceedings of the 31st Symposium on the Foundations of Computer Science (FOCS'90)*, pages 86–95.

Formann, M., and Wagner, F. (1991). A packing problem with applications to lettering of maps. In *Proceedings of the 7th Annual Symposium on Computational Geometry (SCG '91)*, pages 281–288.

Formella, A., and Keller, J. (1995). Generalized fisheye views of graphs. In *Proceedings of the 3rd International Symposium on Graph Drawing (GD'95)*. Springer LNCS 1027, pages 242–253.

Fößmeier, U. (1997a). Interactive orthogonal graph drawing: Algorithms and bounds. In *Proceedings of the 5th International Symposium on Graph Drawing (GD'97)*. Springer LNCS 1353, pages 111–123.

Fößmeier, U. (1997b). *Orthogonale Visualisierungstechniken für Graphen*. PhD thesis, Eberhard-Karls-Universität zu Tübingen.

Fößmeier, U., Heß, C., and Kaufmann, M. (1998). On improving orthogonal drawings: The 4M-algorithm. In *Proceedings of the 6th International Symposium on Graph Drawing (GD'98)*. Springer LNCS 1547, pages 125–137.

Fößmeier, U., Kant, G., and Kaufmann, M. (1996). 2-visibility drawings of planar graphs. In *Proceedings of the 4th International Symposium on Graph Drawing (GD'96)*. Springer LNCS 1190, pages 155–168.

Fößmeier, U., and Kaufmann, M. (1995). Drawing high degree graphs with low bend numbers. In *Proceedings of the 3rd International Symposium on Graph Drawing (GD'95)*. Springer LNCS 1027, pages 254–266.

Foulds, L. R., Gibbons, P. B., and Giffin, J. W. (1985). Facilities layout adjacency determination: An experimental comparison of three graph theoretic heuristics *Operations Research*, 33:1091–1106.

Foulds, L. R., and Robinson, D. F. (1978). Graph theoretic heuristics for the plant layout problem. *International Journal of Production Research*, 16:27–37.

Fowler, R. J., Paterson, M. S., and Tanimoto, S. L. (1981). Optimal packing and covering in the plane are $\mathcal{NP}$-complete. *Information Processing Letters*, 12(3):133–137.

Freeman, L. C. (1999a). The social network graphics source. School of Social Science, University of California Irvine. `http://eclectic.ss.uci.edu/~lin/gallery.html`.

Freeman, L. C. (1999b). Using molecular modeling software in social network analysis: A practicum. School of Social Science, University of California Irvine. `http://eclectic.ss.uci.edu/~lin/chem.html`.

Freuder, E. C., and Wallace, R. J. (1992). Partial constraint satisfaction. *Artificial Intelligence*, 58(1-3):21–70.

Frick, A. (1997). Upper bounds on the number of hidden nodes in Sugiyama's algorithm. In *Proceedings of the 4th International Symposium on Graph Drawing (GD'96)*. Springer LNCS 1190, pages 169–183.

Frick, A., Ludwig, A., and Mehldau, H. (1995). A fast adaptive layout algorithm for undirected graphs. In *Proceedings of the DIMACS International Workshop on Graph Drawing (GD'94)*. Springer LNCS 894, pages 388–403.

Fruchterman, T. M. J., and Reingold, E. M. (1991). Graph-drawing by force-directed placement. *Software — Practice and Experience*, 21(11):1129–1164.

Gansner, E. R., Koutsofios, E., North, S. C., and Vo, K.-P. (1993). A technique for drawing directed graphs. *IEEE Transactions on Software Engineering*, 19(3):214–230.

Ganter, B., and Wille, R. (1999). *Formal Concept Analysis — Mathematical Foundations*. Springer.

Garey, M. R., and Johnson, D. S. (1983). Crossing number is $\mathcal{NP}$-complete. *SIAM Journal on Algebraic and Discrete Methods*, 4(3):312–316.

Garey, M. R., and Johnson, D. S. (1991). *Computers and Intractability: A Guide to the Theory of $\mathcal{NP}$–Completeness*. W.H. Freeman & Co.

Garg, A., Goodrich, M. T., and Tamassia, R. (1996). Planar upward tree drawings with optimal area. *International Journal Computational Geometry and Applications*, 6:333–356.

Garg, A., and Tamassia, R. (1993). Efficient computation of planar straight-line upward drawings. In *Graph Drawing '93 (Proc. ALCOM Workshop on Graph Drawing)*.

Garg, A., and Tamassia, R. (1994). Planar drawings and angular resolution: Algorithms and bounds. In *Proceedings of the 2nd European Symposium on Algorithms (ESA'94)*. Springer LNCS 855, pages 12–23.

Garg, A., and Tamassia, R. (1995a). On the computational complexity of upward and rectilinear planarity testing. In *Proceedings of the DIMACS International Workshop on Graph Drawing (GD'94)*. Springer LNCS 894, pages 286–297.

Garg, A., and Tamassia, R. (1995b). Upward planarity testing. *Order*, 12:109–133.

Garg, A., and Tamassia, R. (1996a). GIOTTO3D: A system for visualizing hierarchical structures in 3D. In *Proceedings of the 4th International Symposium on Graph Drawing (GD'96)*. Springer LNCS 1190, pages 193–200.

Garg, A., and Tamassia, R. (1996b). A new minimum cost flow algorithm with applications to graph drawing. In *Proceedings of the 4th International Symposium on Graph Drawing (GD'96)*. Springer LNCS 1190, pages 201–216.

Garg, A., and Tamassia, R. (1997). A new minimum cost flow algorithm with applications to graph drawing. In *Proceedings of the 4th International Symposium on Graph Drawing (GD'96)*. Springer LNCS 1190, pages 201–216.

Georgakopoulos, D., Hornick, M., and Sheth, A. (1995). An overview of workflow management: From process modeling to workflow automation infrastructure. *Distributed and Parallel Databases*, 3(2):119–153.

German Research Center for Artificial Intelligence GmbH (1999). The Verbmobil project. `http://www.dfki.de/verbmobil`.

Godehardt, E. (1988). *Graphs as Structural Models*, Advances in System Analysis 4. Vieweg.

Goldberg, A. V., and Kennedy, R. (1995). An efficient cost scaling algorithm for the assignment problem. *Mathematical Programming*, 71:153–178.

Goldstein, A. J. (1963). An efficient and constructive algorithm for testing whether a graph can be embedded in a plane. In *Graph and Combinatorics Conference, Contract No. NONR 1858-(21)*. Princeton University.

Grötschel, M., Jünger, M., and Reinelt, G. (1985). On the acyclic subgraph polytope. *Mathematical Programming*, 33(1):28–42.

Gutwenger, C., and Mutzel, P. (1998). Planar polyline drawings with good angular resolution. In *Proceedings of the 6th International Symposium on Graph Drawing (GD'98)*. Springer LNCS 1547, pages 167–182.

Hayashi, K., Inoue, M., Masuzawa, T., and Fujiwara, H. (1998). A layout adjustment problem for disjoint rectangles preserving orthogonal order. In *Proceedings of the 6th International Symposium on Graph Drawing*, number 1547 in LNCS, pages 183–197.

He, W., and Marriott, K. (1998). Constrained graph layout. *Constraints*, 3(4):289–314.

Herdeg, W., editor (1981). *Diagrams*. Graphis Press Corporation.

Hermansson, K., and Ojamae, L. (1994). MOVIEMOL — An easy-to-use molecular display and animation program. Technical Report UUIC-B19-500, Institute of Chemistry, University of Uppsala.

Hochbaum, D. S. (1995). *Approximation Algorithms for $\mathcal{NP}$-hard Problems.* PWS Publishing Company, Boston.

Hochbaum, D. S., and Maass, W. (1985). Approximation schemes for covering and packing problems in image processing and VLSI. *Journal of the ACM*, 32(1):130–136.

Hong, S.-H., Eades, P., Quigley, A., and Lee, S.-H. (1998). Drawing algorithms for series-parallel digraphs in two and three dimensions. In *Proceedings of the 6th International Symposium on Graph Drawing (GD'98).* Springer LNCS 1547, pages 198–209.

Hong, S.-H., Eades, P., Quigley, A., and Lee, S.-H. (1999a). Drawing series-parallel digraphs symmetrically. To appear in *International Journal of Computational Geometry and Applications.*

Hong, S.-H., Eades, P., Quigley, A., and Lee, S.-H. (1999b). A three dimensional drawing algorithm for series-parallel graphs. Manuscript.

Hopcroft, J., and Tarjan, R. E. (1974). Efficient planarity testing. *Journal of the ACM*, 21:549–568.

Hopcroft, J. E., and Tarjan, R. E. (1973). Dividing a graph into triconnected components. *SIAM Journal on Computing*, 2(3):135–158.

Hsu, W.-L. (1995). A linear time algorithm for finding maximal planar subgraphs. In *Proceedings of the 6th International Symposium on Algorithms and Computation (ISAAC'95).* Springer LNCS 1004, pages 352–361.

Huang, M. L., and Eades, P. (1998a). A fully animated interactive system for clustering and navigating huge graphs. In *Proceedings of the 6th International Symposium on Graph Drawing (GD'98)*, Springer LNCS 1547, pages 374–383.

Hughes, J. G. (1993). *Object-Oriented Databases.* International Series in Computer Science. Prentice-Hall.

Humphrey, W., Dalke, A., and Schulten, K. (1996). VMD — Visual molecular dynamics. *Journal of Molecular Graphics*, 14(1):33–38.

Hutton, M. D., and Lubiw, A. (1991). Upward planar drawing of single source acyclic digraphs. In *Proceedings of the 2nd ACM-SIAM Symposium on Discrete Algorithms (SODA'91)*, pages 203–211.

Imai, H., and Asano, T. (1983). Finding the connected components and a maximum clique of an intersection graph of rectangles in the plane. *Journal of Algorithms*, 4:310–323.

Imai, H., and Asano, T. (1986). Efficient algorithms for geometric graph search problems. *SIAM Journal on Computing*, 15(2):478–494.

Imhof, E. (1962). Die Anordnung der Namen in der Karte. *International Yearbook of Cartography*, 2:93–129.

Imhof, E. (1975). Positioning names on maps. *The American Cartographer*, 2(2):128–144.

Indermark, K., Thomas, W., Huch, F., Leucker, M., and Noll, T. (1999). Various texts about the TRUTH system for modelling concurrent systems, Lehrstuhl für Informatik II, RWTH Aachen. `http://www-i2.informatik.rwth-aachen.de/Forschung/MCS/`.

Isoda, S., Shimomura, T., and Ono, Y. (1987). VIPS: A visual debugger. *IEEE Software*, 4(3):8–19.

Iturriaga, C., and Lubiw, A. (1997). $\mathcal{NP}$-hardness of some map labeling problems. Technical Report CS-97-18, University of Waterloo.

Jain, A. K., and Dubes, R. C. (1988). *Algorithms for Clustering Data*. Prentice Hall.

Johnson, D. S. (1982). The $\mathcal{NP}$-completeness column: An ongoing guide. *Journal of Algorithms*, 3:89–99.

Jünger, M., Lee, E. K., Mutzel, P., and Odenthal, T. (1997). A polyhedral approach to the multi-layer crossing minimization problem. In *Proceedings of the 5th International Symposium on Graph Drawing (GD'97)*. Springer LNCS 1353, pages 13–24.

Jünger, M., and Mutzel, P. (1996). Maximum planar subgraphs and nice embeddings: Practical layout tools. *Algorithmica*, 16:33–59.

Jünger, M., and Mutzel, P. (1997). 2-Layer straightline crossing minimization: Performance of exact and heuristic algorithms. *Journal on Graph Algorithms and Applications*, 1(1):1–25.

Kakoulis, K. G., and Tollis, I. G. (1996). On the edge label placement problem. In *Proceedings of the 4th International Symposium on Graph Drawing (GD'96)*. Springer LNCS 1190, pages 241–256.

Kakoulis, K. G., and Tollis, I. G. (1997). An algorithm for labeling edges of hierarchical drawings. In *Proceedings of the 5th International Symposium on Graph Drawing (GD'97)*. Springer LNCS 1353, pages 169–180.

Kakoulis, K. G., and Tollis, I. G. (1998a). On the multiple label placement problem. In *Proceedings of the 10th Canadian Conference on Computational Geometry (CCCG'98)*, pages 66–67.

Kakoulis, K. G., and Tollis, I. G. (1998b). A unified approach to labeling graphical features. In *Proceedings of the 14th Annual ACM Symposium on Computional Geometry (SCG'98)*, pages 347–356.

Kamada, T., and Kawai, S. (1988). A simple method for computing general positions in displaying three-dimensional objects. *Computer Vision, Graphics and Image Processing*, 41:43–56.

Kamada, T., and Kawai, S. (1989). An algorithm for drawing general undirected graphs. *Information Processing Letters*, 31:7–15.

Kant, G. (1996). Drawing planar graphs using the canonical ordering. *Algorithmica*, 16:4–32.

Kant, G., and Bodlaender, H. L. (1991). Planar graph augmentation problems. In *Proceedings of the 2nd Workshop on Algorithms and Data Structures (WADS'91)*, Springer LNCS 519, pages 286–298.

Karp, R. (1972). Reducibility among combinatorical problems. In *Complexity of Computer Computations*, pages 85–103. Plenum Press.

Kato, T., and Imai, H. (1988). The $\mathcal{NP}$-completeness of the character placement problem of 2 or 3 degrees of freedom. In *Record of Joint Conference of Electrical and Electronic Engineers in Kyushu*, page 1138.

Keahey, T. A., and Robertson, E. (1996). Techniques for non-linear maginifaction transformations. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis'96)*, pages 38–45.

Kedem, G., and Watanabe, H. (1984). Graph optimization techniques for IC-layout and compaction. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, CAD-3(1):12–20.

Kelly, D., and Rival, I. (1975). Planar lattices. *Canadian Journal of Mathematics*, 27:636–665.

Kenis, P. (1999). Analysing social network data by means of visualisation techniques. Paper presented at the *19th International Conference on Social Network Analysis (Sunbelt XIX)*, Charleston.

Kernighan, B. W., and Lin, S. (1970). An efficient heuristic procedure for partitioning graphs. *The Bell System Technical Journal*, 49(2):291–307.

Kirchhoff, G. R. (1847). Über die Auflösung der Gleichungen, auf welche man bei der Untersuchung der linearen Verteilung galvanischer Ströme geführt wird. *Annalen der Physik und Chemie*, 72:497–508.

Klau, G. W., and Mutzel, P. (1998). Quasi-orthogonal drawing of planar graphs. Technical Report 98-1-013, Max-Planck-Institut für Informatik, Saarbrücken.

Klau, G. W., and Mutzel, P. (1999a). Combining graph labeling and compaction. In *Proceedings of the 7th International Symposium on Graph Drawing (GD'99)*. Springer LNCS 1731, pages 27–37.

Klau, G. W., and Mutzel, P. (1999b). Optimal compaction of orthogonal grid drawings. In *Integer Programming and Combinatorial Optimization (IPCO'99)*. Springer LNCS 1610, pages 304–319.

Knipping, L. (1998). Beschriftung von Linienzügen. Diplomarbeit, Fachbereich Mathematik und Informatik, Freie Universität Berlin.

Knuth, D. E., and Raghunathan, A. (1992). The problem of compatible representatives. *SIAM Journal on Discrete Mathematics*, 5(3):422–427.

Koebe, P. (1936). Kontaktprobleme auf der konformen Abbildung. *Berichte über die Verhandlungen der Sächsischen Akademie der Wissenschaften zu Leipzig, Mathematisch-Physikalische Klasse*, 88:141–164.

Kolmogorov, A. N., and Bardzin, Y. M. (1967). About realization of sets in 3-dimensional space. *Problems in Cybernetics*, pages 261–268.

Kosak, C., Marks, J., and Shieber, S. (1994). Automating the layout of network diagrams with specified visual organization. *IEEE Transactions on Systems, Man and Cybernetics*, 24(3):440–454.

Krackhardt, D., Blythe, J., and McGrath, C. (1994). KrackPlot 3.0: An improved network drawing program. *Connections*, 17(2):53–55.

Kruskal, J. B., and Wish, M. (1978). *Multidimensional Scaling.* Sage University Paper Series on Quantitative Applications in the Social Sciences 07-011.

Kucera, L., Mehlhorn, K., Preis, B., and Schwarzenecker, E. (1993). Exact algorithms for a geometric packing problem. *Proceedings of the 10th Symposium on the Theoretical Aspects of Computer Science (STACS'93).* Springer LNCS 665, pages 317–322.

Kumar, A., and Fowler, R. H. (1994). A spring modelling algorithm to position nodes of an undirected graph in three dimensions. Technical report, Department of Computer Science, University of Texas.

Laguna, M., and Martí, R. (1999). Grasp and path relinking for 2-layer straight line crossing minimization. *INFORMS Journal on Computing*, 11(1):44–52.

Laguna, M., Martí, R., and Valls, V. (1997). Arc crossing minimization in hierarchical digraphs with tabu search. *Computers and Operations Research*, 24(12):1175–1186.

Lam, S., and Sethi, R. (1977). Worst case analysis of two scheduling problems. *SIAM Journal on Computing*, 6:518–536.

LaPaugh, A. S. (1998). VLSI Layout Algorithms. In *Algorithms and Theory of Computation Handbook*. CRC Press.

Leiserson, C. E. (1980). Area-efficient graph layouts (for VLSI). In *Proceedings of the 21st Annual IEEE Symposium on Foundations of Computer Science (FOCS'80)*, pages 270–281.

Lempel, A., and Cederbaum, I. (1966). Minimum feedback arc and vertex sets of a directed graph. *IEEE Transactions on Circuit Theory*, CT–13(4):339–403.

Lempel, A., Even, S., and Cederbaum, I. (1967). An algorithm for planarity testing of graphs. In *Theory of Graphs: International Symposium (Rome 1966)*, pages 215–232. Gordon and Breach.

Lengauer, T. (1989). Hierarchical planarity testing algorithms. *Journal of the ACM*, 36:474–509.

Lengauer, T. (1990). *Combinatorial Algorithms for Integrated Circuit Layout.* Applicable Theory in Computer Science. B. G. Teubner and John Wiley & Sons.

Leung, J. (1992). A new graph-theoretic heuristic for facility layout. *Management Science*, 38(4):594–605.

Lewis, J. M., and Yannakakis, M. (1980). The node-deletion problem for hereditary properties is $\mathcal{NP}$-complete. *Journal of Computer and System Sciences*, 20(2):219–230.

Liebers, A. (1996). Methods for planarizing graphs - A survey and annotated bibliography. Technical Report Konstanzer Schriften in Mathematik und Informatik Nr. 12, Fakultät für Mathematik und Informatik, Universität Konstanz. ISSN 1430-3558. To appear in *Journal on Graph Algorithms and Applications.*

Lin, X. (1992). *Analysis of Algorithms for Drawing Graphs.* PhD thesis, University of Queensland.

Lino, P., Martí, R., and Valls, V. (1996). A branch and bound algorithm for minimizing the number of crossing arcs in bipartite graphs. *Journal of Operational Research*, 90:303–319.

Lipton, R. J., Rose, D. J., and Tarjan, R. E. (1979). Generalized nested dissection. *SIAM Journal on Numerical Analysis*, 16:346–358.

Lipton, R. J., and Tarjan, R. E. (1970). A seperator theorem for planar graphs. In *Proceedings of the Conference on Theoretical Computer Science*, pages 1–10.

Liu, P. C., and Geldmacher, R. C. (1977). On the deletion of nonplanar edges of a graph. In *Proceedings of the 10th Southeastern Conference on Combinatorics, Graph Theory, and Computing*, pages 727–738.

Lyons, K. A. (1992). Cluster busting in anchored graph drawing. In *Proceedings of the '92 CAS Conference (CASCON'92)*, pages 7–17.

Lyons, K. A., Meijer, H., and Rappaport, D. (1998). Algorithms for cluster busting in anchored graph drawing. *Journal on Graph Algorithms and Applications*, 2(1):1–24.

Mackworth, A. K., and Freuder, E. C. (1985). The complexity of some polynomial network consistency algorithms for constraint satisfaction problem. *Artificial Intelligence*, 25(1):65–74.

Mäkinen, E. (1990). Experiments on drawing 2-level hierarchical graphs. *International Journal of Computer and Mathematics*, 36:175–181.

Mäkinen, E., and Sieranta, M. (1994). Genetic algorithms for drawing bipartite graphs. *Internatonal Journal of Computer Mathematics*, 53:157–166.

Malitz, S., and Papakostas, A. (1992). On the angular resolution of planar graphs. In *Proceedings of the 24th Annual ACM Symposium on the Theory of Computing (STOC'92)*, pages 527–538.

Malitz, S., and Papakostas, A. (1994). On the angular resolution of planar graphs. *SIAM Journal on Discrete Mathematics*, 7(2):172–183.

Manning, J. (1990). *Geometric Symmetry in Graphs.* PhD thesis, Purdue University.

Marks, J., and Shieber, S. (1991). The computational complexity of cartographic label placement. Technical Report TR-05-91, Harvard University Computer Science.

Masuda, S., Kimura, S., Kashiwabara, T., and Fujisawa, T. (1983). On the Manhattan wiring problem. Technical Report CAS 83-20, Institute of Electronics and Communication Engineers of Japan.

Masui, T. (1992). Graphic object layout with interactive genetic algorithms. In *Proceedings of the 1992 IEEE Workshop on Visual Languages (VL'92)*, pages 74–87.

Matuszewski, C., Schönfeld, R., and Molitor, P. (1999). Using sifting for k-layer straightline crossing minimization. *Proceedings of the 7th Symposium on Graph Drawing (GD'99).* Springer LNCS 1731, pages 217–224.

McGrath, C., Blythe, J., and Krackhardt, D. (1996). Seeing groups in graph layouts. *Connections*, 19(2):22–29.

McGrath, C., and Borgatti, S. P. (1999). *The International Network for Social Network Analysis Homepage.* `http://www.heinz.cmu.edu/project/INSNA/`.

Mehlhorn, K. (1984). *Data Structures and Algorithms. Volume 2: Graph Algorithms and $\mathcal{NP}$-Completeness.* EATCS Monographs on Theoretical Computer Science. Springer.

Mehlhorn, K., and Näher, S. (1999). *The* LEDA *Platform of Combinatorial and Geometric Computing.* Cambridge University Press. Project home page at `http://www.mpi-sb.mpg.de/LEDA/`.

Messinger, E. B., Rowe, L. A., and Henry, R. H. (1991). A divide-and-conquer algorithm for the automatic layout of large directed graphs. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-21(1):1–12.

Miriyala, K., Hornik, S. W., and Tamassia, R. (1993). An incremental approach to aesthetic graph layout. In *Proceedings of the 6th International Workshop on Computer-Aided Software Engineering (CASE'93)*, pages 297–308.

Misue, K., Eades, P., Lai, W., and Sugiyama, K. (1995). Layout adjustment and the mental map. *Journal of Visual Languages and Computing*, 6:183–210.

Moen, S. (1990). Drawing dynamic trees. *IEEE Software*, 7:21–28.

Monien, B., Ramme, F., and Salmen, H. (1995). A parallel simulated annealing algorithm for generating 3D layouts of undirected graphs. In *Proceedings of the 3rd International Symposium on Graph Drawing (GD'95).* Springer LNCS 1027, pages 396–408.

Monien, B., Ramme, F., and Salmen, H. (1996). A parallel simulated annealing algorithm for generating 3d layouts of undirected graphs. In *Proceedings of the 3rd International Symposium on Graph Drawing (GD'95).* Springer LNCS 1027, pages 396–408.

MTA (1999). MTA New York City subway map. `http://www.mta.nyc.ny.us/nyct/images/sub1a.gif` and `http://www.mta.nyc.ny.us/nyct/images/sub2a.gif`.

Mukherjea, S., Foley, J., and Hudson, S. (1994). Interactive clustering for navigating in hypermedia systems. In *Proceedings of the ACM European Conference on Hypermedia Tehcnologie.*

Mutzel, P. (1994). *The Maximum Planar Subgraph Problem.* PhD thesis, Universität zu Köln.

Mutzel, P. (1995). A polyhedral approach to planar augmentation and related problems. In *Proceedings of the 3rd European Symposium on Algorithms (ESA'95).* Springer LNCS 979, pages 494–507.

Mutzel, P. (1997). An alternative method to crossing minimization on hierarchical graphs. In *Proceedings of the 4th International Symposium on Graph Drawing (GD'96).* Springer LNCS 1190, pages 318–333.

Nakano, S., Rahman, M. S., and Nishizeki, T. (1997). A linear-time algorithm for four-partitioning four-connected planar graphs. In *Proceedings of the 5th International Symposium on Graph Drawing (GD'97)*. Springer LNCS 1353, pages 334 – 344.

Nemhauser, G. L., and Sigismondi, G. (1992). A strong cutting plane/branch-and-bound algorithm for node packing. *Journal of the Operational Research Society*, 43:443–457.

Nielsen, J. (1990). The art of navigating throuh hypertext. *Communications of the ACM*, 33(3):296–310.

Nishizeki, T., and Chiba, N. (1988). *Planar Graphs: Theory and Algorithms.* North-Holland Mathematics Studies 140/32.

North, S. C. (1996). Incremental layout with DynaDag. In *Proceedings of the 3rd International Symposium on Graph Drawing (GD'95)*. Springer LNCS 1027, pages 409–418.

Oerder, M., and Ney, H. (1993). Word graphs: An efficient interface between continuous-speech recognition and language understanding. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP'93)*, volume II, pages 119–122.

Ostry, D. (1996). Some three-dimensional graph drawing algorithms. Master's thesis, University of Newcastle.

Otten, R. H. J. M., and van Wijk, J. G. (1978). Graph representation in interactive layout design. In *Proceedings of the IEEE International Symposium on Circuits and Systems*, pages 914–918.

Papakostas, A. (1995). Upward planarity testing of outerplanar dags. In *Proceedings of the DIMACS International Workshop on Graph Drawing (GD'94)*. Springer LNCS 894, pages 298–306.

Papakostas, A., Six, J. M., and Tollis, I. G. (1996). Experimental and theoretical results in interactive orthogonal graph drawing. In *Proceedings of the 4th International Symposium on Graph Drawing (GD'96)*. Springer LNCS 1190, pages 371–386.

Papakostas, A., and Tollis, I. G. (1997a). Incremental orthogonal graph drawing in three dimensions. In *Proceedings of the 5th International Symposium on Graph Drawing (GD'97)*. Springer LNCS 1353, pages 52–63.

Papakostas, A., and Tollis, I. G. (1997b). Incremental orthogonal graph drawing in three-dimensions. Technical Report UTDCS-02-97, Dept. of Computer Sciencs, University of Texas at Dallas.

Papakostas, A., and Tollis, I. G. (1997c). Orthogonal drawing of high degree graphs with small area and few bends. In *Proceedings of the 5th Workshop on Algorithms and Data Structures (WADS'97)*. Springer LNCS 1272, pages 354–367.

Papakostas, A., and Tollis, I. G. (1997d). A pairing technique for area-efficient orthogonal drawings. In *Proceedings of the 4th International Symposium on Graph Drawing (GD'96)*. Springer LNCS 1190, pages 354–370.

Papakostas, A., and Tollis, I. G. (1998). Interactive orthogonal graph drawing. *IEEE Transactions on Computers*, 47(11):1297–1309.

Patrignani, M. (1999a). On the complexity of orthogonal compaction. Technical Report RT–DIA–39–99, Dipartimento di Informatica e Automazione, Università degli Studi di Roma Tre.

Patrignani, M. (1999b). On the complexity of orthogonal compaction. *Proceedings of the 6th Workshop on Algorithms and Data Structures (WADS'99)*. Springer LNCS 1663, pages 56–61.

Patrignani, M., and Vargiu, F. (1997). 3DCube: A tool for the three dimensional graph drawing. In *Proceedings of the 5th International Symposium on Graph Drawing (GD'97)*. Springer LNCS 1353, pages 284–290.

Paulish, F. N. (1993). *The Design of an Extendible Graph Editor*. Springer LNCS 704.

Platt, C. (1976). Planar lattices and planar graphs. *Journal of Combinatorial Theory, Series B*, 21:30–39.

Poon, C. K., Zhu, B., and Chin, F. (1998). A polynomial time solution for labeling a rectilinear map. *Information Processing Letters*, 65:201–207.

Poutré, J. A. L. (1994). Alpha-algorithms for incremental planarity testing. In *Proceedings of the 26th Annual ACM Symposium on the Theory of Computation (STOC'94)*, pages 706–715.

Purchase, H. C. (1997). Which aesthetic has the greatest effect on human understanding? In *Proceedings of the 5th International Symposium on Graph Drawing (GD'97)*. Springer LNCS 1353, pages 248–261.

Purchase, H. C., Cohen, R. F., and James, M. (1996). Validating graph drawing aesthetics. In *Proceedings of the 3rd International Symposium on Graph Drawing (GD'95)*. Springer LNCS 1027, pages 435–446.

Purchase, H. C., Cohen, R. F., and James, M. (1997). An experimental study of the basis for graph drawing algorithms. *ACM Journal of Experimental Algorithmics*, 2(4).

Quinn, N. R., and Breuer, M. A. (1979). A force directed component placement procedure for printed circuit boards. *IEEE Transactions on Circuits and Systems*, 26(6):377–388.

Reeves, C. M. (1995). *Modern Heuristic Techniques for Combinatorial Problems*. McGraw-Hill.

Reggiani, M. G., and Marchetti, F. E. (1988). A proposed method for representing hierarchies. *IEEE Transactions on Systems, Man, and Cybernetics*, 18(1):2–8.

Reinelt, G. (1985). *The linear ordering problem: algorithms and applications*. Research and Exposition in Mathematics 8, Heldermann.

Reingold, E. M., and Tilford, J. S. (1981). Tidier drawings of trees. *IEEE Transactions on Software Engineering*, 7(2):223–228.

Rival, I. (1985). The diagram. In *Graphs and Order*, NATO ASI Series, pages 103–133. Reidel Publishing.

Robertson, G. G., Mackinlay, J. D., and Card, S. K. (1993). Cone trees: Animated 3*d* visualizations of hierarchical information. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 189–193.

Rosenstiehl, P., and Tarjan, R. E. (1986). Rectilinear planar layouts of planar graphs and bipolar orientations. *Discrete & Computational Geometry*, 1(4):342–351.

Roxborough, T., and Sen, A. (1997). Graph clustering using multiway ratio cut. In *Proceedings of the 5th International Symposium on Graph Drawing (GD'97)*. Springer LNCS 1353, pages 291–296.

Rudell, R. (1993). Dynamic variable ordering for ordered binary decision diagrams. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD'93)*, pages 42–47.

Sablowski, R., and Frick, A. (1996). Automatic graph clustering. In *Proceedings of the 4th International Symposium on Graph Drawing (GD'96)*. Springer LNCS 1190, pages 395–400.

Sander, G. (1994). Graph layout through the VCG tool. Technical Report A03/94, Universität des Saarlandes.

Sander, G. (1996a). A fast heuristic for hierarchical Manhattan layout. In *Proceedings of the 3rd International Symposium on Graph Drawing (GD'95)*. Springer LNCS 1027, pages 447–458.

Sander, G. (1996b). Graph layout for applications in compiler construction. Technical Report A/01/96, FB 14 Informatik, Universität des Saarlandes.

Sarkar, M., and Brown, M. H. (1994). Graphical fisheye views. *Communications of the ACM*, 37(12):73–84.

Schlag, M., Liao, Y.-Z., and Wong, C. K. (1983). An algorithm for optimal two-dimensional compaction of VLSI layouts. *Integration, the VLSI Journal*, 1:179–209.

Schnyder, W. (1990). Embedding planar graphs on the grid. In *Proceedings of the 1st ACM-SIAM Symposium on Discrete Algorithms (SODA'90)*, pages 138–148.

Schrijver, A. (1986). *Theory of Linear and Integer Programming*. Wiley-Interscience.

Sedgewick, R. (1988). *Algorithms*, pages 438–441. Addison–Wesley, 2nd edition.

Shiloach, Y. (1976). *Arrangements of planar graphs on the planar lattice*. PhD thesis, Weizmann Institute of Science.

Sim, S. (1996). Automatic graph drawing algorithms. Manuscript, available at `http://www.cs.toronto.edu/~simsuz/papers/grafdraw.ps.gz`.

Six, J. M., Kakoulis, K. G., and Tollis, I. G. (1998). Refinement of orthogonal graph drawings. In *Proceedings of the 6th International Symposium on Graph Drawing (GD'98)*. Springer LNCS 1547, pages 302–315.

Strijk, T., and van Kreveld, M. (1999). Labeling a rectilinear map more efficiently. *Information Processing Letters*, 69(1):25–30.

Strijk, T., and Wolf, A. (1999). Labeling points with circles. Technical Report TR-99-08, Institut für Informatik, Freie Universität Berlin.

Stumme, G., and Wille, R. (1995). A geometrical heuristic for drawing concept lattices. In *Proceedings of the DIMACS International Workshop on Graph Drawing (GD'94)*. Springer LNCS 894, pages 452–460.

Sugiyama, K. (1987). A cognitive approach for graph drawing. *Cybernetic Systems*, 18(6):447–488.

Sugiyama, K., and Misue, K. (1991). Visualisation of structural information: Automatic drawing of compound digraphs. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(4):876–892.

Sugiyama, K., and Misue, K. (1995). A simple and unified method for drawing graphs: Magnetic-spring algorithm. In *Proceedings of the DIMACS International Workshop on Graph Drawing (GD'94)*. Springer LNCS 894, pages 364–375.

Sugiyama, K., Tagawa, S., and Toda, M. (1981). Methods for visual understanding of hierarchical system structures. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(2):109–125.

Supowit, K. J., and Reingold, E. M. (1983). The complexity of drawing trees nicely. *Acta Informatica*, 18:377–392.

Tamassia, R. (1987). On embedding a graph in the grid with the minimum number of bends. *SIAM Journal on Computing*, 16(3):421–444.

Tamassia, R. (1998). Constraints in graph drawing algorithms. *Constraints*, 3(1):87–120.

Tamassia, R., Di Battista, G., and Batini, C. (1988). Automatic graph drawing and readability of diagrams. *IEEE Transactions on Systems, Man, and Cybernetics*, 18(1):61–79.

Tamassia, R., and Tollis, I. G. (1986). A unified approach to visibility representations of planar graphs. *Discrete & Computational Geometry*, 1(4):321–341.

Tamassia, R., and Tollis, I. G. (1989). Planar grid embedding in linear time. *IEEE Transactions on Circuits and Systems*, 36(9):1230–1234.

Tamassia, R., Tollis, I. G., and Vitter, J. S. (1991). Lower bounds for planar orthogonal drawings of graphs. *Information Processing Letters*, 39(1):35–40.

Tanenbaum, A. S. (1995). *Distributed Operating Systems*. Prentice Hall.

Tarjan, R. E. (1983). *Data structures and network algorithms* CBMS-NSF Regional Conference Series in Applied Mathematics 44, SIAM.

Thomassen, C. (1980). Planarity and duality of finite and infinite planar graphs. *Journal of Combinatorial Theory, Series B*, 29:244–271.

Thompson, C. D. (1980). *A Complexity Theory for VLSI*. PhD thesis, Carnegie Mellon University.

Tunkelang, D. (1994). A practical approach to drawing undirected graphs. Technical Report CMU-CS-94-161, School of Computer Science, Carnegie Mellon University.

Tutte, W. T. (1960). Convex representations of graphs. *Proceedings of the London Mathematical Society, Third Series*, 10:304–320.

Tutte, W. T. (1963). How to draw a graph. *Proceedings of the London Mathematical Society, Third Series*, 13:743–768.

Ullman, J. (1989). *Principles of Database and Knowledgebase Systems*, volume 1. Computer Science Press.

Utech, J., Branke, J., Schmeck, H., and Eades, P. (1998). An evolutionary algorithm for drawing directed graphs. In *Proceedings of the International Conference on Imaging Science, Systems, and Technology*, pages 154–160.

Valdes, J., Tarjan, R. E., and Lawler, E. L. (1982). The recognition of series parallel digraphs. *SIAM Journal on Computing*, 11:298–313.

Valiant, L. (1981). Universality considerations in VLSI circuits. *IEEE Transactions on Computers*, C-30(2):135–140.

van Kreveld, M., Strijk, T., and Wolff, A. (1998). Point set labeling with sliding labels. In *Proceedings of the 14th Annual ACM Symposium on Computational Geometry (SCG'98)*, pages 337–346.

Verweij, B., and Aardal, K. (1999). An optimisation algorithm for maximum independent set with applications in map labelling. In *Proceedings of the 7th European Symposium on Algorithms (ESA'99)*. Springer LNCS 1643, pages 426–437.

Vogt, F. (1996). *Formale Begriffsanalyse mit C++: Datenstrukturen und Algorithmen*. Springer.

Vogt, F., and Wille, R. (1995). TOSCANA — a graphical tool for analyzing and exploring data. In *Proceedings of the DIMACS International Workshop on Graph Drawing (GD'94)*. Springer LNCS 894, pages 226–233.

Vossen, G. (1991). *Datenbankmodelle, Datenbanksprachen und Datenbankmanagement-Systeme*. Addison-Wesley.

Wagner, F. (1994). Approximate map labeling is in $\Omega(n \log n)$. *Information Processing Letters*, 52(3):161–165.

Wagner, F., and Wolff, A. (1995a). An efficient and effective approximation algorithm for the map labeling problem. In *Proceedings of the 3rd European Symposium on Algorithms (ESA'95)*. Springer LNCS 979, pages 420–433.

Wagner, F., and Wolff, A. (1995b). Map labeling heuristics: Provably good and practically useful. In *Proceedings of the 11th Annual ACM Symposium on Computational Geometry (SCG'95)*, pages 109–118.

Wagner, F., and Wolff, A. (1997). A practical map labeling algorithm. *Computational Geometry: Theory and Applications*, 7:387–404.

Wagner, F., and Wolff, A. (1998). A combinatorial framework for map labeling. In *Proceedings of the 6th International Symposium on Graph Drawing (GD'98)*. Springer LNCS 1547, pages 316–331.

Wang, X., and Miyamoto, I. (1995). Generating cunstomized layouts. In *Proceedings of the 3rd International Symposium on Graph Drawing (GD'95)*. Springer LNCS 1027, pages 504–515.

Wang, X., and Miyamoto, I. (1996). Generating customized layouts. In *Proceedings of the 3rd International Symposium on Graph Drawing (GD'95).* Springer LNCS 1027, pages 504–515.

Warfield, J. (1977). Crossing theory and hierarchy mapping. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-7(7):502–523.

Warnke, V., Kompe, R., Niemann, H., and Nöth, E. (1997). Integrated dialog act segmentation and classification using prosodic features and language models. Technical Report Verbmobil-Report 218, Lehrstuhl für Mustererkennung 5, Universität Erlangen-Nürnberg.

Wasserman, S., and Faust, K. (1994). *Social Network Analysis: Methods and Applications.* Cambridge University Press.

Watanabe, H. (1984). *IC Layout Generation and Compaction Using Mathematical Optimization.* PhD thesis, University of Rochester.

Watanabe, T., Ae, T., and Nakamura, A. (1983). On the $\mathcal{NP}$-hardness of edge-deletion and -contraction problems. *Discrete Applied Mathematics*, 6:63–78.

Webber, R. (1997). Finding the best viewpoints for three-dimensional graph drawings. In *Proceedings of the 5th International Symposium on Graph Drawing (GD'97).* Springer LNCS 1353, pages 87–98.

Webber, R. (1998). *Finding the Best Viewpoint for Three-Dimensional Graph Drawings.* PhD thesis, University of Newcastle. `http://www.cs.mu.oz.au/~rwebber/research/thesis/`.

Wei, Y.-C., and Cheng, C.-K. (1991). Ratio cut partitioning for hierarchical designs. *IEEE Transactions on Computer-Aided Design*, 10(7):911–921.

West, D. (1996). *Introduction to Graph Theory.* Prentice Hall.

White, D. (1999). Pgraph of Canaan genealogy made by Pajek program. Manuscript. `http://eclectic.ss.uci.edu/~drwhite/pgraph/p-graphs.html`.

Wiese, R., and Kaufmann, M. (1998). Adding constraints to an algorithm for orthogonal graph drawing. In *Proceedings of the 6th International Symposium on Graph Drawing (GD'98).* Springer LNCS 1547, pages 462–463.

Wille, R. (1989). Lattices in data analysis: How to draw them with a computer. In *Algorithms and Order*, NATO ASI Series, pages 33–58. Kluwer Academic Publishers.

Wille, R. (1997). Introduction to formal concept analysis. In *Modelli e modellizzazione. Models and modelling.* Consiglio Nazionale delle Ricerche, Instituto di Studi sulli Ricerca e Documentazione Scientifica, Roma, pages 39–51.

Winter, A., and Schürr, A. (1997). Modules and updatable graph views for programmed graph rewriting systems. Technical Report AIB 97-3, Lehrstuhl für Informatik III, RWTH Aachen.

Wolff, A. (1999). *Map Labeling in Theory and Practice.* PhD thesis, Freie Universität Berlin.

Wolff, A., Knipping, L., van Kreveld, M., Strijk, T., and Agarwal, P. K. (1999). A simple and efficient algorithm for high-quality line labeling. In *Proceedings of GISRUK'99*.

Wood, D. (1998a). An algorithm for three-dimensional orthogonal graph drawing. In *Proceedings of the 6th International Symposium on Graph Drawing (GD'98)*. Springer LNCS 1547, pages 332–346.

Wood, D. (1998b). Two-bend three-dimensional orthogonal grid drawing of maximum degree five graphs. Technical Report 98/03, Monash University.

Wood, D. R. (1999a). Multi-dimensional orthogonal graph drawing in the general position model. Technical Report 99/38, Monash University.

Wood, D. R. (1999b). A new algorithm and open problems in three-dimensional orthogonal graph drawing. In *Proceedings of the 10th Australasian Workshop on Combinatorical Algorithms (AWOCA'99)*, pages 157–167.

Wood, D. R. (2000). Three-Dimensional Orthogonal Graph Drawing. PhD thesis, Monash University.

Yannakakis, M. (1978). Node- and edge-deletion $\mathcal{NP}$-complete problems. In *Proceedings 10th Annual ACM Symposium on the Theory of Computing (STOC'78)*, pages 253–264.

Yoeli, P. (1972). The logic of automated map lettering. *The Cartographic Journal*, 9:99–108.

Zeller, A., and Lütkehaus, D. (1996). DDD — A free graphical front-end for UNIX debuggers. *ACM SIGPLAN Notices*, 31(1):22–27.

Zoraster, S. (1986). Integer programming applied to the map label placement problem. *Cartographica*, 23(3):16–27.

Zoraster, S. (1990). The solution of large 0-1 integer programming problems encountered in automated cartography. *Operations Research*, 38(5):752–759.