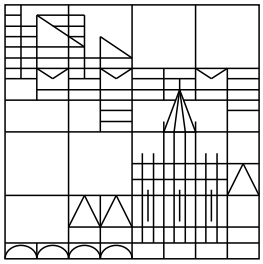
Universität Konstanz



Random Field Models for Graph Layout

Ulrik Brandes
Dorothea Wagner

Konstanzer Schriften in Mathematik und Informatik
Nr. 33, April 1997

ISSN 1430–3558

Universität Konstanz
Fach D 188, 78457 Konstanz, Germany
Email: preprints@informatik.uni–konstanz.de
WWW: http://www.informatik.uni–konstanz.de/Schriften

Random Field Models for Graph Layout

Ulrik Brandes Dorothea Wagner

33/1997
Fakultät für Mathematik und Informatik
Universität Konstanz

Abstract

We propose a new framework for the layout of graphs, which both unifies and generalizes many approaches known from the literature. Doing so, we briefly survey a number of models, particularly including force directed placement approaches. Our model is based on the stochastic concept of random fields, which have applications in various other areas such as physics, biology, economics, or medical imaging. By allowing, yet not enforcing, the decoupling of model and computation, it integrates algorithmic and declarative approaches to graph layout.

1 Introduction

Graphs are an abstraction that is central to computer science and many other disciplines. To communicate and explore graphs visually, drawings that obey domain specific aspects of expressiveness and readability are sought. Graph drawing, even for graphs of moderate size, is a tedious task that calls for automatic help. The growing literature on graph drawing is surveyed in [DETT94].

Generally speaking, existing approaches are divided into *declarative* and *algorithmic* ones [CT94]. The former generate layouts by solving a set of user defined constraints, whereas the latter aim to optimize some aesthetic criteria, specified by means of objective functions. Unlike many authors who start classifying graph drawing problems by the classes of graphs to be drawn, we take a more application governed view. A *graph drawing problem* is given by

- a prescribed *representation* (e.g. points and curves, inclusion diagrams, visibility representation),
- a prescribed *rendering* (e.g. colors, shapes, depth clues),
- a set of *constraints* (e.g. layers, relative positioning), and
- a set of *criteria* (e.g. layout area, number of bends).

Then, the solution to a graph drawing problem is to

1. compute a feasible *layout* (e.g. assign coordinates to visual elements), and to
2. *render* the final drawing.

Representation and constraints usually depend on the problem domain the graph originates from. They typically restrict the class of drawable graphs, which in turn governs the design of efficient layout algorithms. However, drawing problems shall not be classified by the classes of admissible graphs, for these are a consequence rather than a starting point. Rendering is a matter of ergonomy, psychology, computer graphics, and style. Our focus is on the formulation of layout criteria.

Unfortunately, many criteria commonly used to characterize readabilty of network diagrams are computationally intractable. In *force directed placement* approaches like [Ead84, KK89, DH96, Tun94, FLM95, SM95], this problem is tackeled by derivation of an aesthetic cost function from physical analogies like spring or particle systems. This function is used to measure the conformance of a given layout to prescribed criteria. A unifying formalism for those physical systems that inspired force directed placement, and for large systems with interaction in general, are random fields [Isi25, Gri76, Guy94]. We propose to use them as a unifying framework for graph layout models as well. Even though the origins of this formalism are the same as those of known force directed placement approaches, random fields are not at all limited to physical phenomena.

So-called Gibbs random fields are random vectors for which a neighborhood relation is defined on their index set. All dependencies among their component random variables are *local* with respect to these neighborhoods. This concept of local interaction has applications in such diverse areas as image processing, particle physics, agriculture, economics, etc., and constitutes a mature field of research. Embedding graph layout into the general framework of random fields provides access to a welth of theory. Many connections between seemingly different approaches to graph drawing become much clearer in this light. Moreover, the application of results from stochastic theory can be expected to yield substantial new contributions.

In Section 2 we adopt this formalism for graph layout. Section 3 contains a number of well known layout models stated in terms of random fields, and Section 4 discusses the formulation of isolated layout criteria. In a random field model, these can easily be combined and weighted. How to obtain layouts from a given model is discussed in Section 5. Section 6 concludes with a discussion on the benefits of this approach.

2 Random Field Models

In this section, a generic model for the layout step of a graph drawing problem is introduced. To keep the presentation simple, we restrict ourselves to straight-line representations throughout this paper. However, the most fundamental terminology is defined in its full generality as to indicate that our framework works for many other representations just as well.

Let $L = L(G) = \{l_1, \ldots, l_k\}$ be the set of *layout elements* of a graph G, corresponding to those visual elements that need to be assigned state values during the layout process. For each element l_i, there is a set $\mathcal{X}_{l_i}$ of possible *states*, which usually correspond to *locations* in some area or space. Then, every feasible *layout* is a vector $x = (x_{l_1}, \ldots, x_{l_k})$ in the set $\mathcal{X} = \mathcal{X}^L = \mathcal{X}_{l_1} \times \cdots \times \mathcal{X}_{l_k}$.

The most common visual representation of a graph is to place simple graphical objects corresponding to the vertices in a space of given dimension, and to map the edges to curves that connect the objects of adjacent vertices. Other representations include, e.g., proximity and inclusion drawings.

As outline above, we assume from now on that a graph $G = (V, E)$ has to be represented by a *straight-line* drawing. In such drawings, every edge is represented by a straight line segment. Since the layout of a straight-line drawing is completely determined by a placement of the vertices, the set of layout elements is simply the set of vertices, $L = V = \{v_1, \ldots, v_n\}$.

For practical purposes we may assume that each $\mathcal{X}_v$, $v \in V$, is a set of points in a finite two- or three-dimensional grid (the virtual display). If not stated otherwise, we let $\mathcal{X}_u = \mathcal{X}_v$ for all $u, v \in V$. Then, every vector $x = (x_v)_{v \in V} \in \mathcal{X} = \mathcal{X}^V$ is a feasible layout of G. For a layout $x \in \mathcal{X}$ and subset $C \subseteq V$, let $x_C = (x_v)_{v \in C}$ be the vector consisting only of those components of x that have indices in C. Given another layout $y \in \mathcal{X}$, we use x^{y_C} to denote the layout that equals x on $V \setminus C$, and y on C. $V - v$ is shorthand for $V \setminus \{v\}$. The respective quality of feasible layouts is measured by means of a random vector $X = (X_v)_{v \in V}$, or, more precisely, by its joint distribution over $\mathcal{X}$.

Definition 1 *A* random field model *(RFM) for the layout of a graph $G = (V, E)$ is a random vector $X = (X_v)_{v \in V}$ on some probability space $(\Omega, \mathcal{A}, P)$, for which component X_v, $v \in V$, takes values in the set of locations $\mathcal{X}_v$, and the induced probability measure P^X is strictly positive, i.e. $P(X = x) > 0$ for all $x \in \mathcal{X}$.*

In this formulation, a model should be set up such that good layouts have higher probability than bad layouts. The probabilty density is thus the objective function for layout quality. If a drawing is to display structural properties, the elements of the graph must not be placed independently of each other. We use a physical analogy to motivate a formalization of this requirement.

In particle systems, (spatially) neighboring elements mutually affect their states. We assume that their interactions are symmetric, such that the sum of interactions in every set of pairwise interacting elements is zero. Then, each of these sets has a potential, which together sum up for the inner energy of the system. If the system is in an equilibrium state, the probability of a certain configuration of particle states depends on this inner energy. Here, configurations with small energy are more likely than others. Such systems are often modelled by a certain class of random fields, which we now adopt for the purpose of graph layout.

For each vertex $v \in V$, the *neighborhood* $\eta_v \subseteq V - v$ of v is defined to be the set of vertices that can have an effect on the placement of v, and we require that $u \in \eta_v \Leftrightarrow v \in \eta_u$ for all $u, v \in V$. Consequently, the *neighborhood system* $\eta = \bigcup_{v \in V} \eta_v$ induces an undirected *interaction graph* G^η. Let $\mathcal{C} = \mathcal{C}(\eta)$ be the set of cliques in G^η. For each $C \in \mathcal{C}$, the *clique potential* is a function $U_C : \mathcal{X} \to \mathbb{R}$ obeying

$$x_C = y_C \quad \Rightarrow \quad U_C(x) = U_C(y)$$

for all $x, y \in \mathcal{X}$. In other words, the potential of a certain configuration $x \in \mathcal{X}$ with respect to a given clique C in the interaction graph is completely determined by the clique's local configuration x_C. In particle systems, potentials of one-element cliques model external forces acting on the particles, whereas potentials of larger cliques measure the systems grade of relaxation. The family $U_\mathcal{C} = \{U_C \mid C \in \mathcal{C}\}$ of all clique potentials is simply called the *potential*. The *energy* $U(x)$ of a configuration $x \in \mathcal{X}$ is defined to be the sum over all clique potentials,

$$U(x) = \sum_{C \in \mathcal{C}} U_C(x).$$

Definition 2 *A random field model X for the layout of a graph $G = (V, E)$ is called* Gibbs random field model *(GRFM) with respect to neighborhood system η and potential $U_{\mathcal{C}}$, if*

$$P(X = x) = \frac{1}{Z} e^{-U(x)} \tag{1}$$

for all $x \in \mathcal{X}$, where $Z = \sum_{y \in \mathcal{X}} e^{-U(y)}$ is a normalizing constant.

According to Definition 2, configurations with low energy are more likely than others in a GRFM. The specific form of distribution (1) is motivated by results from thermodynamics and statistical mechanics, where it is usually called *Gibbs*, *Boltzmann*, or *canonical distribution*. Z is called *partition function* or *Zustandssumme*. Observe that the energy function U of a GRFM X has two interesting properties: it is unique up to an additive constant, and its minima do not change when it is multiplied by a constant.

Clearly, every random field can be represented by a Gibbs random field, if the interaction graph and the clique potentials are defined appropriately. We therefore do not distinguish between general and Gibbs random fields. Any time we speak of random fields, clique potentials are in the back of our mind.

We show that, in terms of graph layout, clique potentials provide a powerful mechanism to specify layout criteria. On the other hand, layout constraints can to some extent be formulated by introducing additional layout elements (e.g. non-graph graphics, invisible linkages), and by restricting the set of admissible locations (e.g. forbidden areas, layered graphs). A unified way to obtain a good layout consists in first modelling its desired properties by means of a random field (i.e. clique potentials), and secondly minimizing the energy function. Minimizing the energy corresponds to computing a layout of maximum probability. The configurations of maximum probability are also called the *modes* of the respective distribution.

3 Familiar Examples from a New Perspective

Computing nice layouts is a difficult task, since, at least for general graphs, most of the commonly used aesthetic criteria are computationally intractable. Force directed placement refers to a popular class of layout approaches that have been devised to cope with these obstacles. Just like random fields, they are build upon analogies to physical systems, yet different, more specialized formalisms have been derived. Based on models of interaction among elements of a graph, these approaches produce remarkably good layouts. To some degree, they seem to meet a number of common criteria simultaneously (e.g. uniform edge length, symmetry, and clustering).

The variants of the force directed placement approach fit perfectly into the framework of random fields. In fact, with suitably chosen sets of parameters, all of them can be regarded instances of random field models. A straightforward example is the famous spring embedder.

Spring embedder [Ead84]. The basic idea is to simulate a physical system of rings (the vertices) which are either repelling or joined by a logarithmic spring (the edges). Then, an equilibrium state of the system is sought. Such a system can be described by means of a RFM X with neighborhood $\eta_v = V - v$ for all $v \in V$ (i.e., G^η is the complete graph on vertex set

V), and non-zero clique potentials only for two-element cliques of G^η:

$$U_{\{u,v\}} = \begin{cases} |c_1 \cdot \log \frac{d(x_u,x_v)}{c_2}| & \text{if } \{u,v\} \in E \\ \frac{c_3}{\sqrt{d(x_u,x_v)}} & \text{otherwise.} \end{cases}$$

Here, c_1, c_2, and c_3 are constants, and $d(x_u, x_v)$ is the Euclidean distance of locations x_u und x_v. The spring embedder aims to minimize the force on each vertex, which is equivalent to minimizing the energy U in the RFM.[1]

Variants of the spring embedder. Due to the often very pleasing layouts produced by the basic spring embedder, a number of refinements have been designed to overcome some of its limitations. We treat them jointly and indicate their reformulation in terms of clique potentials.

Kamada and Kawai [KK89] introduce a variant of the spring embedder which joins each vertex with every other vertex by a linear spring. The spring's strength depends on the graph theoretical distance of its end vertices. The objective function is set up to minimize a weighted version of the squared error of desired edge lengths. Just like the basic spring embedder, it is equivalent to a RFM with complete interaction, but modified potentials

$$U_{\{u,v\}}(x) = \frac{c_1}{d_G(u,v)^2} \cdot \left(d(x_u, x_v) - c_2 \cdot \frac{d_G(u,v)}{diam(G)} \right)^2,$$

where $d_G(u,v)$ is the distance of u and v in G, i.e. the length of the shortest path connecting u and v. The largest distance of any two vertices is called the diameter of G and denoted by $diam(G)$.

Fruchterman and Reingold [FR91] simulate a system of interacting particles with repelling forces $\rho_{u,v}(x) = c_1^2/d(x_u, x_v)$ between every pair of vertices $u, v \in V$, and additional attracting forces $\alpha_{u,v}(x) = d(x_u, x_v)^2/c_1$ between adjacent vertices. The parameter c_1 is set to $c_2 \cdot \sqrt{A/n}$, where A is the desired layout area, and c_2 is an experimentally chosen constant. Again, only the two-element clique potentials of the spring embedder RFM need to be modified:

$$U_{\{u,v\}}(x) = \begin{cases} |\alpha_{u,v}(x) - \rho_{u,v}(x)| & \text{if } \{u,v\} \in E \\ \rho_{u,v}(x) & \text{otherwise.} \end{cases}$$

A number of additional forces are introduced by Davidson and Harel [DH96]. They use a complex energy function, which can be decomposed into clique potentials of the form

$$U_{\{v\}}(x) = c_1 \cdot \left(\frac{1}{l(x_v)^2} + \frac{1}{r(x_v)^2} + \frac{1}{t(x_v)^2} + \frac{1}{b(x_v)^2} \right)$$

and

$$U_{\{u,v\}}(x) = \begin{cases} \frac{c_2}{d(x_u,x_v)^2} + c_3 \cdot d(x_u, x_v)^2 & \text{if } \{u,v\} \in E \\ \frac{c_2}{d(x_u,x_v)^2} & \text{otherwise} \end{cases}$$

to control the spreading of vertices, distance of vertices, and length of edges. Here, $l(x_v)$, $r(x_v)$, $t(x_v)$, and $b(x_v)$ denote the distance of location x_v from the left, right, top and bottom

[1] More precisely, the algorithm is designed to find a stable configuration corresponding to a local minimum. See Section 5 for further details.

border of the display area, respectively. Additionally, the energy function of [DH96] uses two kinds of terms in order to prevent edges from passing by too close to vertices, and to reduce edge crossings: Let $d_{u,\{v,w\}}(x)$ be the distance of x_u from the line connecting x_v and x_w, if $\{v,w\} \in E$, and zero otherwise. Thus, the two remaining contributions to the energy function, expressed in terms of clique potentials, are

$$U_{\{u,v,w\}}(x) = \frac{c_4}{\min\{\delta, d_{u,\{v,w\}}(x)\}^2} + \frac{c_4}{\min\{\delta, d_{v,\{u,w\}}(x)\}^2} + \frac{c_4}{\min\{\delta, d_{w,\{u,v\}}(x)\}^2}$$

and

$$U_{\{v_1,v_2,v_3,v_4\}}(x) = c_5 \cdot \left(\chi_{\{v_1,v_2\},\{v_3,v_4\}}(x) + \chi_{\{v_1,v_3\},\{v_2,v_4\}}(x) + \chi_{\{v_1,v_4\},\{v_2,v_3\}}(x)\right) \qquad (2)$$

where $\chi_{\{u_1,v_1\},\{u_2,v_2\}}(x)$ equals 1, if $\{u_1,v_1\},\{u_2,v_2\} \in E$ and the lines connecting x_{u_1} with x_{v_1} and x_{u_2} with x_{v_2} cross, and otherwise it is set to 0. Constant δ is a threshold value for the minimum allowed distance between edges and vertices. Constants c_i, $i = 1,\ldots,5$, are used to control the relative influence of each contribution to the energy function.

Tunkelang proposed yet another layout strategy [Tun94] that incorporates an aesthetic cost function composed of attractive and repulsive forces, as well as the number of edge crossing. By adjusting the respective constants, it is hence easy to define a RFM with an energy function that is identical to this cost function.[2]

The GEM algorithm of Frick *et al.* [FLM95] basically uses attractive and repulsive forces similar to those discussed above. Omitting computational issues, it involves one distinct modelling feature. Gravitational forces drag vertices to the barycenter of the layout, with their strength depending on the degree of the vertex. This corresponds to a non-zero potential

$$U_V(x) = c_1 \cdot \sum_{v \in V} \phi(d_G(v)) \left(\zeta(x) - x_v\right)$$

of the clique V containing all vertices of the graph. Here, ϕ is some function of the vertex degree, and $\zeta(x)$ denotes the barycenter of all locations in x.

While most force directed placement approaches are suitable for undirected graphs only, the variant of Sugiyama and Misue [SM95] allows (even mixed in) directed edges.[3] In addition to the springs and repulsive forces of the basic spring embedder, magnetic fields introduce new forces

$$\mu_{\{u,v\}}(x) = c_3 \cdot b \cdot d(x_u,x_v)^{c_4} \cdot \theta^{c_5}_{\{u,v\}}(x)$$

onto the springs (i.e. the edges). According to possible combinations of edge direction, the springs are partitioned into so-called non-magnetic, uni-directional magnetic, and bi-directional magnetic springs. Here, b denotes the strength of the magnetic field, $\theta_{\{u,v\}}(x)$ the angle (radian) between the field's orientation and the edge vector (if the edge is bi-directional, it is the minimum of both angles). c_3, c_4, and c_5 are constants again. For adjacent vertices $u,v \in V$, $\mu_{\{u,v\}}(x)$ is simply an additive term in $U_{\{u,v\}}$ of the basic spring embedder. In case $\{u,v\} \notin E$, it is neglected.

Because of their common analogies, it is not surprising that force directed placement is captured by random field models. The following two examples are less obvious.

[2]Note that the more important differences of Tunkelang's approach compared to [DH96] and [FR91] are in the computation, not in the model.

[3]Tunkelang essentially has the same idea in Section 4.1.3 of [Tun94], but does not elaborate on this.

Circle layout. A traditional way of drawing sociograms (graphs that model social structures) is to place the vertices on an imaginary circle, and to draw the edges as chords inside the circle. A common optimization criterion is to minimize the total edge length, which causes dense subgraphs to be placed within small arcs. To restate the model in terms of a random field, we first choose the interaction graph to be complete. Next, for each $v \in V$, let $\mathcal{X}_v$ be the set of positions allowable on the perimeter of the imaginary circle. Clique potentials are non-zero only for cliques of size two:

$$U_{\{u,v\}}(x) = \begin{cases} M & \text{if } x_u = x_v \\ d(x_u, x_v) & \text{if } x_u \neq x_v \text{ and } \{u,v\} \in E \\ 0 & \text{otherwise} \end{cases}$$

where M is a sufficiently large constant to ensure vertices are placed at different locations. A layout of minimum energy assigns a different location to each vertex and minimizes the sum of the pairwise distances (i.e. the total edge length).

Hierarchy layout. Sugiyama *et al.* [STT81] proposed an algorithm for the layout of hierarchical (directed) graphs. Vertices are placed on horizontal layers such that, in general, edges are directed from upper to lower layers. It is assumed that adjacent vertices are assigned to different layers. After assigning a layer to each vertex (which we consider a constraint and therefore not part of the layout step) directions of edges can be neglected.

Now, choose $\mathcal{X}_v$ to consist of admissible vertex positions on a horizontal line corresponding to the layer of vertex $v \in V$. Vertices $u, v \in V$ are adjacent in the interaction graph G^η, if at least one of the following conditions hold:

- $\{u,v\} \in E$, i.e. they are adjacent in the original graph G,
- they are on the same layer,
- they are on consecutive layers, or
- there is a vertex $w \in V$, such that $\{u,w\} \in E$ and v is on a layer between those of u and w.

Clique potentials are defined for

- pairs of vertices on the same layer (avoid overlap),
- pairs of vertices that are adjacent in G (keep edge length short),
- triples of vertices of which two are adjacent and the other is placed on a layer inbetween (avoid edges going through vertices), and
- pairs of adjacent vertex pairs that span overlapping layer intervals (avoid crossings).

The modelling features needed for this are covered in the following section.

Even though we have treated only some of the most popular layout models, it can be seen from the following section that others can be stated in terms of random fields as well. Note that many constraints can be formulated equivalently in terms of criteria. With respect to the model, there is no difference for a planar graph between a planar drawing, and a drawing with the minimum number of crossings.

4 Specifying Layout Criteria

The ease of formulating specific layout criteria and their relative importance in a random field model is discussed in this section. The uniformity of using clique potentials to express layout features facilitates both design and understanding of models, while their expressive power is obvious from the set of examples we give.

Depending on the problem domain, a good diagram is supposed to convey information that relates to the structural properties of the graph and to its interpretation, respectively. Properties that are expressed solely in the adjacency relation are called the *syntactic* properties of the graph. Every other kind of information associated with the graph is part of its *semantics*.

As far as the layout is concerned, it is reasonable to assume that rules for the appropriate visualization of such properties can be expressed in terms of (mostly fixed-size) subsets of layout elements. This is because, typically, a finite description of layout goals has to be applicable to an infinite number of admissible graphs. Hence, conformance to layout rules should be measurable separately for each of these sets, or, in other words, by clique potentials.

We give a number of examples to provide some evidence for this claim. These examples constitute a sample of important layout criteria used in various graph drawing applications. Even though the criteria are treated separately, they can obviously be combined into a single energy function. Moreover, they can be normalized and weighted in order to express differences in importance.

For each criterion and its associated clique potentials, we implicitly assume that the respective cliques are present in the interaction graph. This is easily guaranteed by choosing G^η to be the complete graph on V. For reasonable comparison of different models and their algorithmic implications, it is beneficial, though, to identify an interaction graph with the least number of edges that still contains each of the cliques that are assigned non-zero potentials.

Vertex location. In some contexts, attributes of vertices cause certain locations to be preferred over others. Consider, for example, a social network. Centrality is an important measure for network analysts, indicating some aspect of an actor's importance in the network. One reasonable idea might be to place vertices corresponding to central actors closer to the center of the drawing. A clique potential measuring the graphical centrality of a vertex v with social centrality $c(v)$ is

$$U_{\{v\}}(x) = \left(d(x_v, \zeta) - c(v)\right)^2$$

where ζ is the center of the drawing. Any other measure of deviation from one or more favourable locations is conceiveable.

On the other hand, there are situations where one wants to fix the location of some special vertices. One way to model this is to simply restrict the corresponding sets of admissible locations. A different formulation is to exclude these vertices from the vertex set and to include their contribution to the energy function into the potentials of the smaller cliques that remain.

Vertex distribution and borderlines. The distribution of vertices in the display area (or space) can to some extend be controlled by using functions of locations as potentials for

all single element cliques. For instance, potentials

$$U_{\{v\}}(x) = \frac{(x_v - \zeta)^2}{2\sigma^2}$$

result in a random field with independent components distributed according to a Gaussian distribution with its mean in the center ζ of the drawing. More complex distribution patterns can be modelled by multivariate distributions. Since optimal layouts place every vertex at a location of maximum probability, these potentials are useful only in addition to other criteria. Note the correspondence to preferred locations for single vertices.

To spread the vertices evenly on a bounded area, two-element clique potentials like in [DH96] (cf. to the previous section) suffice. By adjusting pairwise repulsion according to syntactic or semantic properties, more elaborate distributions can be realized (see also the following criterion).

Vertex distance. In general, vertex images should not overlap. Or, in other words, vertices should not be placed too close to each other. Whatever the notion of distance in a particular problem domain might be, it depends only on two vertices and their assigned locations. Typically, reciprocal quadratic functions of the Euclidean distance of locations are used to penalize small vertex distances:[4]

$$U_{\{u,v\}} = \frac{1}{d(x_u, x_v)^2} \tag{3}$$

But there might also be preferred distances between vertices depending on domain specific semantics of the graph. Interpreting these as virtual bonds between the vertices reduces the problem to the treatment of edge length.

Edge length. Even though vertices should not be placed too close to each other, it is often desireable to keep edge lengths short, or even uniform. We have outlined in the previous section, how these layout goals are achieved in the spring embedder model and its variants, and thus in corresponding random field models. More generally, any distance measuring function is suitable, if it is monotone decreasing in the grade of conformance to the desired relative positioning. Note that it need not be a metric or fulfill any other such properties.

If, for instance, length $d_{u,v}$ is desired for edge $\{u, v\}$,

$$U_{\{u,v\}} = \left(d(x_u, x_v) - d_{u,v}\right)^2$$

measures the deviation of the actual edge length from the desired one (see, e.g., [KK89]). Minimizing the energy then minimizes the square error in the edge lengths.

Edge direction. In upward drawings, edges are directed (either because they are so in the input graph, or because the vertices have been numbered such that an edge is directed from the vertex with a smaller number to the one with a larger number) and shall be drawn pointing upwards. In this case, it is sufficient to assign potentials to pairs of adjacent vertices that take into account the orientation of the difference vector. The most basic example for

[4] Distance measures might need adjustment to account for the size of graphical objects used to render the vertices.

a pair of adjacent vertices u, v joined by an edge directed from u to v in a two-dimensional upward drawing is

$$U_{\{u,v\}}(x) = |\sphericalangle_y(\overline{x_u x_v})|$$

where $\sphericalangle_y(\overline{x_u x_v})$ denotes the angle between the y-axis and the vector from x_u to x_v. The magnetic spring embedder [SM95] uses rotative forces induced by magnetic fields of various shapes.

Bend number. A RFM can represent edge bends by subdividing edges with artificial vertices. The number of possible bends of an edge is bounded by the number of additional vertices introduced. For any three vertices u, v, w inducing a path with an artifical vertex in the middle, potentials like

$$U_{\{u,v,w\}} = \begin{cases} 0 & \text{if } u, v, w \text{ lie on a straight line} \\ M & \text{otherwise} \end{cases}$$

count the number of bends. Instead of simply counting the number of bends, their respective angles might be measured in the same way as is outlined below for pairs of edges.

Angle between incident edges. The angular resolution of a straight line embedding is the smallest angle formed by any two incident edges. Small angles have a thickening effect on the perception of these edges, because their lines seem to merge. It is thus desirable to avoid small angles. In a straight-line representation angles between incident edges are easily determined from the locations of the three vertices involved. E.g., in the two-dimensional plane we have

$$U_{\{u,v,w\}} = \phi\left(\sphericalangle_{\{u,v\},\{v,w\}}(x)\right) + \phi\left(\sphericalangle_{\{v,w\},\{w,u\}}(x)\right) + \phi\left(\sphericalangle_{\{w,u\},\{u,v\}}(x)\right)$$

where $\sphericalangle_{\{v_1,v_2\},\{v_2,v_3\}}(x)$ is the angle between the lines connecting x_{v_1} with x_{v_2} and x_{v_2} with x_{v_3} if $\{v_1, v_2\}, \{v_2, v_3\} \in E$, and 0 otherwise. Again, clique potentials may incorporate any function ϕ of the angle.

Vertex to edge distance. In general, vertices should not be intersected or passed closely by edges they are not incident with. Using the distance penalty of (3), we get

$$U_{\{u,v,w\}} = \frac{1}{d_{u,\{v,w\}}(x)^2} + \frac{1}{d_{u,\{v,w\}}(x)^2} + \frac{1}{d_{u,\{v,w\}}(x)^2},$$

where $d_{u,\{v,w\}}(x)$ is the perpendicular distance of location x_w from the line connecting locations x_u and x_v if $\{v, w\} \in E$, and 0 otherwise.

Edge crossing. Clique potentials can be used to count the number of edge crossings in a drawing (e.g., [Tun94, FLM95]. Since every edge crossing is determined by exactly four vertices, it is sufficient to sum up the number of crossings in each induced subgraph of size four (see (2)).

Angle of edge crossing. Instead of simply counting the number of crossings, clique potentials penalizing crossings with some function of the crossing angle allow for a finer scale of layout ranking. Simply substitute $\chi_{\{u_1,v_1\},\{u_2,v_2\}}$ in (2) by a function

$$\hat{\chi}_{\{u_1,v_1\},\{u_2,v_2\}}(x) = \chi_{\{u_1,v_1\},\{u_2,v_2\}}(x) \cdot \phi\left(\sphericalangle_{\{u_1,v_1\},\{u_2,v_2\}}(x)\right).$$

Face or subgraph shape. Given a planar graph and a fixed combinatorial embedding,[5] it might be desired to find a layout such that all but one face are convex. Assume that such drawing is possible and that the vertices of the outer face are already positioned to form a convex polygon. For graphs allowing such layouts (a certain superclass of all 3-connected planar graphs), the barycentric mapping of Tutte [Tut63] generates a planar layout with convex faces by solving

$$x_v = \frac{1}{d_G(v)} \sum_{\{v,w\} \in E} x_w$$

for every vertex $v \in V$ that does not belong to the outer face.[6] Without going into detail, the following example shows how to formulate convexity and related criteria in terms of clique potentials.

In [TX95], Tollis and Xia layout certain telecommunications networks by placing subsets of the vertices on circles. These sets are given by a ring cover of the graph, i.e. a set of connected cycles such that every vertex is contained in at least one cycle. Let u, v, w be a triple of consecutive vertices on a cycle of length k. Potentials

$$U_{\{u,v,w\}} = \left(\sphericalangle_{\{u,v\},\{v,w\}}(x) - \frac{360}{k} \right)^2$$

measure the squared differences of a ring's internal angles from those of a regular polygon. Note that this model implicitly formulates a compromise for the shape of rings that share more than one vertex (a difficulty noted in [TX95]). For a more elaborate circular layout model, see [DMM97].

In Fig. 1, we have summarized example formalizations for some criteria of general importance. Usually, a good drawing is to meet a number of these. Let $U^{(i)}$, $i = 1, \ldots, k$, be energy functions modelling certain layout features, and let $\mathcal{C}$ be the set of all cliques that are assigned a non-zero potential in any of these. The criteria are easily combined into a single energy function

$$U(x) = \sum_{C \in \mathcal{C}} \left(\sum_{i=1}^{k} f_C^{(i)} \left(U_C^{(i)}(x) \right) \right),$$

where functions $f_C^{(i)} : \mathbb{R} \to \mathbb{R}$, $i = 1, \ldots, k$, $C \in \mathcal{C}$, are applied to normalize and weight the contribution of the input energies. Normalization of contributions is used, for instance, in the implementation of Davidson and Harel's algorithm [DH96] in GraphEd [BHR96].

5 Algorithmic Aspects

It was noted at the end of Section 2 that an optimal layout of a random field model is obtained by computing a mode of the joint distribution. Since random fields are easily seen to be generalizations of Boltzmann machines (see [AK89] for an introduction to Boltzmann

[5] In a combinatorial embedding, only the cyclic order of edges around common incident vertices is fixed, not the vertices' locations.

[6] Interestingly enough, with the outer face fixed, this method is a polynomial optimization algorithm for a simple variant of the spring embedder.

criterion	example potential	clique size
vertex position	$U_{\{v\}}(x) = \left(d(x_v, \zeta) - c(v)\right)^2$	1
vertex distribution	$U_{\{v\}}(x) = \frac{d(x_v-\zeta)^2}{2\sigma^2}$	1
vertex distance	$U_{\{u,v\}}(x) = \frac{1}{d(x_u,x_v)^2}$	2
edge length	$U_{\{u,v\}}(x) = \left(d(x_u, x_v) - d_{u,v}\right)^2$	2
edge orientation	$U_{\{u,v\}}(x) = \lvert\sphericalangle_y(\overline{x_u x_v})\rvert$	2
bend number	$U_{\{u,v,w\}}(x) = \begin{cases} 0 & \text{if } u,v,w \text{ lie on a straight line} \\ M & \text{otherwise} \end{cases}$	3
vertex to edge distance	$U_{\{u,v,w\}} = \frac{1}{d_{u,\{v,w\}}(x)^2}$	3
angular resolution	$U_{\{u,v,w\}}(x) = \phi\left(\sphericalangle_{\{u,v\},\{v,w\}}(x)\right) + \phi\left(\sphericalangle_{\{v,w\},\{w,u\}}(x)\right) + \phi\left(\sphericalangle_{\{w,u\},\{u,v\}}(x)\right)$	3
crossings / their angles	$U_{\{v_1,v_2,v_3,v_4\}}(x) = \hat{\chi}_{\{v_1,v_2\},\{v_3,v_4\}}(x) + \hat{\chi}_{\{v_1,v_3\},\{v_2,v_4\}}(x) + \hat{\chi}_{\{v_1,v_4\},\{v_2,v_3\}}(x)$	4

Figure 1: Selected layout criteria and example formulations in terms of clique potentials. The last column states the size of the cliques, which in general corresponds to a criterion's computational cost (see Section 5).

machines), energy minimization is an $\mathcal{NP}$-hard problem in general. To make this paper self-contained, we outline a simple reduction proving the $\mathcal{NP}$-hardness of energy minimization even for the most simple random field models.

Theorem 1 *Energy minimization for random field models is $\mathcal{NP}$-hard, even if $G^\eta = G$ and only two-element cliques have non-zero potentials.*

Proof: We show that the corresponding decision version is $\mathcal{NP}$-complete. An arbitrary instance (ϕ, k) of MAX 2SAT (Problem [LO5] of [GJ91]) is transformed into a RFM X, such that there is a truth assignment satisfying at least k clauses in the boolean formula $\phi = (x_{1,1} \wedge x_{1,2}) \vee \ldots \vee (x_{m,1} \wedge x_{m,2})$, if and only if there is a layout $x \in \mathcal{X}$ with energy at most $-k$.

We are given a formula ϕ in conjunctive normal form with two literals in each clause, and a positive integer k. It is transformed into a graph $G = (V, E)$, where V equals the set of variables of ϕ. The edges in E are exactly the pairs of variables that have literals in a common clause. Now, let the interaction graph G^η equal G, in other words, only adjacent vertices interact. Let $\mathcal{X}_v = \{0, 1\}$ for $v \in V$, and define non-zero clique potentials except only for the two-element cliques $\{u, v\} \subseteq E$. For these, let $U_{\{u,v\}}(x) = -\kappa(x_u, x_v)$, where $\kappa(x_u, x_v)$ is the number of clauses containing literals of both u and v that is satisfied by the assignment of x_u to u and x_v to v. (Note that $\kappa(x_u, x_v)$ is easily computed in constant time). Clearly, this transformation is polynomial. □

Thus, efficient algorithms for finding a layout of minimum energy cannot be expected for arbitrary models. In graph drawing, though, *optimal* solutions are seldomly needed. In turn, algorithms computing at least near optimal solutions for many different energy functions provide rich variational capabilities. A single layout algorithm can be used to evaluate an abundance of models. Not suprisingly, such algorithms have attracted much attention in the literature on applications of stochastic models, and their performance and limitations are understood quite well. We outline three closely related optimization procedures, and it turns out that not only the models, but also the algorithms used in force directed placement are special cases of well known approaches from the theory of random fields.

Consider a random field X for the layout of $G = (V, E)$. One verifies that, for every vertex $v \in V$ and every layout $x \in \mathcal{X}$, the following equations are satisfied by the conditional probabilities

$$\begin{aligned} P\left(X_v = x_v |\ X_{V-v} = x_{V-v}\right) &= \frac{1}{Z} \exp\left\{-\sum_{C \in \mathcal{C}} U_C(x)\right\} \\ &= \frac{1}{Z_v} \exp\left\{-\sum_{C \in \mathcal{C}: v \in C} U_C(x)\right\} \\ &= P\left(X_v = x_v |\ X_{\eta_v} = x_{\eta_v}\right) \end{aligned} \tag{4}$$

where $Z_v = \sum_{y_v \in \mathcal{X}_v} \exp\left\{-\sum_{C \in \mathcal{C}: v \in C} U_C(x^{y_v})\right\}$ is a normalizing constant, again. The probabilities in (4) are called the *local characteristics* of X.[7] They describe the probability that vertex v is placed at location x_v, when the location of every other vertex $v' \in V - v$ is fixed to be $x_{v'}$. The equations state that these probabilities depend only on the neighbors of v. A local minimum of the energy function can be obtained by iteratively maximizing (4) with respect to x_v for every $v \in V$. In the context of image processing, this procedure was introduced by Besag, who termed it *Iterated Conditional Modes (ICM)* [Bes86]. ICM is known to display rapid convergence to a local energy minimum. In graph drawing, many variants of the spring embedder basically use the same strategy (e.g. [Ead84, KK89, SM95]). In most cases, even the maximum of (4) can only be approximated. For example, the basic spring embedder of [Ead84] multiplies the gradient with a constant factor, whereas the variant of [KK89] uses the Newton-Raphson method.

A common way of avoiding local minima is the use of annealing-type algorithms. Here, the random field is parameterized by a real value $T > 0$, called *temperature*.[8] The joint distribution then becomes

$$P(X^{(T)} = x) = \frac{1}{Z(T)}\, e^{-U(x)/T}$$

where $Z(T) = \sum_{y \in \mathcal{X}} \exp\{-U(y)/T\}$. The temperature parameter is used to control the effect of the energy function on the joint distribution. For higher values of T, $P^{X^{(T)}}$ is close to a uniform distribution over all layouts, while it peaks more sharply for smaller values of

[7] If $P\left(X_v = x_v |\ X_{V-v} = x_{V-v}\right) = P\left(X_v = x_v |\ X_{\eta_v} = x_{\eta_v}\right)$ holds for all $v \in V$, the random field X is said to satisfy the *Markovian property*. If a strictly positive random field is defined in terms of consistent local characteristics, it is called *Markov random field*. An important result from stochastic theory states that Markov and Gibbs random fields are equivalent with respect to the same interaction graph (a detailed proof is contained in, e.g., [Gri76]).

[8] The adaptive parameter of Kamada and Kawai's gradient search approach [KK89] is also called temperature, but of limited correspondence.

T. Since it can be shown that $\lim_{T\downarrow 0} P^{X^{(T)}}$ is the uniform distribution over all layouts of minimum energy, annealing-type algorithms approximate $P^{X^{(T)}}$, and decrease T over time. In a temperatured random field $X^{(T)}$, the local characteristics become

$$P\left(X_v^{(T)} = x_v \,\middle|\, X_{\eta_v}^{(T)} = x_{\eta_v}\right) = \frac{1}{Z_v(T)} \exp\left\{-\sum_{C\in\mathcal{C}:v\in C} U_C(x)/T\right\}$$

where $Z_v(T) = \sum_{y_v\in\mathcal{X}_v} \exp\{-U(x^{y_v})/T\}$. The so-called *Gibbs sampler* [GG84] approximates $P^{X^{(T)}}$ by repeatedly sampling from these temperatured local characteristics. Since it assumes an underlying (Gibbs) random field, the method is inherent to this model, and not yet used in graph layout algorithms. However, the closely related Simulated Annealing framework is used for a variety of combinatorial optimization problems, including graph layout.

Simulated Annealing is a general optimization method for large-scale combinatorial problems, first introduced in Metropolis *et. al.* [MRR+53] and Kirkpatrick *et. al.* [KGV83]. For a rigorous treatment, see [AK89, vLA88]. Configurations of a large solution space are weighted by an energy function, and a configuration of minimum energy is sought. Roughly speaking, the algorithm iteratively modifies a candidate solution. In each iteration, the modified configuration is either accepted to be the new candidate solution, or it is rejected. It is accepted, if its energy is smaller than the energy of the current candidate. In order to avoid local minima, higher energy configurations are also accepted with probability $e^{-\Delta/T}$, where Δ is the energy difference and $T > 0$ the temperature parameter of the annealing. Convergence is enforced by slowly lowering T according to some cooling schedule.

Now, consider two layouts x and x^{y_v} that differ only in the location assigned to vertex v. We have

$$\begin{aligned}\frac{P\left(X_v^{(T)} = y_v \,\middle|\, X_{\eta_v}^{(T)} = x_{\eta_v}\right)}{P\left(X_v^{(T)} = x_v \,\middle|\, X_{\eta_v}^{(T)} = x_{\eta_v}\right)} &= \frac{Z_v^{-1}(T)\, e^{-\sum_{C\in\mathcal{C}:v\in C} U_C(x^{y_v})/T}}{Z_v^{-1}(T)\, e^{-\sum_{C\in\mathcal{C}:v\in C} U_C(x)/T}} \\ &= e^{-\sum_{C\in\mathcal{C}:v\in C} (U_C(x^{y_v})-U_C(x))/T}.\end{aligned}$$

Clearly,

$$\min\left\{1\,,\, \exp\left\{-\sum_{C\in\mathcal{C}:v\in C} \left(U_C(x^{y_v}) - U_C(x)\right)/T\right\}\right\}$$

is identical to the acceptance probability of Simulated Annealing, provided that a new candidate solution is generated by appropriate modification of the location of a single vertex v. Here, the role of the interaction graph G^η becomes evident. During the algorithm, the energy $U(x)$ need never be computed entirely. For each iteration, the transition probability depends on the change alone. Computing energy differences on cliques of G^η with non-zero potential that contain the modified vertex is thus sufficient for an implementation. Since the above procedure differs from the Gibbs sampler only in the sampling strategy used to approximate $P^{X^{(T)}}$, it is often termed *Metropolis sampler*. A comparative study of both algorithms is [CS93].

Simulated Annealing is used for graph layout, e.g., in [DH96, HS95, CT96], even though it is often disregarded because of its computational cost [BHR96]. On the other hand, local optimization algorithms [Ead84, KK89, FR91, FLM95, SM95] are attributed to be relatively

fast, while producing similar results. This comes as no surprise, since (given the discussion above) it is easy to see that vertexwise local optimization corresponds directly to the limit case $T = 0$ of the annealing algorithms. In other words, these algorithms are annealing algorithms which use a cooling schedule that immediately freezes the system. Finally notice that quite a number of graph layout algorithms are implemented in a batch-like fashion (e.g. [Ead84, KK89, SM95]). A new location in a layout with all but this vertex fixed is computed independently for each vertex. Afterwards, all vertices are updated simultaneously. This corresponds directly to a parallel algorithm with synchronous, independent updates of all vertices. In general, these algorithms converge to a different limit distribution, and oszillation effects may occur (see [Aze90], or [Win95] for a more general treatment).

6 Discussion

We have shown how to model graph layout by means of random fields. In a random field model, the probability assigned to a given layout corresponds to its conformance to prescribed layout criteria. Many known layout models can be treated uniformly within our framework. This uniformity greatly simplifies comparison, analysis, combination and extension of existing layout models and their building blocks. Furthermore, the general framework facilitates deeper understanding of relations between different models and algorithms. A few other important aspects are discussed in the following.

Tunkelang states that his algorithm is applicable to any cost function that is a function of node and edge locations only, and that this class of functions provides high flexibility [Tun94]. The objective function defined in [MP91] is composed of terms for locations of singletons and pairs of layout elements. Clearly, both classes are contained in the much larger class of energy functions for random fields. Moreover, the smooth embedding into a stochastic framework opens up a whole toolbox of known methods and relationships that await exploitation. The theory of random fields is a mature field of research that may provide considerable contribution to graph drawing. For example, we are currently adopting ideas from the use of random fields in image restoration for dynamic graph layout.

Clique potentials form a set based specification language for layout features. Hence, random field models can be seen as an integration of algorithmic and declarative approaches to graph drawing. It is a common complaint that constraint based layout algorithms perform too slow in practice, while algorithmic approaches usually offer few options for parameterization. Due to their flexibility, clique potentials might be useful to realize customizable layout approaches like those in [LE95, WM96] in a much more general setting, while preserving the option to later design algorithms tuned to particular energy functions that proved valuable (see also the notes on the experimental framework used in [FR91]). Furthermore, the user of a graph drawing system need not be confronted with the specification of clique potentials directly. Rather, a set of predefined, parameterized layout criteria might be offered by presenting, in a graphical user interface or by means of a specification language, their interpretation only.

The integration of constraints in random field models appears to be a serious problem. Since the joint distribution of a random field is strictly positive, every configuration of admissible locations is feasible. Certain types of constraints can be incorporated directly into an RFM by restricting admissible locations for layout elements. In particular, this includes the fixation of selected vertices and edges to certain locations. A common way to further

restrict the set of possible outcomes in annealing-type algorithms is the use of penalty functions. Here, forbidden configurations are penalized by a factor monotonous in T^{-1}. Since random variables form a powerful analytical tool, it might even be reasonable to drop layout constraints during a formal treatment that changes, analyzes, or combines models, and re-impose them afterwards.

The clean separation of model and computation provides the option to implement a single layout engine for arbitrary random field models (cf. Section 5). At least in general, results and performance might not be satisfactory for every model, since, obviously, many layout criteria are tackeled more efficiently by specialized algorithms. Still, capability to experiment with new or modified layout criteria save from the need to implement a new algorithm every time the model is changed enables some kind of *rapid prototyping of layout models*. Moreover, the model designer has total control over relative importance of combined layout criteria. We envision a system featuring interactive identification of suitable layout criteria by allowing the specification of clique potentials at run time. This is of particular importance when communicating with experts of the problem domain from which the graph arose. Once a suitable model is identified, the implementation can be fine-tuned to its energy function, or tailored to an equivalent combinatorial formulation.

References

[AK89] Emile H.L. Aarts and J.H.M. Korst. *Simulated Annealing and Boltzmann Machines*. Wiley, 1989.

[Aze90] Robert Azencott. Synchronous Boltzmann machines and Gibbs fields: Learning algorithms. In *Neurocomputing*, volume F68 of *NATO ASI Series*, pages 51–62. Springer–Verlag New York, Inc., 1990.

[Bes86] Julian Besag. On the statistical analysis of dirty pictures. *J. R. Statist. Soc. B*, 48(3):259–302, 1986.

[BHR96] Franz J. Brandenburg, Michael Himsolt, and Christoph Rohrer. An Experimental Comparison of Force-Directed and Randomized Graph Drawing Algorithms. In Franz J. Brandenburg, editor, *Graph Drawing. Proceedings of the DIMACS International Workshop, GD'95, Passau, Germany, September 20-22, 1995*, pages 76–87. Springer-Verlag, Lecture Notes in Computer Science, vol. 1027, 1996.

[CS93] Ming-Hui Chen and Bruce Schmeiser. Performance of the Gibbs, Hit-and-Run, and Metropolis Samplers. *J. of Computational and Graphical Statistics*, 2(3):251–272, 1993.

[CT94] Isabel F. Cruz and Roberto Tamassia. How to visualize a graph: Specifications and algorithms. Tutorial on Graph Drawing. Presented at IEEE 10th International Symposium on Visual Languages, VL'94, St. Louis, October 1994. Available at `http://www.cs.brown.edu/calendar/gd94/tutorial.html`.

[CT96] Isabel F. Cruz and Joseph P. Twarog. 3D Graph Drawing with Simulated Annealing. In Franz J. Brandenburg, editor, *Graph Drawing. Proceedings of the DIMACS International Workshop, GD'95, Passau, Germany, September 20-22, 1995*, pages 162–165. Springer-Verlag, Lecture Notes in Computer Science, vol. 1027, 1996.

[DETT94] Giuseppe Di Battista, Peter Eades, Roberto Tamassia, and Ioannis G. Tollis. Algorithms for Drawing Graphs: an Annotated Bibliography. *Computational Geometry*, 4:235–282, 1994.

[DH96] Ron Davidson and David Harel. Drawing graphs nicely using simulated annealing. *ACM Transactions on Graphics*, 15(4):301–331, 1996.

[DMM97] Uğur Doğrusöz, Brendan Madden, and Patrick Madden. Circular Layout in the Graph Layout Toolkit. In Stephen North, editor, *Proceedings of the Symposium on Graph Drawing, GD'96, Berkeley, California, USA, September 18-20, 1996*, pages 92–100. Springer-Verlag, Lecture Notes in Computer Science, vol. 1190, 1997.

[Ead84] Peter Eades. A heuristic for graph drawing. *Congressus Numerantium*, 42:149–160, 1984.

[FLM95] Arne Frick, Andreas Ludwig, and Heiko Mehldau. A Fast Adaptive Layout Algorithm for Undirected Graphs. In Roberto Tamassia and Ioannis G. Tollis, editors, *Graph Drawing. Proceedings of the DIMACS International Workshop, GD'94, Princeton, New Jersey, USA, October 10-12, 1994*, pages 388–403. Springer-Verlag, Lecture Notes in Computer Science, vol. 894, 1995.

[FR91] Thomas M.J. Fruchterman and Edward M. Reingold. Graph-drawing by force-directed placement. *Software–Practice and Experience*, 21(11):1129–1164, 1991.

[GG84] Stuart Geman and Donald Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.

[GJ91] Michael R. Garey and David S. Johnson. *Computers and intractability: a guide to the theory of NP–completeness*. W H Freeman & Co Ltd, 1991.

[Gri76] David Griffeath. Introduction to random fields. In John G. Kemeny, James L. Snell, and Anthony W. Knapp, editors, *Denumerable Markov Chains*, pages 425–458. Springer–Verlag New York, Inc., 1976.

[Guy94] Xavier Guyon. *Random Fields on a Network*. Springer, 1994.

[HS95] David Harel and Meir Sardas. Randomized graph drawing with heavy-duty preprocessing. *J. of Visual Languages and Computing*, 6:233–253, 1995. Special Issue on Graph Visualization.

[Isi25] Ernst Ising. Beitrag zur Theorie des Ferromagnetismus. *Zeitschrift Physik*, 31:235–258, 1925.

[KGV83] S. Kirkpatrick, C.D. Gellatt, and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[KK89] Tomihisa Kamada and Satoru Kawai. An algorithm for drawing general undirected graphs. *Information Processing Letters*, 31:7–15, 1989.

[LE95] Tao Lin and Peter Eades. Integration of Declarative and Algorithmic Approaches for Layout Creation. In Roberto Tamassia and Ioannis G. Tollis, editors, *Graph Drawing. Proceedings of the DIMACS International Workshop, GD'94, Princeton, New Jersey, USA, October 10-12, 1994*, pages 376–387. Springer-Verlag, Lecture Notes in Computer Science, vol. 894, 1995.

[MP91] John Alan McDonald and Jan Pedersen. Geometric abstractions for constrained optimization of layouts. In Andreas Buja and Paul A. Tukey, editors, *Computing and Graphics in Statistics*, volume 36 of *The IMA Volumes in Mathematics and its Applications*, pages 95–105. Springer-Verlag, Berlin, 1991.

[MRR+53] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *J. of Chemical Physics*, 21(6):1087–1092, 1953.

[SM95] Kozo Sugiyama and Kazuo Misue. A simple and Unified Method for Drawing Graphs: Magnetic-Spring Algorithm. In Roberto Tamassia and Ioannis G. Tollis, editors, *Graph Drawing. Proceedings of the DIMACS International Workshop, GD'94, Princeton, New Jersey, USA, October 10-12, 1994*, pages 364–375. Springer-Verlag, Lecture Notes in Computer Science, vol. 894, 1995.

[STT81] Kozo Sugiyama, Shojiro Tagawa, and Mitsuhiko Toda. Methods for visual understanding of hierarchical system structures. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(2):109–125, 1981.

[Tun94] Daniel Tunkelang. A practical approach to drawing undirected graphs. Technical Report CMU-CS-94-161, Carnegie Mellon University, School of Computer Science, 1994.

[Tut63] William T. Tutte. How to draw a graph. *Proc. London Math. Soc.*, 13(3):743–768, 1963.

[TX95] Ioannis G. Tollis and Chunliang Xia. Drawing Telecommunication Networks. In Roberto Tamassia and Ioannis G. Tollis, editors, *Graph Drawing. Proceedings of the DIMACS International Workshop, GD'94, Princeton, New Jersey, USA, October 10-12, 1994*, pages 206–217. Springer-Verlag, Lecture Notes in Computer Science, vol. 894, 1995.

[vLA88] Peter J.M. van Laarhoven and Emile H.L. Aarts. *Simulated Annealing: Theory and Applications*. Reidel, 1988.

[Win95] Gerhard Winkler. *Image Analysis, Random Fields and Dynamic Monte Carlo Methods*, volume 27 of *Applications of Mathematics*. Springer, 1995.

[WM96] Xiaobo Wang and Isao Miyamoto. Generating Customized Layouts. In Franz J. Brandenburg, editor, *Graph Drawing. Proceedings of the DIMACS International Workshop, GD'95, Passau, Germany, September 20-22, 1995*, pages 504–515. Springer-Verlag, Lecture Notes in Computer Science, vol. 1027, 1996.