Visual Clustering of Graphs with Nonuniform Degrees*

Andreas Noack

Institute of Computer Science
Brandenburg University of Technology at Cottbus
PO Box 10 13 44, 03013 Cottbus, Germany
Phone: +49 355 69-3803; Fax: +49 355 69-3810
an@informatik.tu-cottbus.de

Abstract. We discuss several criteria for clustering graphs, and identify two criteria which are not biased towards certain cluster sizes: the node-normalized cut (also called cut ratio) and the edge-normalized cut. We present two energy models whose minimum energy drawings reveal clusters with respect to these criteria. The energy model that corresponds to the edge-normalized cut differs from the other energy model in that it is also useful for graphs with very nonuniform node degrees. We show that its drawings provide insights into the structure of an airline routing graph, a citation graph, a social network, and a thesaurus graph.

1 Introduction

Force-directed and energy-based graph drawing methods are widely used. They are applicable to general undirected graphs, adaptable to different drawing criteria, reasonably easy to implement, and give satisfactory results for many graphs ([10, Chap. 10], [7]).

Energy-based methods generally have two parts: an energy model, and an algorithm that searches a state with minimum total energy. In force-directed methods, the model is a force system, and an algorithm searches for an equilibrium state where the total force on each node is zero. Because force is the negative gradient of energy, this corresponds to searching a local minimum of energy. This paper takes the perspective of energy minimization, and focusses on energy models, not on minimization algorithms.

Finding clusters, i.e. subsets of nodes with many internal edges and few edges to outside nodes, in graphs is an important problem in VLSI design [2], parallel computing [28], software engineering [25], and graph drawing [8]. This paper presents two energy models that reveal the clusters of the drawn graph, named node-repulsion LinLog and edge-repulsion LinLog. While the applicability of the node-repulsion LinLog model (introduced in [27]) is basically limited to graphs with fairly uniform degrees, the new edge-repulsion LinLog model removes this limitation.

Drawings of graphs are useful because their viewers can infer properties of the graph from properties of the drawing. Therefore we define and discuss two graph clustering criteria, and specify precisely how the minimum energy drawings of the LinLog energy models can be interpreted with respect to these criteria.

* Computer Science Report 02/2004, BTU Cottbus

The clustering criteria are discussed in Sect. 2. Section 3 presents the two LinLog energy models and their formal characterizations. Section 4 shows example drawings of several artificial and real-world graphs.

1.1 Basic Definitions

For a set M, let $|M|$ be the number of elements of M, and let $M^{(2)}$ be the set of all subsets of M which have exactly two elements. A *bipartition* of a set M is a pair (M_1, M_2) of sets such that $M_1 \cup M_2 = M$, $M_1 \cap M_2 = \emptyset$, $M_1 \neq \emptyset$, and $M_2 \neq \emptyset$.

A *graph* $G = (V, E)$ consists of a finite set V of *nodes* and a finite set E of *edges* with $E \subseteq V^{(2)}$. We only consider graphs with at least two nodes. Because layouts can be computed separately for different components of a graph, we restrict ourselves to connected graphs, i.e. graphs where every pair of nodes is connected by a path.

For a node v, the *degree* $\deg(v)$ is the number $|\{u \mid \{u, v\} \in E\}|$ of nodes adjacent to v. The total degree $\sum_{v \in V_1} \deg(v)$ of all nodes in a set V_1 is denoted by $\deg(V_1)$. For two sets of nodes V_1 and V_2, the number of edges $\big|\{\{u, v\} \in E \mid u \in V_1, v \in V_2\}\big|$ between V_1 and V_2 is called the *cut* between V_1 and V_2 and denoted by $\mathrm{cut}(V_1, V_2)$. We often identify a set of nodes V_1 with the subgraph $(V_1, \{e \in E \mid e \subseteq V_1\})$ it induces.

A d-*dimensional drawing* of the graph G is a vector $p = (p_v)_{v \in V}$ of node positions $p_v \in \mathrm{I\!R}^d$. For a drawing p and two nodes $u, v \in V$ the length of the difference vector $p_v - p_u$ is called the *distance* of u and v in p and denoted by $||p_v - p_u||$.

2 Graph Clustering Criteria

Many different definitions of the term *cluster of a graph* have been proposed (e.g. in [31,30,13,23,32,20]). Informally, we denote by a cluster a set of nodes with many internal edges and few edges to nodes outside the set. To formalize this notion, this section discusses several measures of the coupling between subgraphs, and identifies two measures that are not biased towards certain cluster sizes: the node-normalized cut and the edge-normalized cut. For simplicity, the discussion is restricted to the coupling between two subgraphs, but the generalization to more subgraphs is straightforward.

2.1 The Cut

A simple measure of the coupling between two disjoint sets of nodes V_1 and V_2 is their cut $\mathrm{cut}(V_1, V_2)$. There exist efficient algorithms for finding a bipartition of a given graph with the minimum cut [37]. However, such bipartitions do not capture our intuition of a cluster, because they mostly consist of a very small and a very large subgraph.

To make this more precise, consider the probability space $G(V, p)$ of graphs with the set of nodes V where the probability of an edge between each (unordered) pair of nodes is p, and all edges are chosen independently. For a graph from this probability space, the expected cut of a bipartition (V_1, V_2) of V is $p|V_1||V_2|$, and is thus much smaller for $|V_1| \ll |V_2|$ than for $|V_1| = |V_2|$.

2.2 The Node-Normalized Cut

Removing the bias of the cut results in an improved measure of coupling, called the *node-normalized cut*:

$$\mathrm{nodenormcut}(V_1, V_2) = \frac{\mathrm{cut}(V_1, V_2)}{|V_1| \cdot |V_2|}$$

The node-normalized cut of every bipartition of a graph from $G(V, p)$ has the same expected value p because it is normalized with $|V_1| \cdot |V_2|$.

This measure is also called the ratio of the cut, and has been used in VLSI design [2] and software engineering [25]. The decision whether a graph has a bipartition with a node-normalized cut smaller than a given constant is NP-complete [4, Problem ND23], but efficient algorithms for approximations within guaranteed bounds exist [24,3].

The node-normalized cut still does not capture our intuition of a cluster for graphs with very nonuniform node degrees. Consider two partitions of the set of nodes into two sets V_1 and V_2 with $|V_1| = |V_2|$. In the first partition $\deg(V_1) = \deg(V_2)$, while in the second partition $\deg(V_1) \ll \deg(V_2)$. Then we expect the cut (and therefore the node-normalized cut) to be much larger for the first partition than for the second.

To make this more precise, consider the random graph model $G(V, (p_v)_{v \in V})$ where the probability of an edge $\{u, v\}$ (including, for technical reasons, the case $u = v$, i.e. loops) is $p_u p_v$, and all edges are chosen independently. Then the expected cut between two sets of nodes V_1 and V_2 is $\sum_{v \in V_1} p_v \cdot \sum_{v \in V_2} p_v$. For example, let $e = \sum_{v \in V} p_v$ and $e \gg 1$, then the expected cut is $e^2/4$ if $\sum_{v \in V_1} p_v = \sum_{v \in V_2} p_v = e/2$, and the expected cut is only $e-1$ if $\sum_{v \in V_1} p_v = 1$ and $\sum_{v \in V_2} p_v = e-1$.

2.3 The Edge-Normalized Cut

To remove the bias observed in the previous subsection, we consider a given graph (V, E) as element of the probability space $G\big(V, (\deg(v)/\sqrt{\deg(V)})_{v \in V}\big)$, i.e. we choose the p_v such that the expected degree of every node equals its actual degree in the given graph. Then the expected cut between two disjoint sets of nodes V_1 and V_2 is $\deg(V_1)\deg(V_2)/\deg(V)$. So the following measure of coupling called the *edge-normalized cut* has the same expected value $1/\deg(V)$ for every bipartition of a graph from $G\big(V, (\deg(v)/\sqrt{\deg(V)})_{v \in V}\big)$:

$$\mathrm{edgenormcut}(V_1, V_2) = \frac{\mathrm{cut}(V_1, V_2)}{\deg(V_1)\deg(V_2)}$$

A similar measure has been introduced by Shi and Malik [32] as normalized cut:

$$\mathrm{ncut}(V_1, V_2) = \frac{\mathrm{cut}(V_1, V_2)}{\deg(V_1)} + \frac{\mathrm{cut}(V_1, V_2)}{\deg(V_2)}.$$

Because $\deg(V)\,\mathrm{edgenormcut}(V_1, V_2) = \mathrm{ncut}(V_1, V_2)$, the values of the two measures differ only by a constant factor for a given graph. We prefer the name edge-normalized cut and the earlier formula to emphasize the parallels to the node-normalized cut. The problem of deciding whether a given graph has a bipartition with an edge-normalized cut smaller than a given constant is NP-hard [32].

2.4 Related Work: Other Measures of Coupling

Other measures of the coupling between two nonempty disjoint sets of nodes V_1 and V_2 include the expansion [20] (also called quotient cut [4])

$$\text{expansion}(V_1, V_2) = \frac{\text{cut}(V_1, V_2)}{\min(|V_1|, |V_2|)}$$

and the conductance [20]

$$\text{conductance}(V_1, V_2) = \frac{\text{cut}(V_1, V_2)}{\min(\deg(V_1), \deg(V_2))}.$$

Minimizing the expansion is NP-complete [4, Problem ND26], but there are efficient approximation algorithms [22].

The expansion is biased towards similarly-sized clusters for random graphs from $G(V, p)$: For $|V_1| = 1$ and $|V_2| = |V| - 1$, the expected expansion between V_1 and V_2 is $p(|V| - 1)$, while for $|V_1| = |V_2| = |V|/2$, the expected expansion is only $0.5p|V|$. The conductance has a similar bias for random graphs from $G(V, (p_v)_{v \in V})$. Therefore we prefer the node-normalized and edge-normalized cut as measures of coupling.

3 Energy Models for Visual Graph Clustering

In this section, we present two energy models for visual graph clustering: The node-repulsion LinLog energy model was shown to clearly visualize clusters with respect to the node-normalized cut in [27]. We extend this energy model to the edge-repulsion LinLog energy model, and show that this new energy model clearly visualizes clusters with respect to the edge-normalized cut. This makes the edge-repulsion LinLog energy model more suitable for drawing graphs with very nonuniform degrees.

3.1 Definition of the Energy Models

The *node-repulsion LinLog energy* of a drawing p is defined in [27] as

$$U_{NodeLinLog}(p) = \sum\nolimits_{\{u,v\} \in E} ||p_u - p_v|| - \sum\nolimits_{\{u,v\} \in V^{(2)}} \ln ||p_u - p_v||$$

To avoid infinite energies we assume that different nodes have different positions, which is no serious restriction because we are interested in drawings with low energy. The first term of the difference can be interpreted as attraction between adjacent nodes, the second term as repulsion between different nodes.

In the *edge-repulsion LinLog energy model* the repulsion between nodes is replaced by repulsion between edges. More precisely, the repulsion does not act between entire edges, but only between their end nodes. So the repulsion between two nodes is weighted by the number of edges of which they are an end node, i.e. by their degrees:

$$U_{EdgeLinLog}(p) = \sum\nolimits_{\{u,v\} \in E} ||p_u - p_v|| - \sum\nolimits_{\{u,v\} \in V^{(2)}} \deg(u) \deg(v) \ln ||p_u - p_v||$$

The basic idea behind the edge-repulsion LinLog model is that the edges cause both attraction and repulsion. From the perspective of nodes, the total strength of the attractive *force* originating from a node is its degree, and its repulsion is weighted with its degree, too. This gives each node consistently – in terms of attraction and repulsion – an influence on the layout proportional to its degree. By the way, this can be visualized by setting the size of a node to its degree, as in the figures in Sect. 4.

In a node-repulsion LinLog drawing of a graph with very nonuniform degrees, the positions of the nodes mainly reflect their degrees: The (strongly attracting) high-degree nodes are mostly placed at the center, and the (weakly attracting, but equally repulsing) low-degree nodes at the borders. This bias is removed in the edge-repulsion LinLog model. For graphs with uniform node degrees, both models are equivalent up to scaling.

3.2 Separation of Clusters

Minimum energy drawings of the LinLog models separate clusters from the remaining graph, and place nodes of the same cluster closely together. They do this by minimizing a simple distance ratio, namely the ratio of the arithmetic mean of edge lengths to the geometric mean of the node distances. The only difference is that for the edge-repulsion LinLog model, the nodes are weighted according to their degree in the geometric mean.

Theorem 1. *Let $G = (V, E)$ be a connected graph, and let p^0 be a drawing of G with minimum node-repulsion or edge-repulsion LinLog energy. Then p^0 is a drawing that minimizes $\frac{\mathrm{arithmean}_p(E)}{\mathrm{geomean}_p(V^{(2)})}$, where $\mathrm{arithmean}_p(E)$ denotes the arithmetic mean of the edge lengths*

$$\mathrm{arithmean}_p(E) = \frac{1}{|E|} \sum\nolimits_{\{u,v\} \in E} ||p_v - p_u||,$$

and $\mathrm{geomean}_p(V^{(2)})$ denotes the weighted geometric mean of the node distances

$$\mathrm{geomean}_p(V^{(2)}) = \prod\nolimits_{\{u,v\} \in V^{(2)}} ||p_v - p_u||^{w_u w_v \,/ \sum_{\{u,v\} \in V^{(2)}} w_u w_v},$$

where $w_v = 1$ for node-repulsion LinLog and $w_v = \deg(v)$ for edge-repulsion LinLog.

Proof: Given in [27] for node-repulsion LinLog, similar for edge-repulsion LinLog.

3.3 Interpretable Distances between Clusters

In minimum energy drawings of the LinLog models, the distance of each cluster to the remaining nodes is interpretable. In node-repulsion LinLog drawings, the harmonic mean of the distances from the cluster nodes to the remaining nodes equals the inverse node-normalized cut between the cluster and the remaining nodes. Similarly, in edge-repulsion LinLog drawings, the harmonic mean of the distances from the cluster nodes to the remaining nodes, weighted according to the node degrees, equals the inverse edge-normalized cut between the cluster and the remaining nodes. This holds exactly in all one-dimensional drawings, and approximately in most practical higher-dimensional drawings (in particular, when the distance of a cluster to the remaining nodes is large).

Theorem 2. *Let $G = (V, E)$ be a connected graph, and let p be a one-dimensional drawing of G. Let (V_1, V_2) be a bipartition of V such that the nodes of V_1 have smaller positions than the nodes in V_2 (i.e. $\forall v_1 \in V_1 \forall v_2 \in V_2 : p_{v_1} < p_{v_2}$).*
If p has minimum node-repulsion LinLog energy, then

$$\frac{|V^{(2)}[V_1, V_2]|}{\sum_{\{u,v\} \in V^{(2)}[V_1,V_2]} \frac{1}{||p_v - p_u||}} = \frac{1}{\mathrm{nodenormcut}(V_1, V_2)}.$$

If p has minimum edge-repulsion LinLog energy, then

$$\frac{\sum_{\{u,v\} \in V^{(2)}[V_1,V_2]} \deg(u)\deg(v)}{\sum_{\{u,v\} \in V^{(2)}[V_1,V_2]} \frac{\deg(u)\deg(v)}{||p_v - p_u||}} = \frac{1}{\mathrm{edgenormcut}(V_1, V_2)}.$$

Here $V^{(2)}[V_1, V_2]$ contains all pairs of nodes $\{u, v\}$ with $u \in V_1$ and $v \in V_2$.

Proof: Given in [27] for node-repulsion LinLog, similar for edge-repulsion LinLog.

3.4 Algorithms for Energy Minimization

As usual in force- and energy-based graph drawing (with the exception of Hall's energy model [17]), we have no practical algorithm that finds global minima of the LinLog energy models. In our experiments we use the hierarchical energy minimization algorithm of Barnes and Hut [5], which was introduced to graph drawing by Tunkelang [34] and Quigley [29]. Its runtime is in $O(|E| + |V| \log |V|)$ per iteration. The overall runtime grows somewhat faster because the number of iterations needed for convergence tends to grow with n. Some other efficient minimization algorithms are not expected to find good energy minima for dense graphs or graphs with small diameter [16,18,35].

Besides being efficient, the algorithm has to find good energy minima. Our experience is encouraging: We have computed drawings of several graphs with known but difficult to find clusters, and the results indeed reflect the cluster structure (see Sect. 4).

3.5 Related Work

Force and Energy Models. In contrast to the LinLog models, the well-known energy models of Eades [11], of Fruchterman and Reingold [15], of Davidson and Harel [9], of Kamada and Kawai [19], of Hall [17], and multidimensional scaling [21] do not isolate clusters (with respect to the node-normalized and edge-normalized cut) well, as discussed in [26,27], and illustrated for the Fruchterman-Reingold model in Sect. 4.

Evaluation of Force and Energy Models. Force and energy models have been evaluated mainly empirically (e.g. in [6]). We complement the empirical evaluation of the edge-repulsion LinLog model in Sect. 4 with theoretical results in this section.

Drawing Clustered Graphs. Force-directed methods have been applied to draw graphs with an explicitly given hierarchical structure [36,12]. In contrast, our goal is to visualize clusters without requiring knowledge of clusters as input.

Clustering by Minimizing Distance Ratios. The relationship between clustering and minimizing distance ratios has been exploited in approximation algorithms for bipartitions with small node-normalized cuts [24,3]. However, the minima used in these algorithms are not intended to be interpretable to human viewers. Our work stresses the equivalence of minimizing such ratios to minimizing energy models, which enables the application of algorithms from energy based graph drawing for their minimization.

4 Examples

This section shows example drawings of the edge-repulsion LinLog energy model, and, for comparison, of the node-repulsion LinLog energy model and the well-known Fruchterman-Reingold force model [15]. The first subsection illustrates the differences between the models with drawings of two pseudo-random graphs. The second subsection shows that drawings of the edge-repulsion LinLog model can provide new, non-trivial, and useful insights into the structure of real-world graphs.

In all drawings, the diameter of the nodes is proportional to their degree, with the exception that there is a minimum size to ensure visibility. In most drawings, the edges are omitted to avoid clutter. All node colors and textual annotations were added manually. Some drawings were rotated manually. (Rotation does not change the energy.)

4.1 Pseudo-Random Graphs

The graph in Fig. 1 is a pseudo-random graph with eight clusters of 50 nodes each. The probability of an edge $\{u, v\}$ is 1 if u and v belong to the same cluster and 0.2 otherwise. In contrast to the Fruchterman-Reingold drawing, the LinLog drawings show the clusters clearly. They are similar because the degrees of the nodes are fairly uniform.

The graph in Fig. 2 is a pseudo-random graph with eight cluster of 50 nodes each. The probability of an edge $\{u, v\}$ is

- 1 if u and v belong to the same of the first four clusters,
- 0.5 if u and v belong to the same of the second four clusters,
- 0.2 if u and v belong to different of the first four clusters,
- 0.05 if u and v belong to different of the second four clusters, and
- 0.1 if u belongs to one of the first and v belongs to one of the second four clusters.

Both LinLog models reveal the clusters, but their drawings differ because the degrees of the nodes are nonuniform. The node-repulsion LinLog drawing places the first four clusters more closely than the second four clusters, which reflects that node-normalized cuts between the first four clusters are higher than between the second four clusters. In the edge-repulsion LinLog drawing the distances between all clusters are similar, which reflects that the edge-normalized cuts between all pairs of clusters are similar.

4.2 Real-World Graphs

Table 1 gives an overview of the graphs in Fig. 3 to 6. In all graphs, the edges were considered as undirected, and only the largest connected component was drawn. (It always contained more than 90 percent of the nodes and 98 percent of the edges.)

In the drawings of the Fruchterman-Reingold model (Fig. 3a and 6a) and the node-repulsion LinLog model (Fig. 3b and 6b), nodes with high degree are placed in the center, and nodes with low degree near the borders. So the positions of the nodes mainly reflect their degree.

In Fig. 3, only the edge-repulsion LinLog drawing clearly shows that there are two groups of friends – the left group around Chris and Rick and the right group around Steve and Irv – which are mainly connected by Upton and Dan.

Table 1. Graphs in Fig. 3 to 6

Fig.	Description	Node	Edge	# nodes	# edges	Source
3	friendship network	person	is friend of	33	91[1]	Pajek project[2]
4	GD citations 1994-2002	GD paper	cites	314	772	GraphAEL project[3]
5	Roget's thesaurus	category	references or is related to	994	5058	The Stanford GraphBase[4]
6	US airline routing	airport	direct flight	332	2126	Pajek project[2]

In the edge-repulsion LinLog drawing of the Graph Drawing citations in Fig. 4, the positions of the nodes roughly reflect the topics of the papers. (This is hard to show in a single figure because we could only annotate the nodes with the highest degrees.) For example, papers 1 to 5 and 26 to 28 deal with orthogonal graph drawing, papers 7 to 9 with 3D orthogonal graph drawing, papers 10 to 14 with 3D graph drawing, papers 15 to 17 with visibility representations, and papers 19 to 21 with force-directed methods. The difficulty of visualizing this graph is shown by an earlier drawing in [14].

The edge-repulsion LinLog drawing of Roget's thesaurus in Fig. 5a provides a nice map of (parts of) the English language, because semantically related categories are grouped together. This is exemplified by the two zoomed areas in Fig. 5b and c.

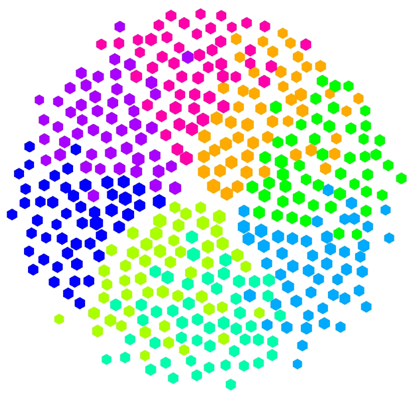
Figure 6c shows that the edge-repulsion LinLog model discovers (roughly) the relative geographical locations of the US airports from the airline routing graph. This is possible because close airports are more likely to be connected by a direct flight. In all drawings in Fig. 6 the airports in Alaska and the South Sea (e.g. Guam) are omitted to improve the readability of the remaining drawing.

5 Conclusion

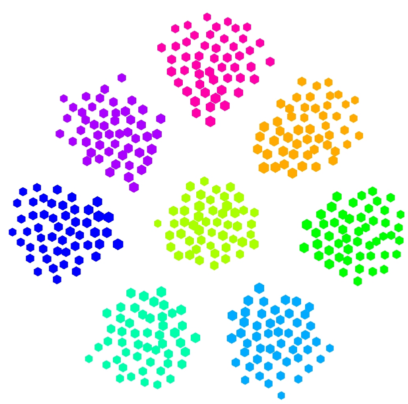
When graphs with very nonuniform degrees are drawn with the node-repulsion LinLog energy model, the positions of the nodes mainly reflect their degree. We identified a measure of coupling, called edge-normalized cut, that is normalized against degrees, and introduced an energy model, called edge-repulsion LinLog, whose drawings reveal clusters with respect to this coupling measure. Examples drawings of this energy model provided useful insights into practical graphs. Because many real-world graphs have clusters and nonuniform degrees [33,1], we expect the edge-repulsion LinLog model to be widely applicable.

We see two main directions for future work. Firstly, the development of new energy minimization algorithms and the adaption of existing algorithms specifically for the LinLog models and their addressed graphs is largely unexplored. Secondly, the edge-repulsion LinLog model is only suitable for graphs where the degree of a node can be interpreted as its importance. Other graphs with nonuniform degrees, like graphs of module uses in software systems, require other energy models.

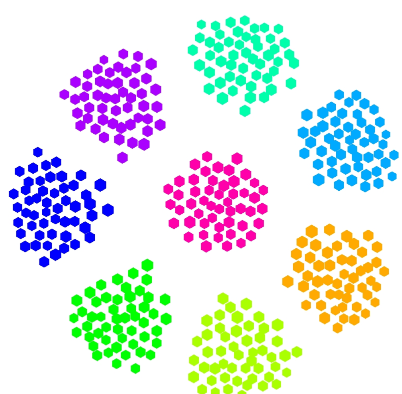
[1] includes reciprocated and nonreciprocated relationships
[2] `http://vlado.fmf.uni-lj.si/pub/networks/data/`
[3] `http://graphael.cs.arizona.edu/`
[4] `http://www-cs-faculty.stanford.edu/~knuth/sgb.html`



(a) Fruchterman-Reingold model

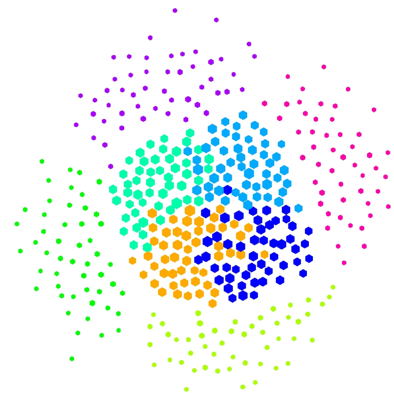


(b) Node-repulsion LinLog model

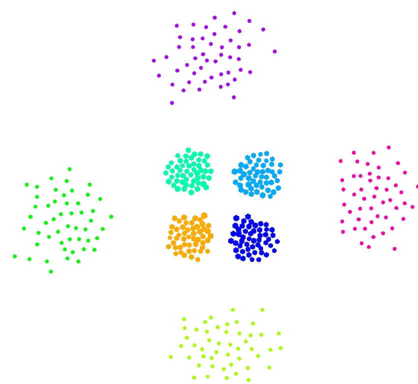


(c) Edge-repulsion LinLog model

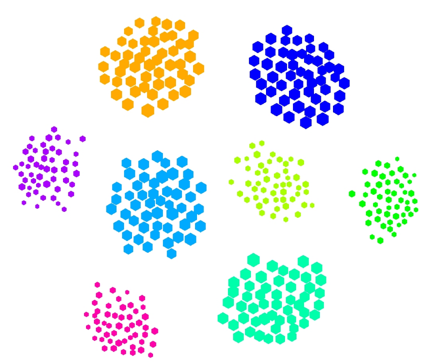
Fig. 1. First random graph



(a) Fruchterman-Reingold model

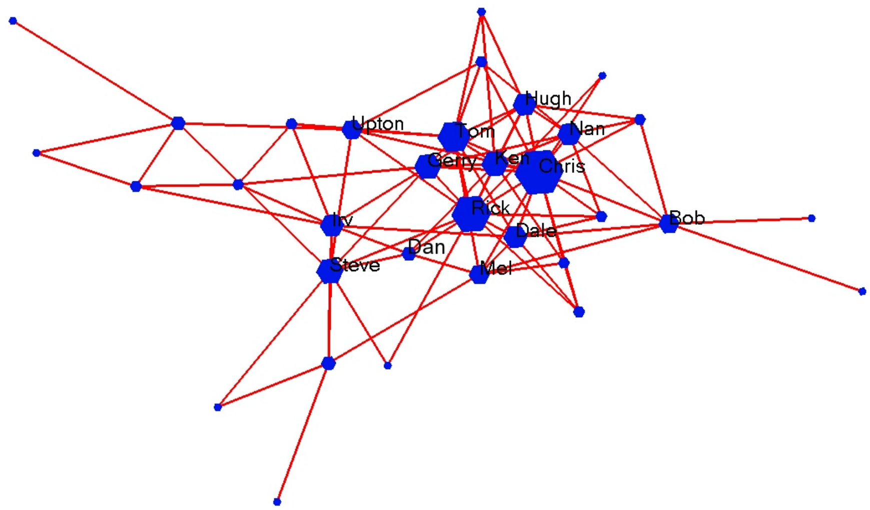


(b) Node-repulsion LinLog model

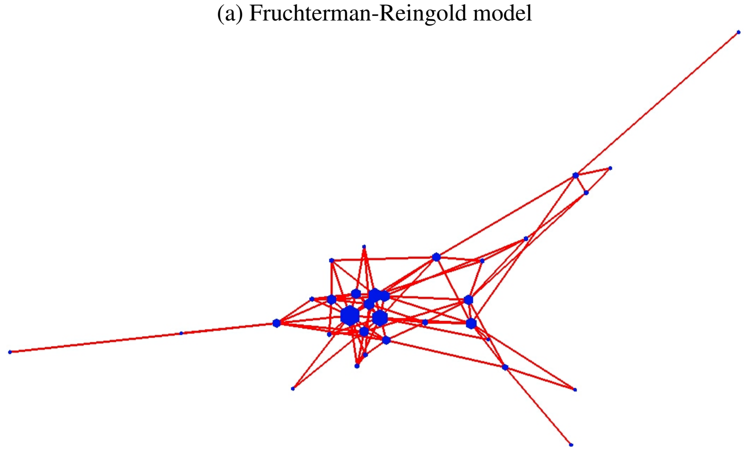


(c) Edge-repulsion LinLog model

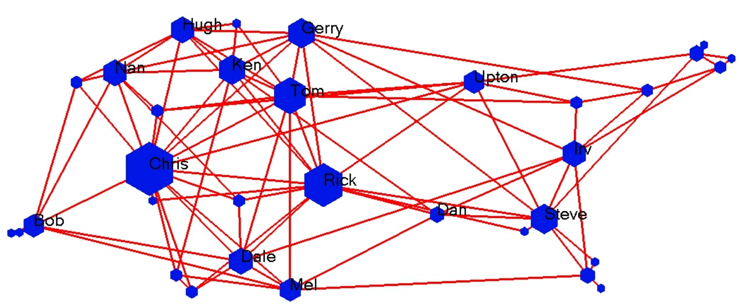
Fig. 2. Second random graph



(a) Fruchterman-Reingold model

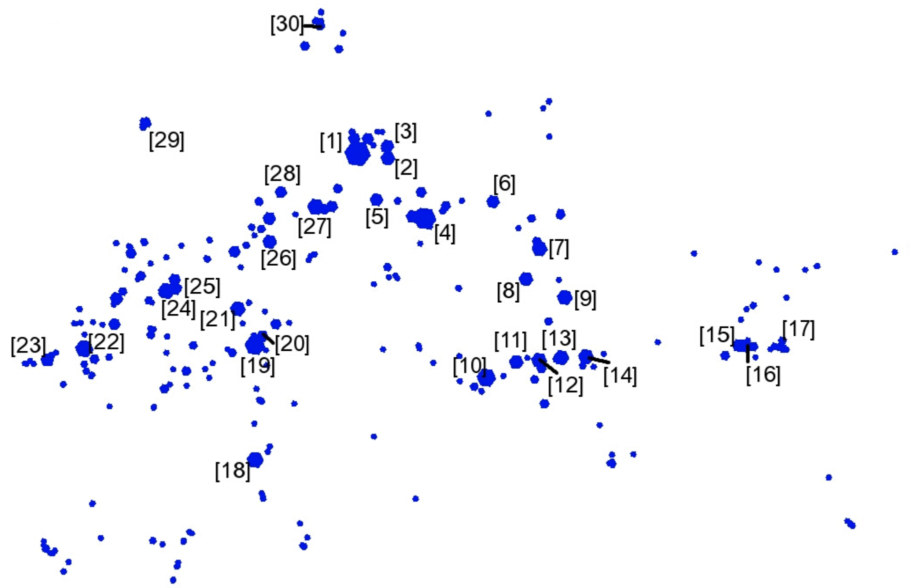


(b) Node-repulsion LinLog model



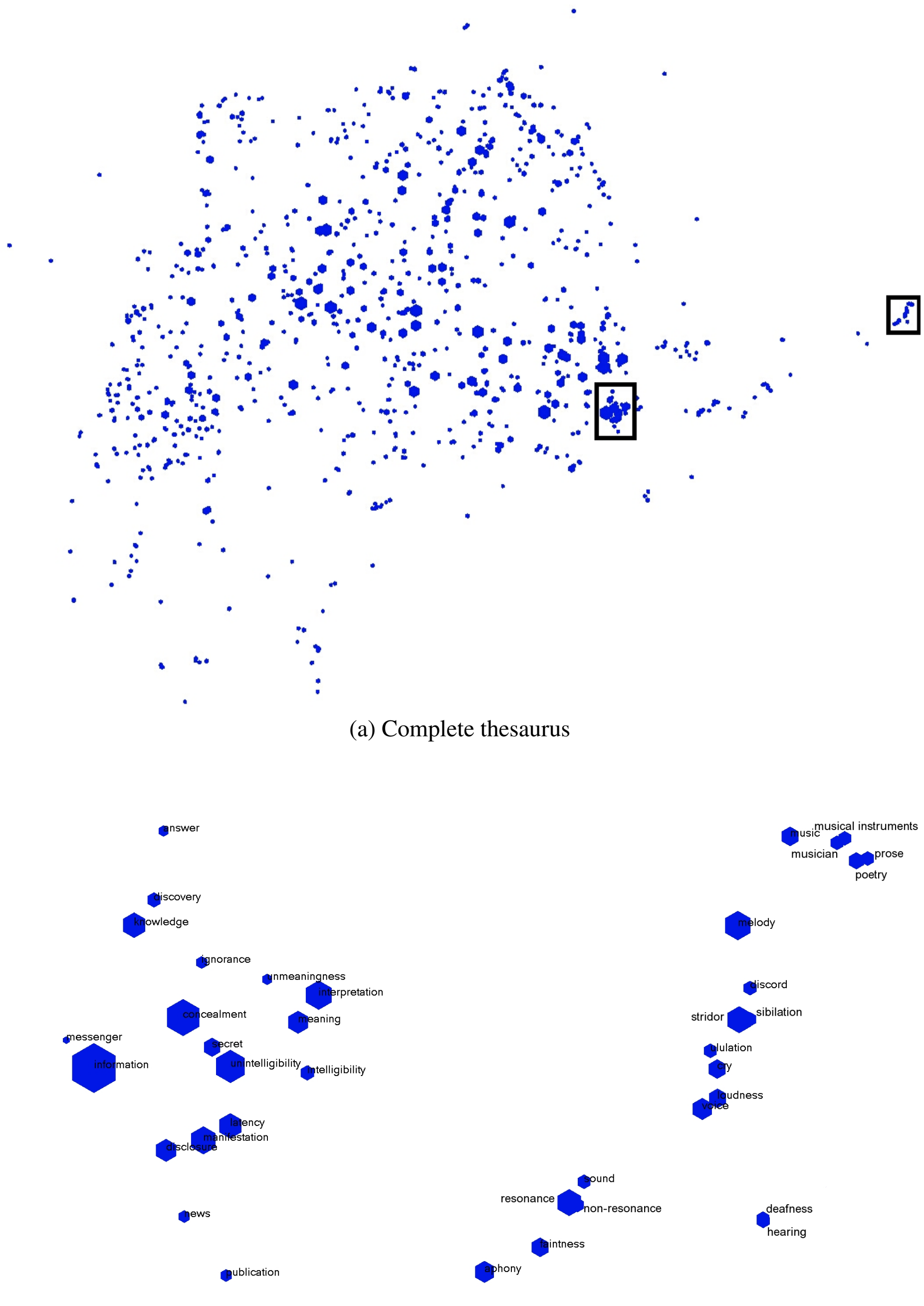
(c) Edge-repulsion LinLog model

Fig. 3. Friendship network



1. Fößmeier, Kaufmann: Drawing High Degree Graphs with Low Bend Numbers (95)
2. Biedl, Madden, Tollis: The Three-Phase Method: A Unified Approach to Orthogonal Graph Drawing (97)
3. Fößmeier, Kant, Kaufmann: 2-Visibility Drawings of Planar Graphs (96)
4. Garg, Tamassia: On the Computational Complexity of Upward and Rectilinear Planarity Testing (94)
5. Papakostas, Tollis: Improved Algorithms and Bounds for Orthogonal Drawings (94)
6. Wood: Multi-dimensional Orthogonal Graph Drawing with Small Boxes (99)
7. Eades, Symvonis, Whitesides: Two Algorithms for Three Dimensional Orthogonal Graph Drawing (96)
8. Papakostas, Tollis: Incremental Orthogonal Graph Drawing in Three Dimensions (97)
9. Wood: An Algorithm for Three-Dimensional Orthogonal Graph Drawing (98)
10. Bruß, Frick: Fast Interactive 3-D Graph Visualization (95)
11. Garg, Tamassia: GIOTTO3D: A System for Visualizing Hierarchical Structures in 3D (96)
12. Patrignani, Vargiu: 3DCube: A Tool for Three Dimensional Graph Drawing (97)
13. Webber, Scott: GOVE Grammar-Oriented Visualisation Environment (95)
14. Cohen, Eades, Lin, Ruskey: Three-Dimensional Graph Drawing (94)
15. Fekete, Houle, Whitesides: New Results on a Visibility Representation of Graphs in 3D (95)
16. Alt, Godau, Whitesides: Universal 3-Dimensional Visibility Representations for Graphs (95)
17. Bose, Di Battista, Lenhart, Liotta: Proximity Constraints and Representable Trees (94)
18. Purchase: Which Aesthetic has the Greatest Effect on Human Understanding (97)
19. Frick, Ludwig, Mehldau: A Fast Adaptive Layout Algorithm for Undirected Graphs (94)
20. Gansner, North: Improved Force-Directed Layouts (98)
21. He, Marriott: Constrained Graph Layout (96)
22. Eades, Feng: Multilevel Visualization of Clustered Graphs (96)
23. Brandenburg: Graph Clustering 1: Circles of Cliques (97)
24. Himsolt: The Graphlet System (96)
25. Lauer, Ettrich, Soukup: GraVis – System Demonstration (97)
26. Bridgeman, Tamassia: Difference Metrics for Interactive Orthogonal Graph Drawing Algorithms (98)
27. Papakostas, Tollis: Issues in Interactive Orthogonal Graph Drawing (95)
28. Brandes, Eiglsperger, Kaufmann, Wagner: Sketch-Driven Orthogonal Graph Drawing (02)
29. Graph-Drawing Contest Reports 1995-2002
30. Binucci, Didimo, Liotta, Nonato: Labeling Heuristics for Orthogonal Drawings (01)

Fig. 4. Citation graph of the Graph Drawing Symposia 1994-2002, drawn with the edge-repulsion LinLog model

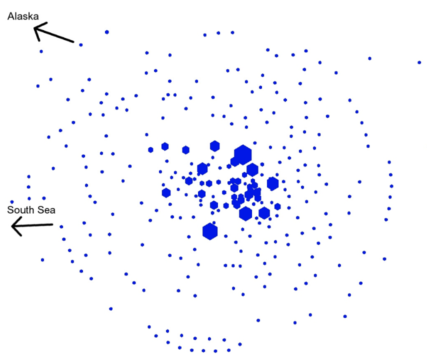


(a) Complete thesaurus

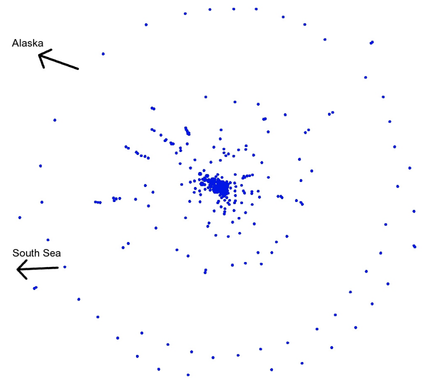
(b) Zoom into the central part

(c) Zoom into the right part

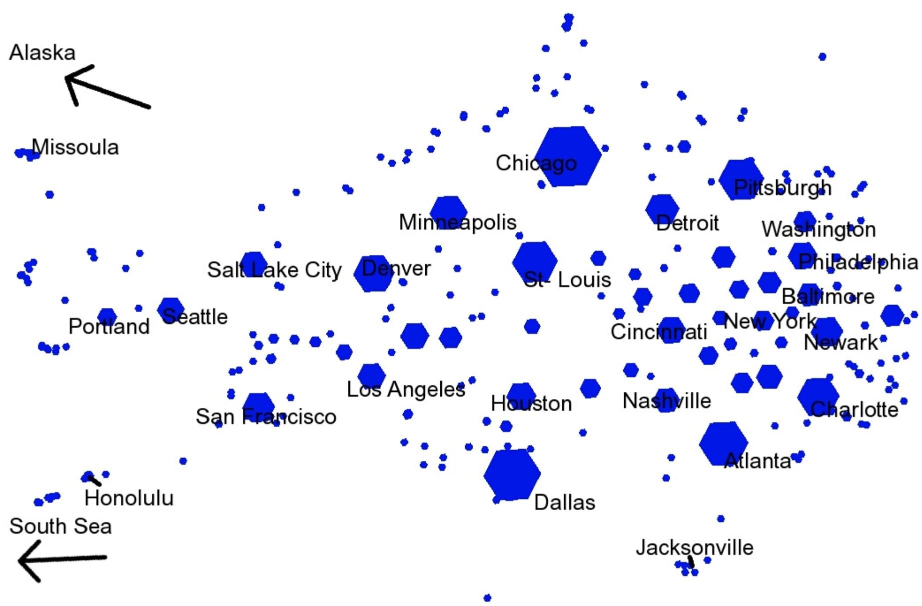
Fig. 5. Roget's thesaurus, drawn with the edge-repulsion LinLog model



(a) Fruchterman-Reingold model



(b) Node-repulsion LinLog model



(c) Edge-repulsion LinLog model

Fig. 6. Flights between US airports

References

1. Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47–97, 2002.
2. Charles J. Alpert and Andrew B. Kahng. Recent directions in netlist partitioning: A survey. *Integration, the VLSI Journal*, 19(1-2):1–81, 1995.
3. Yonatan Aumann and Yuval Rabani. An $O(logk)$ approximate min-cut max-flow theorem and approximation algorithm. *SIAM Journal on Computing*, 27(1):291–301, 1998.
4. G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi. *Complexity and Approximation*. Springer-Verlag, Berlin, 1999.
5. Josh Barnes and Piet Hut. A hierarchical O(N log N) force-calculation algorithm. *Nature*, 324:446–449, 1986.
6. Franz-Josef Brandenburg, Michael Himsolt, and Christoph Rohrer. An experimental comparison of force-directed and randomized graph drawing algorithms. In F.-J. Brandenburg, editor, *Proceedings of the Symposium on Graph Drawing (GD 1995)*, LNCS 1027, pages 76–87, Berlin, 1996. Springer-Verlag.
7. Ulrik Brandes. Drawing on physical analogies. In M. Kaufmann and D. Wagner, editors, *Drawing Graphs: Methods and Models*, LNCS 2025, pages 71–86. Springer-Verlag, Berlin, 2001.
8. Ralf Brockenauer and Sabine Cornelsen. Drawing clusters and hierarchies. In M. Kaufmann and D. Wagner, editors, *Drawing Graphs: Methods and Models*, LNCS 2025. Springer-Verlag, Berlin, 2001.
9. Ron Davidson and David Harel. Drawing graphs nicely using simulated annealing. *ACM Transactions on Graphics*, 15(4):301–331, 1996.
10. Giuseppe Di Battista, Peter Eades, Roberto Tamassia, and Ioannis G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall, Upper Saddle River, NJ, 1999.
11. Peter Eades. A heuristic for graph drawing. *Congressus Numerantium*, 42:149–160, 1984.
12. Peter Eades and Mao Lin Huang. Navigating clustered graphs using force-directed methods. *Journal of Graph Algorithms and Applications*, 4(3):157–181, 2000.
13. Jubin Edachery, Arunabha Sen, and Franz J. Brandenburg. Graph clustering using distance-k cliques. In J. Kratochvíl, editor, *Proceedings of the 7th International Symposium on Graph Drawing (GD 1999)*, LNCS 1731, pages 98–106, Berlin, 1999. Springer-Verlag.
14. Cesim Erten, Philip J. Harding, Stephen G. Kobourov, Kevin Wampler, and Gary Yee. GraphAEL: Graph animations with evolving layouts. In G. Liotta, editor, *Proceedings of the 11th International Symposium on Graph Drawing (GD 2003)*, LNCS 2912, pages 98–110, Berlin, 2004. Springer-Verlag.
15. Thomas M. J. Fruchterman and Edward M. Reingold. Graph drawing by force-directed placement. *Software – Practice and Experience*, 21(11):1129–1164, 1991.
16. Pawel Gajer, Michael T. Goodrich, and Stephen G. Kobourov. A multi-dimensional approach to force-directed layouts of large graphs. In J. Marks, editor, *Proceedings of the 8th International Symposium on Graph Drawing (GD 2000)*, LNCS 1984, pages 211–221, Berlin, 2001. Springer-Verlag.
17. Kenneth M. Hall. An r-dimensional quadratic placement algorithm. *Management Science*, 17(3):219–229, 1970.
18. David Harel and Yehuda Koren. A fast multi-scale method for drawing large graphs. In J. Marks, editor, *Proceedings of the 8th International Symposium on Graph Drawing (GD 2000)*, LNCS 1984, pages 183–196, Berlin, 2001. Springer-Verlag.
19. Tomihisa Kamada and Satoru Kawai. An algorithm for drawing general undirected graphs. *Information Processing Letters*, 31(1):7–15, 1989.

20. Ravi Kannan, Santosh Vempala, and Adrian Vetta. On clusterings: Good, bad and spectral. *Journal of the ACM*, 51(3):497–515, 2004.
21. Joseph B. Kruskal and Myron Wish. *Multidimensional Scaling*. Sage Publications, Beverly Hills, CA, 1978.
22. Tom Leighton and Satish Rao. An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms. In *Proceedings of the 29th Annual Symposium on Foundations of Computer Science (FOCS 1988)*, pages 422–431, 1988.
23. Neal Lesh, Joe Marks, and Maurizio Patrignani. Interactive partitioning. In J. Marks, editor, *Proceedings of the 8th International Symposium on Graph Drawing (GD 2000)*, LNCS 1984, pages 31–36, Berlin, 2001. Springer-Verlag.
24. Nathan Linial, Eran London, and Yuri Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15(2):215–245, 1995.
25. S. Mancoridis, B. S. Mitchell, C. Rorres, Y. Chen, and E. R. Gansner. Using automatic clustering to produce high-level system organizations of source code. In *Proceedings of the 6th IEEE International Workshop on Program Comprehension (IWPC 1998)*, pages 45–52. IEEE Computer Society, 1998.
26. Andreas Noack. Energy models for drawing clustered small-world graphs. Technical Report 07/03, Institute of Computer Science, Brandenburg University of Technology at Cottbus, 2003.
27. Andreas Noack. An energy model for visual graph clustering. In G. Liotta, editor, *Proceedings of the 11th International Symposium on Graph Drawing (GD 2003)*, LNCS 2912, pages 425–436, Berlin, 2004. Springer-Verlag.
28. Alex Pothen. Graph partitioning algorithms with applications to scientific computing. In D. E. Keyes, A. Sameh, and V. Venkatakrishnan, editors, *Parallel Numerical Algorithms*, pages 323–368. Kluwer, 1997.
29. Aaron J. Quigley and Peter Eades. FADE: Graph drawing, clustering, and visual abstraction. In J. Marks, editor, *Proceedings of the 8th International Symposium on Graph Drawing (GD 2000)*, LNCS 1984, pages 197–210, Berlin, 2001. Springer-Verlag.
30. Tom Roxborough and Arunabha Sen. Graph clustering using multiway ratio cut. In G. Di Battista, editor, *Proceedings of the 5th International Symposium on Graph Drawing (GD 1997)*, LNCS 1353, pages 291–296, Berlin, 1998. Springer-Verlag.
31. Reinhard Sablowski and Arne Frick. Automatic graph clustering. In S. C. North, editor, *Proceedings of the Symposium on Graph Drawing (GD 1996)*, LNCS 1190, pages 395–400, Berlin, 1997. Springer-Verlag.
32. Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.
33. Steven H. Strogatz. Exploring complex networks. *Nature*, 410:268–276, 2001.
34. Daniel Tunkelang. *A Numerical Optimization Approach to General Graph Drawing*. PhD thesis, School of Computer Science, Carnegie Mellon University, 1999.
35. Chris Walshaw. A multilevel algorithm for force-directed graph drawing. In J. Marks, editor, *Proceedings of the 8th International Symposium on Graph Drawing (GD 2000)*, LNCS 1984, pages 171–182, Berlin, 2001. Springer-Verlag.
36. Xiaobo Wang and Isao Miyamoto. Generating customized layouts. In F.-J. Brandenburg, editor, *Proceedings of the Symposium on Graph Drawing (GD 1995)*, LNCS 1027, pages 504–515, Berlin, 1996. Springer-Verlag.
37. Zhenyu Wu and Richard Leahy. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 15(11):1101–1113, 1993.