
Journal of Graph Algorithms and Applications
http://jgaa.info/ vol. 11, no. 2, pp. 453–480 (2007)

Energy Models for Graph Clustering

Andreas Noack

Institute of Computer Science
Brandenburg Technical University, Cottbus, Germany
an@informatik.tu-cottbus.de

Abstract

The cluster structure of many real-world graphs is of great interest, as the clusters may correspond e.g. to communities in social networks or to cohesive modules in software systems. Layouts can naturally represent the cluster structure of graphs by grouping densely connected nodes and separating sparsely connected nodes. This article introduces two energy models whose minimum energy layouts represent the cluster structure, one based on repulsion between nodes (like most existing energy models) and one based on repulsion between edges. The latter model is not biased towards grouping nodes with high degrees, and is thus more appropriate for the many real-world graphs with right-skewed degree distributions. The two energy models are shown to be closely related to widely used quality criteria for graph clusterings – namely the density of the cut, Shi and Malik's normalized cut, and Newman's modularity – and to objective functions optimized by eigenvector-based graph drawing methods.

Article Type	Communicated by	Submitted	Revised
Regular paper	P. Eades and P. Healy	January 2006	February 2007

1 Introduction

Researchers from Herbert Simon [47] to Mark Newman [39] have observed that many real-world systems share a common structure: They are decomposable into subsystems with strong intra-subsystem interactions and relatively weak inter-subsystem interactions. These subsystems are of great interest, as they potentially correspond e.g. to groups of friends or collaborators in social networks, closely interlocked countries in international trade, semantically related documents in hypertexts, or cohesive modules in software systems. If the system elements are modeled as nodes and their interactions as edges, then the subsystems correspond to *graph clusters*, i.e. to groups of densely connected nodes, and the subsystem structure can be represented as a *graph layout*, i.e. as assignment of the nodes to positions in low-dimensional Euclidean space. The goal of this article is to introduce and evaluate measures (called *energy models*) that quantify how well a given graph layout reflects the graph clusters, i.e. how well it groups densely connected nodes and separates sparsely connected nodes. Together with existing energy minimization algorithms, these energy models enable the efficient computation and comprehensible presentation of the subsystem structure in many real-world systems.

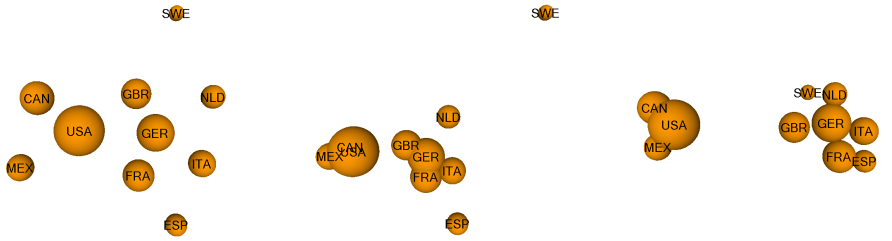
Most existing energy models and force models[1] for general undirected graphs (e.g. [16, 30, 20, 13]) have not been designed to find clusters, but to produce readable visualizations. They enforce small and uniform edge lengths, which often prevents the separation of nodes in different clusters. As a side effect, they tend to group nodes with large degree (i.e. with many edges) in the center of the layout, where their distance to the remaining nodes is relatively small. The two new energy models in this work, called *node-repulsion LinLog* and *edge-repulsion LinLog*, will be shown to group nodes according to two well-known clustering criteria, namely the density of the cut (e.g. [33, 38]) and Shi and Malik's normalized cut [45]. The normalized cut and the edge-repulsion LinLog energy model are not biased towards grouping nodes with high degree, and are thus particularly appropriate for graphs with right-skewed degree distributions, which are very common in practice [48, 1, 39].

The difference between conventional energy models, node-repulsion LinLog, and edge-repulsion LinLog can be illustrated with a model of the trade between ten North American and European countries[2]. The nodes of the graph correspond to the countries, and the edge weights specify the trade volume between each pair of countries. Because of geographical closeness and free trade agreements, countries on the same continent trade more intensively than countries on different continents. Figure 1 shows the minimum energy layouts of the trade graph for the three force and energy models. The layout of the widely used Fruchterman-Reingold model [20] does not show any clear groups at all. The layout of the node-repulsion LinLog energy model groups the countries (nodes)

[1] Force models are only alternative representations of energy models: Force is the negative gradient of energy, and thus an equilibrium of forces is a local minimum of energy.

[2] Data source: Subset of the bilateral trade data for the year 1999 from the World Bank (www.worldbank.org).

primarily according to their total trade volume (degree). Only the layout of the edge-repulsion LinLog energy model shows the expected grouping according to continents.



(a) Fruchterman-Reingold (b) Node-repulsion LinLog (c) Edge-repulsion LinLog

Figure 1: Minimum energy layouts of a trade graph

The remainder of this section clarifies the goals of this article by contrasting them with related non-goals, and introduces some notations. Section 2 defines and motivates two graph clustering criteria, namely the density of the cut and Shi and Malik's normalized cut. Section 3 introduces the two LinLog energy models, and demonstrates the internal validity of their layouts, by showing that they group the nodes according to the two clustering criteria. Section 4 shows that the edge-repulsion LinLog energy model produces externally valid layouts of several real-world graphs.

1.1 Goals and Limitations

Interpretability vs. Readability The primary purpose of most energy-based graph layout techniques is to produce easily *readable* box-and-line *visualizations* of graphs. For example, the classic energy models of Eades [16], Fruchterman and Reingold [20], and Davidson and Harel [13] primarily reward the conformance to aesthetic criteria like small and uniform edge lengths, and uniformly distributed nodes.

Graph layout techniques may also produce *interpretable* layouts, where the positions or Euclidean distances of the nodes reflect certain properties of the graph. Examples of such properties include the density of subgraphs (in this work), the graph-theoretic distances of nodes (e.g. in [30]), or the direction of edges in directed graphs (e.g. in [49]). Interpretable layouts can be seen as simple *models* of a graph, which reflect some properties of the graph and abstract from others, and which have the additional benefit of being easily visualizable. Visualizations of interpretable layouts can convey information about the edges without actually showing edges, which is essential for non-sparse graphs where showing all edges inevitably results in heavy clutter.

The goal of this work are layouts that group densely connected nodes and separate sparsely connected nodes; such layouts often violate aesthetic criteria like small edge lengths or uniformly distributed nodes.

Energy Models vs. Energy Minimization Algorithms Energy-based graph layout methods have two components: an *energy model*, which specifies *what* layouts to compute, and an *energy minimization algorithm*, which specifies *how* to compute these layouts. The algorithms are usually heuristics that do not guarantee to find global minima of the energy model.

The main contribution of this work are the LinLog energy models; algorithms for minimizing these energy models are already available. In our experiments we use the hierarchical algorithm of Barnes and Hut [5], which was introduced to graph drawing by Tunkelang [50] and Quigley and Eades [44]. Its runtime is in $O(e + n \log n)$ per iteration, where e is the number of edges and n is the number of nodes. The overall runtime grows somewhat faster because the number of iterations needed for convergence tends to grow with n. A Java implementation of the algorithm is freely available[3].

Efficient multi-scale algorithms for energy-based graph layout have been developed by Gajer et al. [21], Harel and Koren [27], Walshaw [51], and Hachul and Jünger [24]. These algorithms rely on the assumption that nodes with a small graph-theoretic distance (e.g. adjacent nodes) also have a small Euclidean distance in the optimal layout. This assumption is usually satisfied for conventional energy models that enforce uniform edge lengths, but it is not satisfied for the LinLog energy models. However, the design and evaluation of new multi-scale algorithms is beyond the scope of this article.

Theoretical vs. Empirical Validation The literature on cluster analysis distinguishes between the internal validity and the external validity of clusters [28, Chapter 4], and this distinction also applies to graph layouts that reflect clusters. *Internal validity* requires that densely connected nodes are grouped and sparsely connected nodes are separated. *External validity* requires that the grouping of the nodes conforms to an independently obtained authoritative grouping.

This work provides examples for externally valid layouts with small LinLog energy in Section 4; however, the focus is on internal validity, which is addressed in Section 3. Internal validity is demonstrated *theoretically*: It is proved that in layouts with minimum LinLog energy, 1) the ratio of the mean edge length to the mean node distance is minimal, and 2) the distance between dense subgraphs is proportional to the sparsity of their connections. Of course, these properties could also be checked *empirically* for example layouts, like previous studies of force and energy models have verified the conformance to aesthetic criteria (e.g. [9, 25]). However, such empirical validation is inferior to theoretical validation, because it is limited to a relatively small number of graphs, and because properties of the used minimization heuristics interfere with properties of the energy models. Besides the LinLog energy models, techniques for the theoretical validation of energy models are the main contribution of this article.

[3] www-sst.informatik.tu-cottbus.de/GD/erlinlog.html

1.2 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 that have exactly two elements. A *bipartition* of a set M is a pair (M_1, M_2) of sets with $M_1 \cup M_2 = M$, $M_1 \cap M_2 = \emptyset$, $M_1 \neq \emptyset$, and $M_2 \neq \emptyset$.

A *graph*[4] $G = (V, E)$ consists of a finite set V of *nodes* and a finite set E of *edges* with $E \subseteq V^{(2)}$. Because layouts can be computed separately for different components of a graph, it is assumed that graphs are connected, i.e. that every pair of nodes is connected by a path.

For a node v, the *degree* $\deg(v)$ is the number $|\{e \in E \mid v \in e\}|$ of edges incident 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 $\big|\{\{v_1, v_2\} \in E \mid v_1 \in V_1, v_2 \in V_2\}\big|$ between V_1 and V_2 is called the *cut* between V_1 and V_2 and denoted by $\mathrm{cut}(V_1, V_2)$. A set of nodes V_1 is often identified with the subgraph $(V_1, \{e \in E \mid e \subseteq V_1\})$ that it induces.

A *d-dimensional layout* of the graph G is a vector $p = (p(v))_{v \in V}$ of node positions $p(v) \in \mathbb{R}^d$. For a layout p and two nodes $u, v \in V$, the Euclidean norm 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 Clustering Criteria

Informally, we denote a subgraph as a graph cluster if it has many internal edges and few edges to the remaining graph. This can be formalized by defining a measure for the coupling between subgraphs, such that a smaller coupling indicates a better clustering. This section discusses such measures, starting with the cut. The main result is that the cut is biased towards uneven cluster sizes, and needs to be normalized. For graphs with uniform degrees, normalizing the cut with the number of nodes of the subgraphs is equivalent to normalizing the cut with the number of edges, but for graphs with nonuniform degrees, these two alternatives lead to considerably different notions of a cluster.

2.1 The Cut

A simple measure of the coupling between two disjoint sets of nodes V_1 and V_2 of a graph (V, E) 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 [23].

However, the cut prefers bipartitions that consist of a very small and a very large subgraph, as the following calculation shows. Among the $\frac{1}{2}(|V|^2 - |V|)$ unordered pairs of nodes from V, there are $|V_1| \cdot |V_2|$ pairs of one node from V_1 and one node from V_2. So the expected cut between V_1 and V_2 is $\frac{2|V_1| \cdot |V_2|}{|V|^2 - |V|}|E|$, which is much smaller for bipartitions with $|V_1| \ll |V_2|$ than for bipartitions with $|V_1| = |V_2|$.

[4] To simplify the presentation, only graphs without edge weights are considered. The generalization to graphs with edge weights is straightforward, and is discussed in Section 3.6.

2.2 The Node-Normalized Cut

An unbiased measure of the coupling between two disjoint sets of nodes V_1 and V_2 called the *node-normalized cut* is obtained by normalizing the cut with the expected cut (and ignoring constant factors for simplicity):

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

For a fixed graph (V, E) and all cluster sizes $|V_1|$ and $|V_2|$, the node-normalized cut has the same expected value $\frac{2|E|}{|V|^2-|V|}$.

This measure is also known as the density of the cut or the ratio of the cut, and has been used in VLSI design [2] and software engineering [37]. The problem of finding the bipartition of a given graph with minimum node-normalized cut is NP-hard for edge-weighted graphs [38], but approximable within factor $O(\sqrt{\log(|V|)})$ in deterministic polynomial time [3].

The node-normalized cut is still biased towards bipartitions with a very small and a very large subgraph if the number of edges is used as measure of subgraph size. Consider two bipartitions of the set of nodes V into two sets V_1 and V_2 of equal cardinality, where $\deg(V_1) = \deg(V_2)$ in the first bipartition, and $\deg(V_1) \ll \deg(V_2)$ in the second bipartition. (Note that such bipartitions only exist in graphs with nonuniform degrees.) Then the expected cut, and therefore the expected node-normalized cut, is much larger for the first bipartition than for the second.

The following calculation makes this more precise. The $|E|$ edges of a graph (V, E) have $\deg(V) = 2|E|$ end nodes. So there are $\frac{1}{2}\left(\deg(V)^2 - \sum_{v\in V} \deg(v)^2\right)$ unordered pairs of end nodes. (The negative term accounts for "pairs" of two equal end nodes.) Among these pairs, there are $\deg(V_1)\deg(V_2)$ pairs of one node from V_1 and one node from V_2. So the expected cut between $|V_1|$ and $|V_2|$ is $\frac{2\deg(V_1)\deg(V_2)}{\deg(V)^2-\sum_{v\in V}\deg(v)^2}|E|$, which is much smaller for bipartitions with $\deg(V_1) \ll \deg(V_2)$ than for bipartitions with $\deg(V_1) = \deg(V_2)$.

2.3 The Edge-Normalized Cut

Normalizing the cut with the expected cut (without constant factors) results in another measure of coupling called the *edge-normalized cut*:

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

For a fixed graph (V, E) and all clusters sizes $\deg(V_1)$ and $\deg(V_2)$, the edge-normalized cut has the same expected value $\frac{2|E|}{\deg(V)^2-\sum_{v\in V}\deg(v)^2}$.

The problem of finding the bipartition of a given graph with minimum edge-normalized cut is NP-hard for edge-weighted graphs [45], but approximable within factor $O(\sqrt{\log(|V|)})$ in deterministic polynomial time [3].

2.4 Related Work: Other Clustering Criteria

Shi and Malik's Normalized Cut Shi and Malik [45] introduced the *normalized cut* between two disjoint sets of nodes V_1 and V_2 of a graph (V, E):

$$\text{smcut}(V_1, V_2) = \frac{\text{cut}(V_1, V_2)}{\deg(V_1)} + \frac{\text{cut}(V_1, V_2)}{\deg(V_2)} \ .$$

Shi and Malik's normalized cut is closely related to the edge-normalized cut:

$$\text{smcut}(V_1, V_2) = (\deg(V_1) + \deg(V_2))\, \text{ecut}(V_1, V_2) \ .$$

So both measures differ only by a constant factor of $\deg(V)$ if $V_1 \cup V_2 = V$. However, Shi and Malik's normalized cut is biased towards small clusters when $\deg(V_1) + \deg(V_2)$ is not fixed.

Expansion and Conductance Two other well-known measures of coupling are the *expansion* (e.g. [31])

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

and the *conductance* (e.g. [31])

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

The terms *isoperimetric number* and *Cheeger constant* of a graph have been used to denote both the minimum expansion and the minimum conductance over all bipartitions, because both the number of nodes and the total degree of a (sub)graph can be considered as its area or volume.

The problems of finding the bipartition of a graph with minimum expansion or conductance are NP-hard [29, 46]. A recent $O(\sqrt{\log(|V|)})$-approximation algorithm for both problems by Arora, Rao and Vazirani [3] improves the classic $O(\log(|V|))$-approximation of Leighton and Rao [33, 34].

The expansion is related to the node-normalized cut by

$$\text{expansion}(V_1, V_2) \ = \ \max(|V_1|, |V_2|)\, \text{ncut}(V_1, V_2)$$

and thus

$$\frac{1}{2}(|V_1| + |V_2|)\, \text{ncut}(V_1, V_2) \ \leq \ \text{expansion}(V_1, V_2) \ \leq \ (|V_1| + |V_2|)\, \text{ncut}(V_1, V_2) \ .$$

The conductance is similarly related to the edge-normalized cut. Thus a bipartition whose expansion is k times the optimal expansion has also a node-normalized cut of at most $2k$ times the optimal node-normalized cut. So the

algorithms of Arora, Rao and Vazirani also approximate the node-normalized cut, the edge-normalized cut, and Shi and Malik's normalized cut within factor $O(\sqrt{\log(|V|)})$.

The expansion is biased towards similarly-sized clusters: For $|V_1| = |V|-1$ and $|V_2| = 1$, the expected expansion is $\frac{2|E|}{|V|}$, while for $|V_1| = |V_2| = \frac{1}{2}|V|$, the expected expansion is only $\frac{|E|}{|V|-1}$. The conductance has a similar bias when the total degree is used as measure of cluster size.

Newman's Modularity Newman [40] proposed a measure of coupling for k disjoint sets of nodes called *modularity*[5]:

$$Q(V_1, ..., V_k) \;=\; \sum_{i=1}^{k} \left(\frac{\mathrm{cut}(V_i, V_i)}{|E|} - \frac{\deg(V_i)^2}{\deg(V)^2} \right).$$

The first term is the fraction of all edges that are within V_i, and the second term is the *expected* value of this quantity (for graphs with loops, i.e. edges from a node to itself). To make this measure comparable to the other measures in this section, it is restricted to two sets of nodes, and corrected for the absence of loops, which yields

$$Q'(V_1, V_2) \;=\; \frac{\mathrm{cut}(V_1, V_1) + \mathrm{cut}(V_2, V_2)}{|E|} - \frac{\deg(V_1)^2 + \deg(V_2)^2}{\deg(V)^2 - \sum_{v \in V} \deg(v)^2} \;.$$

If $V_1 \cup V_2 = V$, then

$$\mathrm{cut}(V_1, V_1) + \mathrm{cut}(V_2, V_2) = |E| - \mathrm{cut}(V_1, V_2)$$

and

$$\deg(V_1)^2 + \deg(V_2)^2 = \deg(V)^2 - 2\deg(V_1)\deg(V_2) \;,$$

and thus maximizing $Q'(V_1, V_2)$ is equivalent to minimizing

$$\mathrm{cut}(V_1, V_2) - \frac{2\deg(V_1)\deg(V_2)}{\deg(V)^2 - \sum_{v \in V} \deg(v)^2} |E| \;.$$

The second term is precisely the *expected* cut of V_1 and V_2, as derived in Section 2.2. So if $V_1 \cup V_2 = V$, maximizing Newman's modularity is equivalent to minimizing the *difference* between the actual cut and the expected cut, while the edge-normalized cut is the *quotient* of the actual cut and the expected cut.

Our reason for preferring the edge-normalized cut over Newman's modularity is rather pragmatic: In the next section, the coupling of subgraphs will be related to their distance in layouts, and this is easier for a coupling measure that is nonnegative and takes the value 0 in the case of no coupling.

[5] This version of the measure differs slightly from an earlier version published in [41], where the second term is the squared fraction of edges that connect to nodes in V_i.

3 Energy Models for Graph Clustering

As representations of the cluster structure, graph layouts offer several benefits over the more common partitions of the set of nodes. They do not simply assign nodes to clusters, but can show how closely nodes are associated with their cluster, and how clearly clusters are separated; and they facilitate the comprehension of the clusters, because viewers naturally interpret closely positioned nodes as strongly related [8, 14].

This section introduces two energy models that correspond to the two unbiased clustering criteria of the previous section, and demonstrates the internal validity of their minimum energy layouts. Specifically, it is shown that the layouts group densely connected nodes and separate sparsely connected nodes, and that the Euclidean distances of groups reflect their coupling (as measured by the clustering criteria).

3.1 The LinLog Energy Models

The *node-repulsion LinLog energy* of a layout p is defined as

$$\mathrm{U}_{\mathrm{NodeLinLog}}(p) \;=\; \sum_{\{u,v\}\in E} ||p(u)-p(v)|| \;-\; \sum_{\{u,v\}\in V^{(2)}} \ln ||p(u)-p(v)|| \;.$$

To avoid infinite energies we assume that different nodes have different positions, which is no serious restriction because we are interested in layouts 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 with repulsion between edges. In our formalization, 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:

$$\mathrm{U}_{\mathrm{EdgeLinLog}}(p) \;=\; \sum_{\{u,v\}\in E} ||p(u)-p(v)|| - \sum_{\{u,v\}\in V^{(2)}} \deg(u)\deg(v) \ln ||p(u)-p(v)||.$$

The beauty of edge repulsion lies in its symmetry: Edges cause both attraction and repulsion. In other words, nodes that attract strongly also repulse strongly. More precisely, each node has consistently – in terms of attraction and repulsion – an influence on the layout proportional to its degree. (This can be visualized by setting the size of each node to its degree, as in the figures of this article.)

In a node-repulsion LinLog layout 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 have equivalent minima up to scaling.

3.2 Separation of Clusters

In a graph layout that reflects the cluster structure, nodes of the same dense subgraph are close to each other, and nodes of different sparsely connected subgraphs are clearly separated. This can be achieved by minimizing the distances between connected nodes (i.e. by minimizing the edge lengths), and at the same time maximizing the distances between all pairs of nodes.

The first theorem in this subsection states that *layouts with minimal node-repulsion LinLog energy minimize the ratio of the mean distance between connected nodes to the mean distance between all nodes.* The second theorem is a similar statement for edge-repulsion LinLog, with the difference that the distances of nodes in the denominator are weighted by their degrees. The analogy to the first theorem becomes more clear when each node v is considered as consisting of $\deg(v)$ *end nodes* of edges. Then the second theorem states that *layouts with minimal edge-repulsion LinLog energy minimize the ratio of the mean distance between connected end nodes to the mean distance between all end nodes.* In both theorems, the mean in the numerator is the arithmetic mean, while the mean in the denominator is the geometric mean, which penalizes very short distances more and rewards very large distances less than the arithmetic mean.

For a graph (V, E), a set $F \subseteq V^{(2)}$ of unordered node pairs, and a layout p, the arithmetic mean of the distances of F is defined as

$$\text{arithmean}(F,p) = \frac{1}{|F|} \sum\nolimits_{\{u,v\}\in F} ||p(v) - p(u)|| \ ,$$

the geometric mean of the distances of F is defined as

$$\text{geomean}(F,p) = \left(\prod\nolimits_{\{u,v\}\in F} ||p(v) - p(u)|| \right)^{1/|F|} ,$$

and the degree-weighted geometric mean of the distances of F is defined as

$$\text{geomean}'(F,p) = \left(\prod\nolimits_{\{u,v\}\in F} ||p(v) - p(u)||^{\deg(u)\deg(v)} \right)^{1/\sum_{\{u,v\}\in F} \deg(u)\deg(v)} .$$

Theorem 1 *Let $G = (V, E)$ be a connected graph, and let p^0 be a layout of G with minimum node-repulsion LinLog energy. Then p^0 is a layout of G that minimizes $\frac{\text{arithmean}(E,p)}{\text{geomean}(V^{(2)},p)}$.*

Proof: The basic idea is to fix the average edge length temporarily. This does not restrict generality, but only the scaling factor, and thus can be removed at the end of the proof. It permits transforming the minimization of energy into a minimization of the inverse geometric mean of the node distances.

Let the layout p^0 be a solution of the minimization problem:

$$\text{Minimize } \mathrm{U}_{\mathrm{NodeLinLog}}(p).$$

Let $c := \sum_{\{u,v\}\in E} ||p^0(u) - p^0(v)||$. Note that $c \geq 0$. Then p^0 is also a solution of

$$\text{Minimize } \mathrm{U}_{\mathrm{NodeLinLog}}(p) \text{ subject to } \sum\nolimits_{\{u,v\}\in E} ||p(u) - p(v)|| = c.$$

This is equivalent to

$$\text{Minimize } -\sum\nolimits_{\{u,v\}\in V^{(2)}} \ln ||p(u)-p(v)|| \text{ subj. to } \sum\nolimits_{\{u,v\}\in E} ||p(u)-p(v)|| = c.$$

Because $\exp\left(x/|V^{(2)}|\right)$ is monotonically increasing in x, p^0 is a solution of

$$\text{Min. } \exp\left(-\sum\nolimits_{\{u,v\}\in V^{(2)}} \frac{\ln ||p(u)-p(v)||}{|V^{(2)}|}\right) \text{subj. to} \sum\nolimits_{\{u,v\}\in E} ||p(u)-p(v)|| = c.$$

This is equivalent to

$$\text{Minimize } \frac{1}{\text{geomean}(V^{(2)}, p)} \text{ subject to } \text{arithmean}(E, p) = \frac{c}{|E|}.$$

($|E| > 0$ because we only consider connected graphs with at least two nodes.) Because $\frac{c}{|E|}$ is nonnegative, p^0 is also a solution of

$$\text{Minimize } \frac{\text{arithmean}(E, p)}{\text{geomean}(V^{(2)}, p)} \text{ subject to } \text{arithmean}(E, p) = \frac{c}{|E|}. \qquad (1)$$

Assume that there is a layout q^0 of G with $\frac{\text{arithmean}(E,q^0)}{\text{geomean}(V^{(2)},q^0)} < \frac{\text{arithmean}(E,p^0)}{\text{geomean}(V^{(2)},p^0)}$. Because $\text{geomean}(V^{(2)}, q^0) > 0$, no two different nodes in q^0 have the same position, and thus $\text{arithmean}(E, q^0) > 0$. The layout $q^1 := \frac{c}{|E|\,\text{arithmean}(E,q^0)} q^0$ has $\text{arithmean}(E, q^1) = \frac{c}{|E|}$ and $\frac{\text{arithmean}(E,q^1)}{\text{geomean}(V^{(2)},q^1)} = \frac{\text{arithmean}(E,q^0)}{\text{geomean}(V^{(2)},q^0)} < \frac{\text{arithmean}(E,p^0)}{\text{geomean}(V^{(2)},p^0)}$. This contradicts statement (1); thus the assumption is wrong, and p^0 is also a solution of

$$\text{Minimize } \frac{\text{arithmean}(E, p)}{\text{geomean}(V^{(2)}, p)}.$$

□

Theorem 2 *Let $G = (V, E)$ be a connected graph, and let p^0 be a layout of G with minimum edge-repulsion LinLog energy. Then p^0 is a layout of G that minimizes $\frac{\text{arithmean}(E,p)}{\text{geomean}'(V^{(2)},p)}$.*

Proof: Similar to the proof of Theorem 1. □

3.3 Interpretable Distances between Clusters

Ideally, a graph layout that reflects the cluster structure not only shows clusters, but also the coupling between clusters. This subsection shows that *the distance of two dense, sparsely connected clusters approximates their inverse*

node-normalized cut in layouts with minimal node-repulsion LinLog energy, and approximates their inverse edge-normalized cut in layouts with minimal edge-repulsion LinLog energy.

A formal proof is given for an idealization of this statement. Let (V, E) be a graph, and let (V_1, V_2) be a bipartition of the set of nodes V into two dense, sparsely connected subgraphs. In a minimum LinLog energy layout p, the distances *within* V_1 and *within* V_2 should be much smaller than the distance *between* V_1 and V_2. In the theorems, this situation is approximated by assuming that all nodes in V_1 have the same position and all nodes in V_2 have the same position in p. For this simplified situation it can be shown that the distance between V_1 and V_2 equals the inverse normalized cut between V_1 and V_2. A similar theorem for less restricted layouts is proved in [42].

Theorem 3 *Let $G = (V, E)$ be a connected graph, and let (V_1, V_2) be a bipartition of its set of nodes. Let P be the set of layouts of G that assign the same position to all nodes in V_1, and the same position to all nodes in V_2. Let p^0 be a layout in P with minimum node-repulsion LinLog energy. Then the distance of V_1 and V_2 in p^0 is $\frac{1}{\mathrm{ncut}(V_1,V_2)}$.*

Proof: The basic idea is to express the node-repulsion LinLog energy as a function of the distance of the two sets of nodes, and to exploit the fact that the minimum energy layout is a minimum of this function.

Let p^0 be a layout in P with minimum node-repulsion LinLog energy, and let d^0 be the distance of V_1 and V_2 in p^0. Because the distances between all nodes in V_1 and between all nodes in V_2 are equal (namely, 0) for all layouts in P, the distance d^0 is a minimum of

$$U(d) = \mathrm{cut}(V_1, V_2)\, d - |V_1||V_2| \ln d \ .$$

The derivative of this function is 0 at its minimum d^0.

$$\begin{aligned} 0 \;=\; U'(d^0) \;&=\; \mathrm{cut}(V_1, V_2) - |V_1| \cdot |V_2| / d^0 \\ d^0 \;&=\; \frac{|V_1| \cdot |V_2|}{\mathrm{cut}(V_1, V_2)} = \frac{1}{\mathrm{ncut}(V_1, V_2)} \end{aligned}$$

□

Theorem 4 *Let G, V_1, V_2, P be defined as in Theorem 3. Let p^0 be a layout in P with minimum edge-repulsion LinLog energy. Then the distance of V_1 and V_2 in p^0 is $\frac{1}{\mathrm{ecut}(V_1,V_2)}$.*

Proof: Similar to the proof of Theorem 3. □

The simple technique used in the proofs allows a quick assessment of the clustering properties not only of the LinLog energy models, but also of other energy models that are based on pairwise attraction and repulsion.

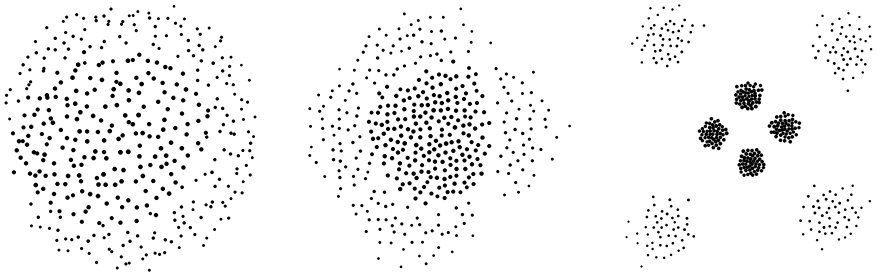
3.4 Example

Figure 2 shows six layouts of a pseudo-random graph with eight clusters of 50 nodes. The probability of an edge $\{u, v\}$ is

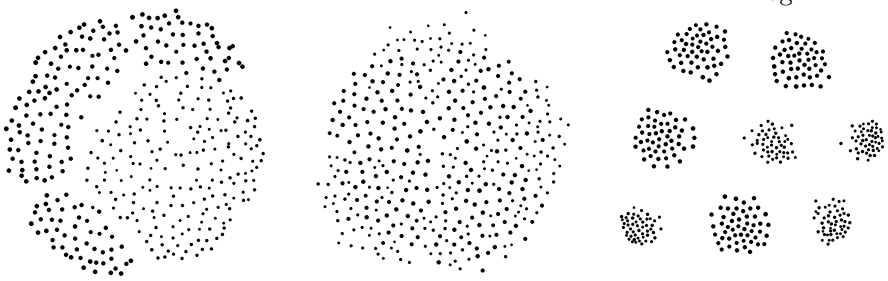
- 1 if u and v belong to the same of the first four clusters,
- 0.5 if u and v belong to the same of the second four clusters,
- 0.2 if u and v belong to different of the first four clusters,
- 0.05 if u and v belong to different of the second four clusters, and
- 0.1 if u belongs to one of the first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

In total, the graph has 400 nodes, 14 738 edges between nodes of the same cluster, and 7 770 edges between nodes of different clusters. Nodes of the first four clusters generally have larger degrees than nodes of the second four clusters, and thus have larger representations in Figure 2.

The layouts of the LinLog models clearly show the clusters; the other four layouts will be discussed in the next subsection. The node-repulsion LinLog layout places the first four clusters more closely than the second four clusters, which reflects the fact that node-normalized cuts between the first four clusters are higher than between the second four clusters. In the edge-repulsion LinLog layout the distances between all clusters are similar, reflecting the fact that the edge-normalized cuts between all pairs of clusters are similar.



(a) Kamada-Kawai (c) Fruchterman-Reingold (e) Node-repulsion LinLog



(b) Kamada-Kawai, adapted by Gansner et al. (d) Edge-repulsion Fruchterman-Reingold (f) Edge-repulsion LinLog

Figure 2: Pseudo-random graph

3.5 Related Work

Graph Clustering with Energy Models Most existing force and energy models have not been designed to find clusters, but to produce readable box-and-line visualizations. For example, the classic models of Eades [16], Fruchterman and Reingold [20], and Davidson and Harel [13] enforce uniform (or other given) edge lengths, which often prevents the separation of clusters. Because they are (like LinLog) based on the pairwise attraction and repulsion of nodes, the analysis technique of Section 3.3 is applicable, and it can be proved that the coupling and the Euclidean distance of subgraphs are only weakly related in their layouts. The model of Kamada and Kawai [30] enforces Euclidean distances to approximate graph-theoretic distances, which are only weakly related to the density (and thus to the cluster structure). The example layouts for the Kamada-Kawai model and the Fruchterman-Reingold model in Figures 2(a) and 2(c) indeed fail to show the clusters.

Graph Clustering by Minimizing Distance Ratios Theorem 1 states that a layout that minimizes the node-repulsion LinLog energy also minimizes the ratio $\frac{\text{arithmean}(E,p)}{\text{geomean}(V^{(2)},p)}$. It was introduced as a characterization of LinLog layouts, but may also be seen as basis for applying force-directed layout algorithms to minimize distance ratios, and thus to reveal the cluster structure. Earlier works have minimized similar distance ratios to find clusters. In particular, some approximation algorithms for graph clustering problems (e.g. [36, 4]) derive partitions from layouts that minimize the ratio $\frac{\text{arithmean}(E,p)}{\text{arithmean}(V^{(2)},p)}$. However, these layouts are not suitable for human viewers (e.g. many nodes are placed at the same position), and are not even computed in Euclidean space.

Edge Repulsion in Energy-Based Graph Drawing Several works introduce forces that are similar to the edge repulsion in the LinLog energy model, but differ from this work in two respects: First, they do not suggest to replace repulsion between all pairs of nodes with repulsion between all pairs of edges. Second, the forces are intended to improve the conformance to specific aesthetic criteria, and do not enable interpretations with respect to the cluster structure.

Coleman and Parker [11] propose a repulsive force between edge midpoints, and Davidson and Harel [13] and Bertault [6] introduce a repulsive force between edges and nodes, all to avoid edges that are very close or cross each other. Cruz and Twarog [12] suggest (without giving details) that for 3D layouts, the latter force can be replaced with a repulsive force between non-adjacent edges. Lin and Yen [35] use a repulsive force only between adjacent edges, mainly to improve angular resolution. Frick, Ludwig and Mehldau [19, Section 4.3] scale the *attractive* force acting on each node v with a factor $\frac{1}{\deg(v)(1+\deg(v)/2)}$ (without justifying the choice of this factor), to distribute the nodes more uniformly. Gansner, Koren and North [22, Section 3] adapt the Kamada-Kawai energy model (and similar models) for the same purpose, by increasing the desired edge length between high-degree nodes.

Edge Repulsion in Spectral Graph Drawing According to Theorems 1 and 2, layouts with minimal LinLog energy minimize certain ratios of mean edge lengths to mean all-pairs distances. Spectral graph layouts minimize similar distance ratios. Also, we distinguished between a node-repulsion and an edge-repulsion version of the LinLog energy model. A similar distinction exists in spectral graph layout.

Spectral graph layout methods compute layouts of graphs from eigenvectors of related matrices, most commonly the Laplacian. The *adjacency matrix* A of a graph $G = (V, E)$ is a symmetric $|V| \times |V|$ matrix with

$$A(u,v) = \begin{cases} 0 & \text{if } \{u,v\} \notin E \\ 1 & \text{if } \{u,v\} \in E \end{cases} .$$

The *degree matrix* D of G is a $|V| \times |V|$ diagonal matrix with $D(v,v) = \deg(v)$. The *Laplacian* L of G is defined as $L = D - A$.

For connected graphs, all eigenvalues of the Laplacian are real, the smallest eigenvalue is 0 (with associated eigenvector $(1, 1, ..., 1)^T$), and all other eigenvalues are positive. The eigenvector corresponding to the second smallest eigenvalue is called the *Fiedler vector*.

Theorem 5 (Fiedler [17]) *The Fiedler vector of a graph (V, E) minimizes*

$$\frac{\sum_{\{u,v\} \in E} (x(u) - x(v))^2}{\sum_{\{u,v\} \in V^{(2)}} (x(u) - x(v))^2}$$

over all vectors $x \in \mathbb{R}^{|V|}$ that are non-constant (i.e. have at least two different entries).

This property justifies the use of the Fiedler vector not only as node coordinate vector in one-dimensional graph layouts (pioneered by Hall [26]), but also for deriving graph clusters e.g. by simple thresholding (pioneered by Donath and Hoffman [15]). The next eigenvectors of the Laplacian have similar properties and can be used as additional coordinates in higher-dimensional layouts (see [26, 32] for details).

More recently, solutions of the generalized Laplacian eigensystem $Ly = \mu Dy$ have received considerable attention [10, 45, 32]. We denote the generalized eigenvector corresponding to the second smallest generalized eigenvalue as the *degree-normalized Fiedler vector*.

Theorem 6 (Chung [10, Chapter 1.2], similarly Koren [32, Section 4]) *The degree-normalized Fiedler vector of a graph (V, E) minimizes*

$$\frac{\sum_{\{u,v\} \in E} (x(u) - x(v))^2}{\sum_{\{u,v\} \in V^{(2)}} \deg(u) \deg(v) (x(u) - x(v))^2}$$

over all non-constant vectors $x \in \mathbb{R}^{|V|}$.

Comparing Theorems 1 and 2 with Theorems 5 and 6 shows a striking analogy between LinLog layouts and spectral layouts: LinLog layouts minimize the ratio of the arithmetic mean of edge lengths to the geometric mean of all-pairs distances. Spectral layouts minimize the ratio of the sum (or equivalently, arithmetic mean) of squared edge lengths to the sum (or arithmetic mean) of squared all-pairs distances. And for both LinLog and spectral methods, there exists a variant (with edge repulsion and degree normalization, respectively) where the distances in the denominator are weighted by the node degrees.

From the applications perspective, the benefits of edge repulsion for drawing and clustering graphs with nonuniform degrees are available with both force-directed and spectral methods. However, spectral layouts do not reflect coupling as directly as LinLog layouts (see Theorem 3 and 4), and tend to place many nodes at the same position, which impairs readability. This latter property can be easily seen from the minimized distance ratios, and has been observed empirically e.g. in [25, 32]. On the other hand, efficient algorithms for computing globally optimal layouts exist only for spectral methods.

3.6 Extensions

Classes of Energy Models As discussed in Section 3.5, no single energy model can both isolate clusters and enforce uniform edge lengths, but classes of energy models may provide users with a choice. An example for such a class is r-PolyLog. For all $r \in \mathbb{R}$ with $r > 0$, the *node-repulsion r-PolyLog energy* of a layout p is defined as

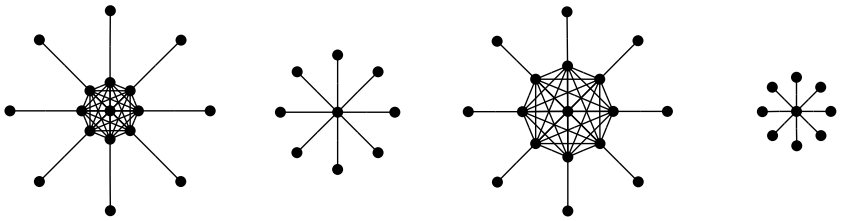
$$\mathrm{U}_{r\text{-NodePolyLog}}(p) \;=\; \sum_{\{u,v\}\in E} \frac{1}{r}||p(u)-p(v)||^r \;-\; \sum_{\{u,v\}\in V^{(2)}} \ln||p(u)-p(v)|| \;,$$

and the *edge-repulsion r-PolyLog energy* is defined similarly.

This class of energy models contains two models that were already mentioned: The 1-PolyLog energy model is the LinLog model, and the 3-PolyLog energy model is the Fruchterman-Reingold model [20] (which is usually expressed as a force model). The class contains energy models that isolate clusters ($r \to 0$), energy models that enforce uniform edge lengths ($r \to \infty$), and many compromises between both extremes ($0 < r < \infty$).

Edge Repulsion for Conventional Energy Models In many force and energy models, including those of Eades [16] and Fruchterman and Reingold [20], adjacent nodes attract and all pairs of nodes repulse. Like node-repulsion LinLog, these models tend to draw dense subgraphs too small (because attraction dominates repulsion) and sparse subgraphs too large.

Figure 3(a) shows two examples for the Fruchterman-Reingold model: The eight central nodes of the left graph are connected by many edges, but use only a small part of the area. Much area is wasted by the unnecessarily long edges to the eight peripheral nodes. The (sparse) right graph is drawn much larger than the dense part of the left graph, although it contains much fewer edges.



(a) (Node-repulsion) Fruchterman-Reingold

(b) Edge-repulsion Fruchterman-Reingold

Figure 3: Two small graphs

Like for LinLog, replacing node repulsion with edge repulsion improves the balance between attraction and repulsion, because both are caused by the edges. Figure 3(b) shows that this leads to a more uniform information density and thus better readability. A full evaluation of these readability improvements is beyond the scope of this article, which focuses on interpretability.

Weighted Graphs A *weighted graph* $G = (V, E, w)$ consists of a finite set V of nodes, a finite set E of edges with $E \subseteq V^{(2)}$, and a function $w : V \cup E \to \mathbb{R}_+$ that assigns a positive weight to each node and each edge. The *degree* $\deg(v)$ of a node v is the sum of the weights of its incident edges $\sum_{e \in E : v \in e} w(e)$.

The node-repulsion LinLog energy of a layout p of a weighted graph (V, E, w) is

$$\sum_{\{u,v\} \in E} w(\{u,v\}) ||p(u) - p(v)|| \; - \sum_{\{u,v\} \in V^{(2)}} w(u) w(v) \ln ||p(u) - p(v)||,$$

and the edge-repulsion LinLog energy is

$$\sum_{\{u,v\} \in E} w(\{u,v\}) ||p(u) - p(v)|| \; - \sum_{\{u,v\} \in V^{(2)}} \deg(u) \deg(v) \ln ||p(u) - p(v)||.$$

So the node-repulsion LinLog energy of a layout equals the edge-repulsion LinLog energy if the weight of each node is its degree.

Unconnected Graphs If a graph has more than one connected component, the distances of the connected components approach infinity in layouts with minimum LinLog energy. This can be avoided by adding a *gravitational energy* that attracts each node to the barycenter of the layout [19].

For a weighted graph $G = (V, E, w)$, a layout p of G with the barycenter $b(p) := \frac{\sum_{v \in V} w(v) p(v)}{\sum_{v \in V} w(v)}$, and a small constant g that determines the distances of the components, the gravitational energy $\sum_{v \in V} g w(v) ||b(p) - p(v)||$ can be added to the node-repulsion LinLog energy of p. For edge-repulsion LinLog, the weight $w(v)$ should be replaced with the degree $\deg(v)$.

A special case of a connected component is a node with the degree 0. The edge-repulsion energy of such a node is 0, independent of its position. Appropriate positions for such nodes can be determined by treating them as if they had a small positive degree.

Graphs with Clusterings A *clustering* of a graph (V, E) is a partition of its set of nodes V into nonempty subsets. Sometimes a layout should group the nodes according to a *given* clustering, as opposed to a *good* clustering with respect to one of the clustering criteria in Section 2. This can be achieved by generalizing the gravitational energy (introduced in the previous paragraph) to attract each node to the barycenter of its subset (see [43] for details).

4 Real-World Examples

This section discusses example layouts of the two LinLog energy models and, for comparison, of the widely used Fruchterman-Reingold force model [20]. It provides evidence for the external validity of layouts with small edge-repulsion LinLog energy, by showing that the grouping of the nodes in these layouts conforms with authoritative groupings.

The layouts are shown in Figures 4 to 7. The degree of each node is proportional to the area of its representing circle. (A certain minimum area ensures visibility.) The edges are not represented, because they are not relevant to external validity, and because they could not be discerned due to their relatively large density. Some layouts were rotated manually. (Rotation does not change the energy.) The graph data, a tool for computing the layouts, and VRML files of the larger layouts are available on the supplementary web page[6]. Unlike static pictures, the VRML files enable zooming and the selective display of node labels, which is particularly useful for visualizations of large graphs.

For all four graphs, the Fruchterman-Reingold model (Subfigures (a)) and the node-repulsion LinLog model (Subfigures (b)) tend to place nodes with high degree in the center, and nodes with low degree near the borders. Thus the positions of the nodes in the node-repulsion layouts mainly reflect their degree, and only the edge-repulsion LinLog layouts will be discussed in more detail.

Event Participation (Figure 4) The graph represents the participation of 18 women in 14 informal social events. Each woman and each event is modeled by a node, and each participation is modeled by an edge. Freeman [18] performed a meta-analysis of 21 earlier studies that assigned the women to groups. Applying consensus analysis to combine the results of these studies, he obtained a decomposition into two groups, with the first group containing Brenda, Charlotte, Eleanor, Evelyn, Frances, Laura, Pearl, Ruth, and Theresa, and the other group containing the remaining nine women. The individual studies show considerable disagreement about the assignment of Pearl, Olivia, and Flora. Some

[6] www-sst.informatik.tu-cottbus.de/GD/erlinlog.html

studies assign Pearl to the first group, some to the second group, others to no group or both groups. Olivia and Flora are often assigned to the second group, but sometimes considered as a separate group, or assigned to no group at all.

The edge-repulsion LinLog layout shows the two main groups of women, but also shows that Pearl is rather between these groups. The disagreement between edge-repulsion LinLog and node-repulsion LinLog mirrors the disagreement of previous studies: While the earlier assigns Olivia and Flora to the second group because they exclusively attend events of this group, the latter separates them from the second group because they attend few such events.

Airline Routing (Figure 5) The nodes of this graph represent US airports, and the (unweighted) edges represent direct flights. The probability that two airports are connected by a direct flight is strongly related to their geographical distance: Most direct flights are relatively short, and only few large hub airports are connected by direct long-range flights.

The distances in the edge-repulsion LinLog layout resemble the relative geographical distances of the airports remarkably closely, given that the graph does not contain any explicit information about geographical distances.

World Trade (Figure 6) The nodes of this graph represent countries, and the edge weights specify the trade volume between each pair of countries. The main factor that determines the transaction costs and thus the intensity of trade between two countries is their geographical distance.

The edge-repulsion LinLog layout reflects the relative geographical distances on all scales. Globally, it separates the three large economic areas of the world, namely America, East Asia and Australia, and Europe. Locally, it groups, for example, the North European countries (Norway, Sweden, Finland, Denmark), and pairs Spain and Portugal, Australia and New Zealand, and China and Hong Kong.

The degrees of the nodes in the world trade graph are extremely non-uniform, because the total trade of the largest and the smallest countries differs by more than three orders of magnitude. As a consequence, the distances of economically large countries in the node-repulsion layouts are very small compared to the distances of small countries, and the difference between edge-repulsion layouts and node-repulsion layouts is huge.

Dictionary (Figure 7) The nodes represent terms in the Online Dictionary of Library and Information Science (ODLIS), and the edges represent hyperlinks. A hyperlink between two terms exists if one term is used to describe the meaning of the other term, and thus connects semantically related terms.

The edge-repulsion LinLog layout indeed groups semantically related terms, which is better reflected in the VRML file on the supplementary web page than in Figure 7(c). Such a grouping is useful, for example, for discovering the global topic areas (like publishing, printing, information technology, etc.), for

identifying entry points for the exploration of topics, or for finding semantically related terms even if they are not explicitly linked.

Coupling of Software Artifacts [7] In large software systems, the individual software artifacts (e.g. files or classes) are hierarchically organized into subsystems. Artifacts that are frequently changed together should belong to the same subsystem, because changes across subsystem boundaries tend to be more expensive and error-prone. Thus grouping artifacts with respect to common changes helps to propose new subsystem hierarchies, and to evaluate and improve existing subsystem hierarchies. Beyer and Noack have applied the edge-repulsion LinLog energy model to identify such groups of artifacts [7]. Example layouts and the tool CCVisu with particular support for this application can be found at the web page `mtc.epfl.ch/~beyer/co-change/`.

5 Summary

The main contributions of this article are

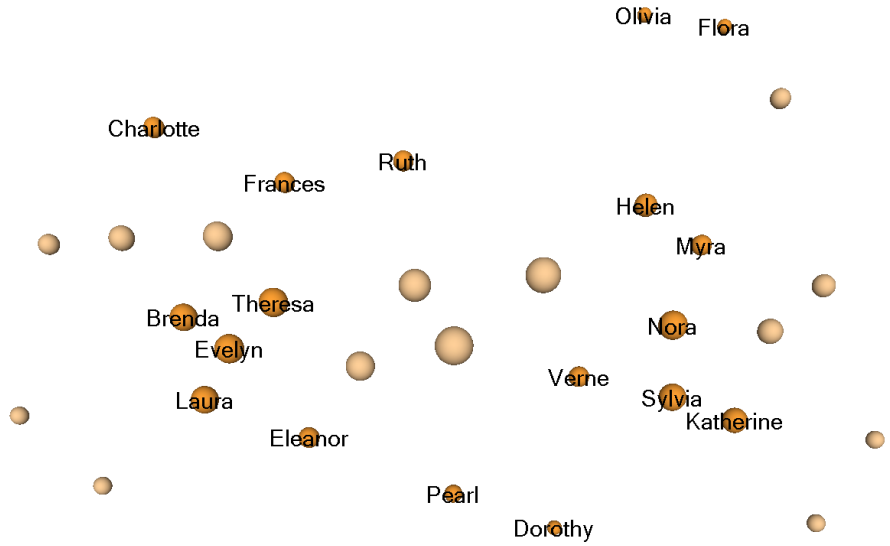
- the LinLog energy models, whose minimum energy layouts reflect the cluster structure of graphs with respect to two well-defined clustering criteria.
- edge repulsion in energy models, which avoids or reduces the bias towards grouping nodes with high degree when used instead of or in addition to node repulsion.

Some techniques from the development and evaluation of these results may also be applicable in other contexts, in particular

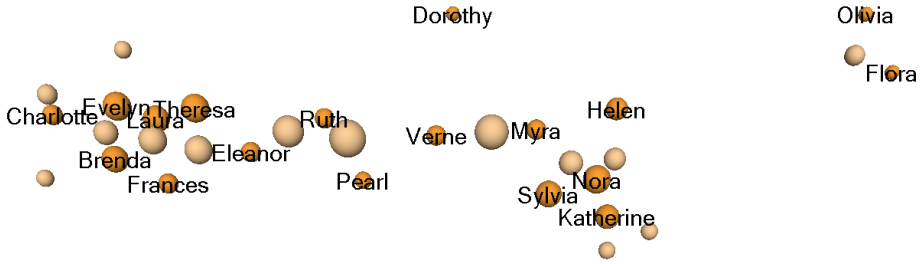
- the identification and elimination of biases in clustering criteria in Section 2, and
- the analysis of minimum energy layouts with respect to optimized ratios of mean edge lengths to mean node distances in Section 3.2, and with respect to the correspondence of distances to clustering criteria in Section 3.3.

Main results that connect the proposed clustering criteria and energy models with previous work are

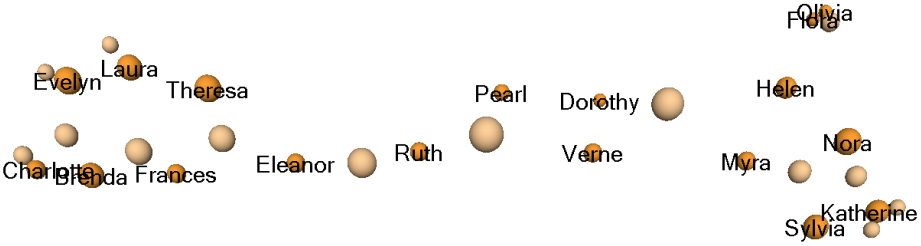
- formal relationships between Shi and Malik's normalized cut, Newman's modularity, and the edge-normalized cut (which is the clustering criterion associated with the edge-repulsion LinLog energy model).
- the analogy of the distance ratios minimized by the node-repulsion and the edge-repulsion version of the LinLog energy model, and the distance ratios minimized by the unnormalized and the degree-normalized version of spectral graph drawing.
- the class r-PolyLog of energy models which contains the LinLog energy models, the Fruchterman-Reingold force model, and many other compromises between isolating clusters and enforcing uniform edge lengths.



(a) Fruchterman-Reingold model



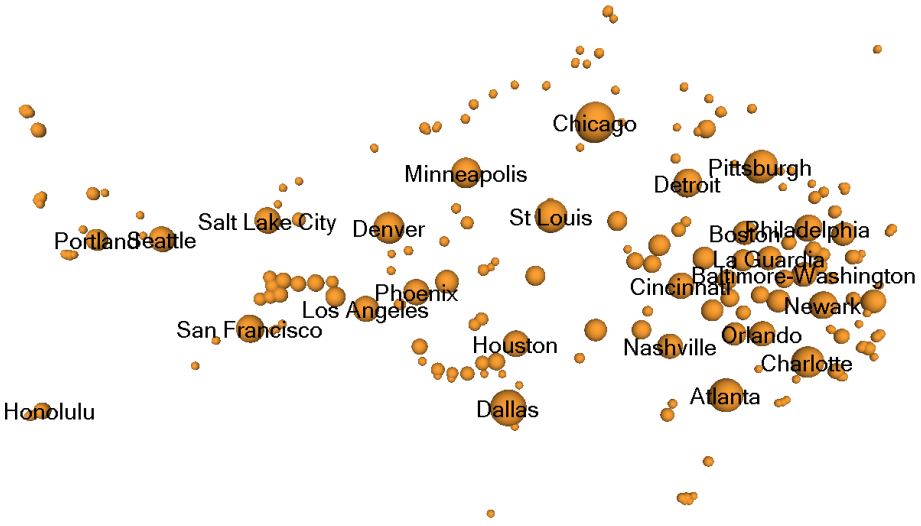
(b) Node-repulsion LinLog model



(c) Edge-repulsion LinLog model

Figure 4: Women participating in social events (18 women, 14 events, 89 edges). The lighter-colored nodes correspond to the events.
Data source: [18, Figure 1]

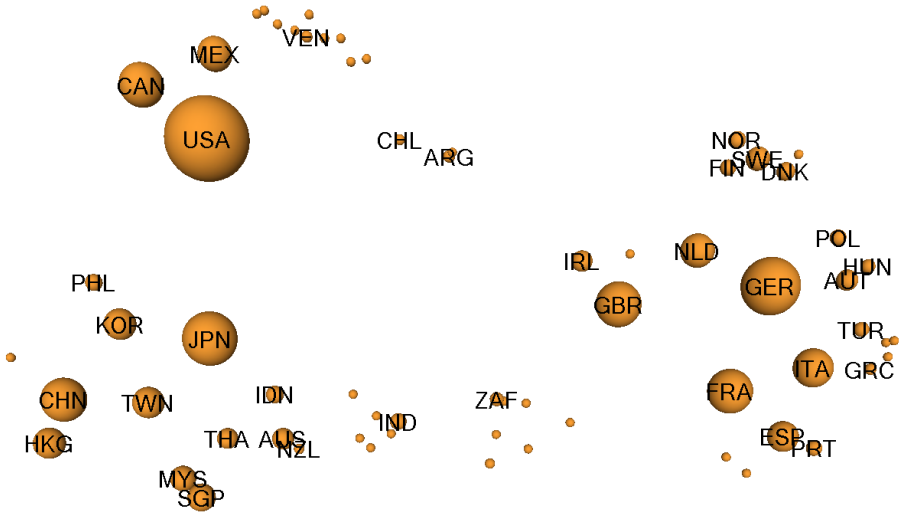
(a) Fruchterman-Reingold model (b) Node-repulsion LinLog model



(c) Edge-repulsion LinLog model

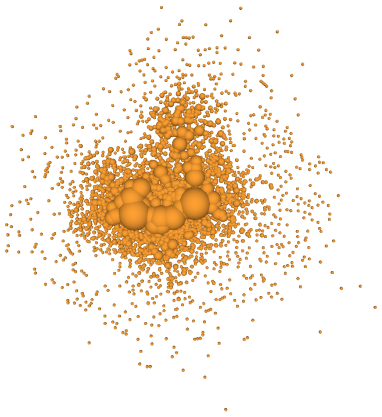
Figure 5: Direct flights between US airports (332 nodes, 2126 edges). Some distant airports in Alaska and the South Sea (e.g. Guam) are omitted to improve readability.
Data source: Pajek project (vlado.fmf.uni-lj.si/pub/networks/data/)

(a) Fruchterman-Reingold model (b) Node-repulsion LinLog model

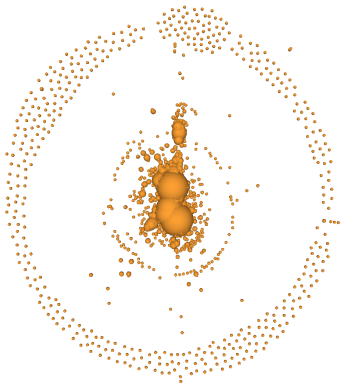


(c) Edge-repulsion LinLog model

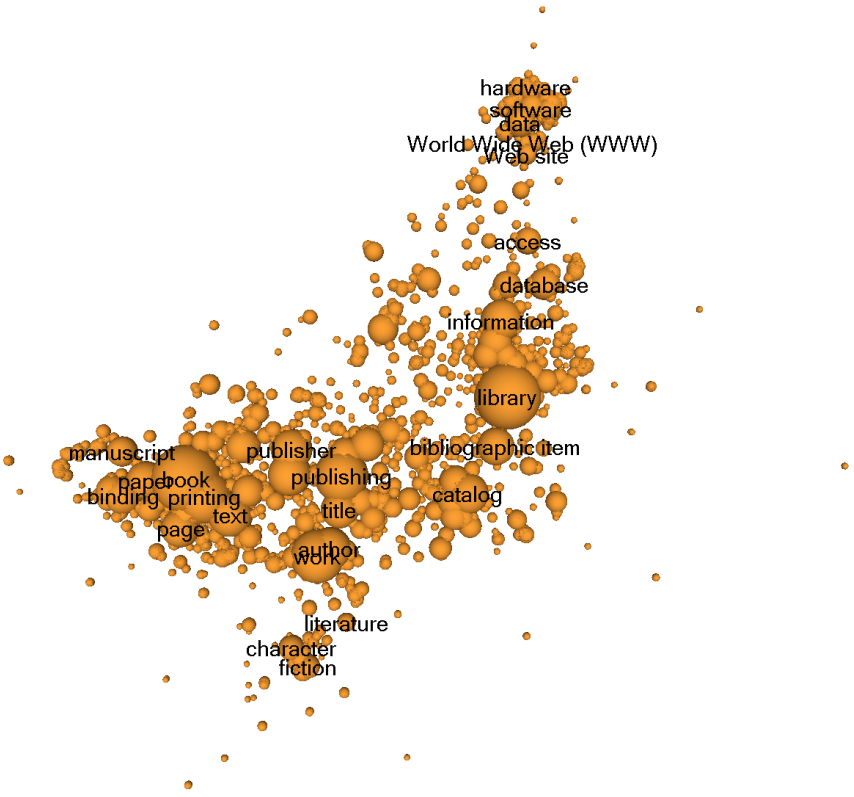
Figure 6: Trade between 66 countries. Edges are weighted with the trade volume.
Data source: Bilateral trade data for the year 1999 from the World Bank (www.worldbank.org)



(a) Fruchterman-Reingold model



(b) Node-repulsion LinLog model



(c) Edge-repulsion LinLog model

Figure 7: Hyperlinks between terms in the Online Dictionary for Library and Information Science ODLIS (2896 nodes, 18238 edges).
Data source: Pajek project (vlado.fmf.uni-lj.si/pub/networks/data/)

References

[1] R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47–97, 2002.

[2] C. J. Alpert and A. B. Kahng. Recent directions in netlist partitioning: A survey. *Integration, the VLSI Journal*, 19(1-2):1–81, 1995.

[3] S. Arora, S. Rao, and U. V. Vazirani. Expander flows, geometric embeddings and graph partitioning. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing (STOC 2004)*, pages 222–231. ACM, 2004.

[4] Y. Aumann and Y. Rabani. An $O(\log k)$ approximate min-cut max-flow theorem and approximation algorithm. *SIAM Journal on Computing*, 27(1):291–301, 1998.

[5] J. Barnes and P. Hut. A hierarchical $O(N \log N)$ force-calculation algorithm. *Nature*, 324:446–449, 1986.

[6] F. Bertault. A force-directed algorithm that preserves edge crossing properties. In J. Kratochvíl, editor, *Proceedings of the 7th International Symposium on Graph Drawing (GD 1999)*, LNCS 1731, pages 351–358, Berlin, 1999. Springer-Verlag.

[7] D. Beyer and A. Noack. Clustering software artifacts based on frequent common changes. In *Proceedings of the 13th International Workshop on Program Comprehension (IWPC 2005)*, pages 259–268. IEEE Computer Society, 2005.

[8] J. Blythe, C. McGrath, and D. Krackhardt. The effect of graph layout on inference from social network data. In F.-J. Brandenburg, editor, *Proceedings of the Symposium on Graph Drawing (GD 1995)*, LNCS 1027, pages 40–51, Berlin, 1996. Springer-Verlag.

[9] F.-J. Brandenburg, M. Himsolt, and C. 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.

[10] F. R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, Providence, RI, 1997.

[11] M. K. Coleman and D. S. Parker. Aesthetics-based graph layout for human consumption. *Software – Practice & Experience*, 26(12):1415–1438, 1996.

[12] I. F. Cruz and J. P. Twarog. 3D graph drawing with simulated annealing. In F.-J. Brandenburg, editor, *Proceedings of the Symposium on Graph Drawing (GD 1995)*, LNCS 1027, pages 162–165, Berlin, 1996. Springer-Verlag.

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

[14] E. Dengler and W. Cowan. Human perception of laid-out graphs. In S. H. Whitesides, editor, *Proceedings of the 6th International Symposium on Graph Drawing (GD 1998)*, LNCS 1547, pages 441–443, Berlin, 1998. Springer-Verlag.

[15] W. E. Donath and A. J. Hoffman. Lower bounds for the partitioning of graphs. *IBM Journal of Research and Development*, 17:420–425, 1973.

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

[17] M. Fiedler. A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. *Czechoslovak Mathematical Journal*, 25(100):619–633, 1975.

[18] L. C. Freeman. Finding social groups: A meta-analysis of the southern women data. In R. Breiger, K. Carley, and P. Pattison, editors, *Dynamic Social Network Modeling and Analysis*, pages 37–77. The National Academies Press, Washington, DC, 2003.

[19] A. Frick, A. Ludwig, and H. Mehldau. A fast adaptive layout algorithm for undirected graphs. In R. Tamassia and I. G. Tollis, editors, *Proceedings of the DIMACS International Workshop on Graph Drawing (GD 1994)*, LNCS 894, pages 388–403, Berlin, 1995. Springer-Verlag.

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

[21] P. Gajer, M. T. Goodrich, and S. G. Kobourov. A multi-dimensional approach to force-directed layouts of large graphs. *Computational Geometry*, 29(1):3–18, 2004.

[22] E. R. Gansner, Y. Koren, and S. North. Graph drawing by stress majorization. In J. Pach, editor, *Proceedings of the 12th International Symposium on Graph Drawing (GD 2004)*, LNCS 3383, pages 239–250, Berlin, 2005. Springer-Verlag.

[23] R. E. Gomory and T. C. Hu. Multi-terminal network flows. *Journal of SIAM*, 9(4):551–570, 1961.

[24] S. Hachul and M. Jünger. Drawing large graphs with a potential-field-based multilevel algorithm. In J. Pach, editor, *Proceedings of the 12th International Symposium on Graph Drawing (GD 2004)*, LNCS 3383, pages 285–295, Berlin, 2004. Springer-Verlag.

[25] S. Hachul and M. Jünger. An experimental comparison of fast algorithms for drawing general large graphs. In P. Healy and N. S. Nikolov, editors, *Proceedings of the 13th International Symposium on Graph Drawing (GD 2005)*, LNCS 3843, pages 235–250, Berlin, 2006. Springer-Verlag.

[26] K. M. Hall. An r-dimensional quadratic placement algorithm. *Management Science*, 17(3):219–229, 1970.

[27] D. Harel and Y. Koren. A fast multi-scale method for drawing large graphs. *Journal of Graph Algorithms and Applications*, 6(3):179–202, 2002.

[28] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, Englewood Cliffs, NJ, 1988.

[29] V. Kaibel. On the expansion of graphs of 0/1-polytopes. In M. Grötschel, editor, *The Sharpest Cut: The Impact of Manfred Padberg and His Work*, pages 199–216. SIAM, 2004.

[30] T. Kamada and S. Kawai. An algorithm for drawing general undirected graphs. *Information Processing Letters*, 31(1):7–15, 1989.

[31] R. Kannan, S. Vempala, and A. Vetta. On clusterings: Good, bad and spectral. *Journal of the ACM*, 51(3):497–515, 2004.

[32] Y. Koren. Drawing graphs by eigenvectors: Theory and practice. *Computers and Mathematics with Applications*, 49(11-12):1867–1888, 2005.

[33] T. Leighton and S. 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. IEEE Computer Society, 1988.

[34] T. Leighton and S. Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *Journal of the ACM*, 46(6):787–832, 1999.

[35] C.-C. Lin and H.-C. Yen. A new force-directed graph drawing method based on edge-edge repulsion. In *Proceedings of the 9th International Conference on Information Visualisation (IV 2005)*, pages 329–334. IEEE Computer Society, 2005.

[36] N. Linial, E. London, and Y. Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15(2):215–245, 1995.

[37] 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.

[38] D. W. Matula and F. Shahrokhi. Sparsest cuts and bottlenecks in graphs. *Discrete Applied Mathematics*, 27(1-2):113–123, 1990.

[39] M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45(2):167–256, 2003.

[40] M. E. J. Newman. Analysis of weighted networks. *Physical Review E*, 70:056131, 2004.

[41] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69:026113, 2004.

[42] A. 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.

[43] A. Noack and C. Lewerentz. A space of layout styles for hierarchical graph models of software systems. In *Proceedings of the 2nd ACM Symposium on Software Visualization (SoftVis 2005)*, pages 155–164. ACM, 2005.

[44] A. J. Quigley and P. 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.

[45] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[46] J. Síma and S. E. Schaeffer. On the NP-completeness of some graph cluster measures. In *Proceedings of the 32nd Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM 2006)*, LNCS 3831, pages 530–537, Berlin, 2006. Springer-Verlag.

[47] H. A. Simon. The architecture of complexity. *Proceedings of the American Philosophical Society*, 106(6):467–482, 1962.

[48] S. H. Strogatz. Exploring complex networks. *Nature*, 410:268–276, 2001.

[49] K. Sugiyama and K. Misue. Graph drawing by the magnetic spring model. *Journal of Visual Languages and Computing*, 6(3):217–231, 1995.

[50] D. Tunkelang. JIGGLE: Java interactive graph layout environment. In S. H. Whitesides, editor, *Proceedings of the 6th International Symposium on Graph Drawing (GD 1998)*, LNCS 1547, pages 413–422, Berlin, 1998. Springer-Verlag.

[51] C. Walshaw. A multilevel algorithm for force-directed graph-drawing. *Journal of Graph Algorithms and Applications*, 7(3):253–285, 2003.