

Drawing Power Law Graphs

Reid Andersen, Fan Chung*, and Lincoln Lu

University of California, San Diego

Abstract. We present methods for drawing graphs that arise in various information networks. It has been noted that many realistic graphs have a power law degree distribution and exhibit the small world phenomenon. Our methods are influenced by recent developments in the modeling of such graphs.

1 Introduction

Several research groups have observed that many networks, including Internet graphs, call graphs and social networks, have a *power law* degree distribution, where the fraction of nodes with degree k is proportional to $k^{-\beta}$ for some positive exponent β [8]. Many networks also exhibit a so-called “small world phenomenon” consisting of two distinct properties — small average distance between nodes, and a clustering effect where two nodes sharing a common neighbor are more likely to be adjacent. It was shown in [2] that a random power law graph has small average distance and small diameter. However, random power law graphs do not adequately capture the clustering effect.

To model the small world phenomenon, several researchers have introduced random graph models with additional geometric structure. Kleinberg [7] proposed a model where a grid graph G is augmented with random edges between nodes u, v with probability proportional to $[d_G(u, v)]^{-r}$ for some constant r . Fabrikant, Koutsoupias and Paradimitriou [4] proposed a model where vertices are points in the Euclidean plane and edges are added by optimizing a function involving both Euclidean distance and graph distance to a central node.

Chung and Lu [3] introduced a hybrid graph model where a random power law graph called the “global” graph is added to a “local graph” having a certain kind of local connectivity. In [1] an efficient algorithm was presented for extracting a highly connected local graph from an arbitrary graph. For a graph generated by the hybrid model, this algorithm recovers the original local graph up to a small error.

In this paper, we present a drawing method using the algorithm for extracting local graphs. This algorithm may be useful for drawing graphs similar to those produced by the hybrid model. A graph from the hybrid model contains a random power law graph which will not be amenable to most drawing methods, but also contains a local graph which can be more geometric in nature. The recovery theorem in [1] guarantees that when applied to a graph from the hybrid model, our algorithm produces a layout which depends largely on the local graph.

* Research supported in part by NSF Grants DMS 0100472 and ITR 0205061.

2 Preliminaries

2.1 Weighted Graphs and Quotient Graphs

Although our input graphs are unweighted, our algorithm will form weighted graphs by collapsing connected components into single vertices. A weighted graph is a simple graph G together with a vertex weight function $w_G(v)$ and an edge weight function $\phi_G(e)$. Suppose that $V(G)$ has a partition $V(G) = C_1 \cup C_2 \cup \dots \cup C_k$. The quotient graph Q is defined as follows. The vertices of Q are communities C_1, \dots, C_k , and we set

$$w_Q(C_k) = \sum_{u \in C_k} w_G(u).$$

$$\phi_Q(C_i, C_j) = \sum_{u \in C_i, v \in C_j} \phi_G(u, v).$$

There is an edge between C_i and C_j if $\phi_Q(C_i, C_j) > 0$.

2.2 Local Flow and Local Graphs

Given a weighted graph with edge capacity function ϕ , we will define a notion of local connectivity between vertices. We will say a path is *short* if it has length less than or equal to ℓ . A short flow is a positive linear combination of short paths where no edge carries more than its capacity. The maximum short flow problem can be viewed as a linear program, and can be computed in polynomial time using nontrivial but relatively efficient algorithms for fractional packing (See 2.3).

Definition 1 (Short Flow). *A short flow is a feasible solution to the following linear program. The flow connectivity $f(u, v)$ between two vertices is the maximum value of any short flow, which is the optimum value of the following LP problem. Let P_ℓ be the collection of short u - v paths, and let P_e be the collection of short u - v paths which intersect the edge e .*

$$\begin{aligned} \text{maximize} \quad & \sum_{p \in P_\ell} f_p & (1) \\ \text{subject to} \quad & \sum_{p \in P_e} f_p \leq \phi(e) & \text{for each } e \in L \\ & f_p \geq 0 & \text{for each } p \in P_\ell \end{aligned}$$

We say two vertices u and v are (f, ℓ) -connected if there exists a short flow between them of size at least f . We say a graph L is an (f, ℓ) -local graph if for each edge $e = (u, v)$ in L , the vertices u and v are (f, ℓ) -connected in L .

2.3 Computing the Maximum Short Flow

Finding the maximum short flow between u and v in a graph G with given edge capacities $\phi(e)$ can be viewed as a fractional packing problem, which has the form

$$\max\{\mathbf{c}^T \mathbf{x} \mid A\mathbf{x} \leq \mathbf{b}, \mathbf{x} \succeq \mathbf{0}\}.$$

To view the maximum short flow as a fractional packing problem, first let $G(u, v)$ be a subgraph containing all short paths from u to v . For example, we may take $G(u, v) = N_{\ell/2}(u) \cup N_{\ell/2}(v)$. Let A be the incidence matrix where each row represents an edge in $G(u, v)$ and each column represents a short path from u to v . Let $\mathbf{b} = \phi$, and $\mathbf{c} = \mathbf{1}$.

Using the algorithm of Garg and Könemann in [5] for general fractional packing problems, one can obtain a $(1 - \epsilon)^{-2}$ -approximation to the maximum short flow in time $O(M^2 \ell \lceil \frac{1}{\epsilon} \log_{1+\epsilon} M \rceil)$, where M is the number of edges in $G(u, v)$.

3 Extracting the Local Graph

For a given graph, we wish to extract the largest (f, ℓ) -local subgraph. We define $L_{f, \ell}(G)$ to be the union of all (f, ℓ) -local subgraphs in G . By definition, the union of two (f, ℓ) -local graphs is an (f, ℓ) -local graph, and so $L_{f, \ell}(G)$ is in fact the unique largest (f, ℓ) -local subgraph in G . We remark that $L_{f, \ell}(G)$ is not necessarily connected. The simple greedy algorithm **Extract** computes $L_{f, \ell}(G)$ in any graph G using $O(m^2)$ max-short-flow computations, where m is the number of edges in G . The number of max-short-flow computations can be reduced by using a standard random sampling approach if we are willing to accept approximate local graphs. We say L is an α -approximate (f, ℓ) -local graph if $L(f, \ell) \subseteq L$, and at most an α -fraction of the edges in L are not (f, ℓ) -connected. The algorithm **Approximate Extract** computes a series of approximate local graphs.

Extract:

Input: G, f, ℓ

If there is an edge $e = (u, v) \in G$ where u, v are not (f, ℓ) -connected in G ,
remove e from G .

When no further edges can be removed, output G .

Approximate Extract:

Input: $G, \ell, \{f_1 \leq \dots \leq f_k\}$

Let m be the number of edges in G .

For $i = 1 \dots k$:

Repeat until no edge is removed for $\frac{1}{\alpha} \log \frac{mk}{\delta}$ consecutive attempts:

Pick an edge $e = (u, v)$ from G uniformly at random.

If u, v are not (f_i, ℓ) -connected, remove (u, v) from G .

Let $L_i = G$, reset m to be the number of edges in L_i ,
and proceed to compute L_{i+1} .

Stop when graphs $L_1 \supseteq \dots \supseteq L_k$ have been output.

Since at most m edges are removed from G and there are at most $\frac{1}{\alpha} \log \frac{mk}{\delta}$ attempted removals for every edge removed, **Approximate Extract** performs at most $\frac{m}{\alpha} \log \frac{mk}{\delta}$ max-short-flow computations.

Theorem 1. *Given G, ℓ , and $\{f_1 \leq \dots \leq f_k\}$, let $L_1 \supseteq \dots \supseteq L_k$ be the output of **Approximate Extract**. With probability at least $1 - \delta$, each of the graphs L_i is an α -approximate (f_i, ℓ) -local graph.*

Proof: Given $i \in [1, k]$, let $e_1 \dots e_J$ be the edges removed from L_{i-1} to obtain L_i . Let m_i be the number of edges in L_{i-1} and note that $J \leq m_i$. Let T_j be the number of attempts between the removal of the e_{j-1} and e_j . If L_i is not an α -approximate local graph, then some T_j must be at least $\frac{1}{\alpha} \log \frac{m_i k}{\delta}$ when at least an α -fraction of the edges remaining in L_i were not (f_i, ℓ) -connected. For a given j , this occurs with probability at most

$$(1 - \alpha)^{T_j} \leq e^{-\alpha T_j} \leq e^{-\log \frac{m_i k}{\delta}} \leq \delta m_i^{-1} / k.$$

Since $J \leq m_i$, the probability that this occurs for any T_j is at most δ/k . The probability that a bad T_j occurs for any L_i is at most δ , and the result follows.

4 An Algorithm for Drawing Power Law Graphs

In this section we describe a framework for producing drawings of power law graphs that reflect local connectivity. In the algorithm `Local Draw` below, a local subgraph is used to determine the layout of the vertices. Our algorithm uses as a subroutine a standard force-directed drawing method which we describe in section 4.2, but other methods can be used in its place. The algorithm is motivated by the structure of power law graphs, but can be applied to general graphs as well.

4.1 The Algorithm

Local Draw:

Given an input graph G , compute the local graph $L_{f,\ell}$ for some choice of f and ℓ using `Extract` or `Approximate Extract`. Let $\Pi_{f,\ell}$ be the partition induced by the connected components $C_1 \dots C_k$ of $L_{f,\ell}$, and let Q be the quotient graph of G with respect to this partition. Use the force-based drawing algorithm to produce drawings of each component $C_1 \dots C_k$ and Q separately. To combine into a single drawing, let $q_1 \dots q_k$ be the coordinates of the vertices in Q corresponding to $C_1 \dots C_k$, and let

$$r_i = \frac{1}{2} \min_j \|q_i - q_j\|.$$

Scale each drawing of C_i by r_i , and place at location q_i to create a new drawing which only contains edges in $L_{f,\ell}$. Apply the force-based algorithm to this drawing to determine the final layout of the vertices, and then add back the edges in $G \setminus L_{f,\ell}$.

4.2 A Force-Directed Drawing Method

Our algorithms use a standard force-based drawing method, modified for use on graphs with vertex weights $w(v)$ and edge weights $\phi(e)$. We define a repulsive force between every pair of vertices, where the force acting on vertex u due to vertex v is

$$R_{u,v} = \frac{1}{n^2} \frac{u - v}{\|u - v\|^2} w(u)w(v)$$

Each edge also acts as a spring, with the force on a vertex u from the edge $e = (u, v)$ defined to be

$$S_{u,v} = \frac{1}{n}(v - u)\phi(e)$$

To keep the drawing in a bounded area, we place all vertices within the unit circle and define a force between each vertex and the boundary of the circle.

$$B_u = -\frac{u}{\|u\|} \frac{1}{(1 - \|u\|)} w(u)$$

The standard force-based approach is to compute the sum of the forces acting on each vertex and move in the resulting direction at each time step.

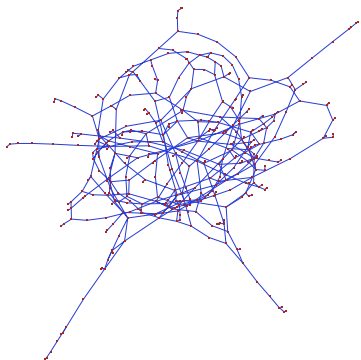


Fig. 1. Local Draw applied to the giant component of random graph $G(n, p)$ with $n = 500$ and $p = 0.004$.

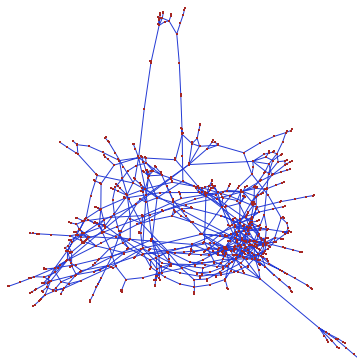


Fig. 2. Local Draw applied to the induced subgraph of G : the collaboration graph on authors with Erdős number exactly 2.

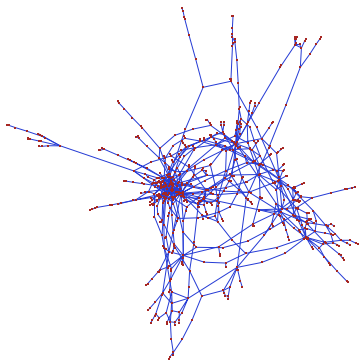


Fig. 3. The quotient graph of G .

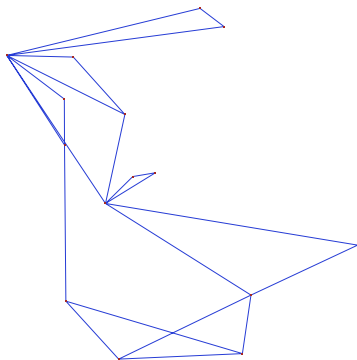


Fig. 4. The largest connected component in the local graph of G , of size 15.

5 Implementation and Examples

We have implemented `Extract` and `Local Draw` and experimented on several examples. Figure 1 is a drawing of a sparse random graph, generated from the Erdős-Rényi model $G(n, p)$ with $n = 500$ and $p = .004$. Jerry Grossman [6] has graciously provided data from a collaboration graph of the second kind, where each vertex represents an author and each edge represents a joint paper with two authors. Our example graph G is the largest component of the induced subgraph on authors with Erdős number exactly 2. This graph contains 834 vertices. We applied `Local Draw` to G with parameters ($f = 2, \ell = 3$), obtaining the drawing in Figure 2, and in the process obtaining the quotient graph shown in Figure 3 and the local graph. The largest connected component of the local graph is shown in Figure 4.

References

1. R. Andersen, F. Chung and L. Lu, Analyzing the small world phenomenon using a hybrid model with local network flow, *Proceedings of the Third Workshop on Algorithms and Models for the Web-Graph* (2004).
2. F. Chung and L. Lu, Average distances in random graphs with given expected degree sequences, *Proceedings of National Academy of Science*, **99** (2002).
3. F. Chung and L. Lu, The small world phenomenon in hybrid power law graphs *Complex Networks*, (Eds. E. Ben-Naim et. al.), Springer-Verlag, (2004).
4. A. Fabrikant, E. Koutsoupias and C. H. Papadimitriou, Heuristically optimized trade-offs: a new paradigm for power laws in the Internet, *STOC 2002*.
5. N. Garg, J. Könemann, Faster and simpler algorithms for multicommodity flow and other fractional packing problems. *Technical Report, Max-Planck-Institut für Informatik, Saarbrücken, Germany* (1997).
6. Jerry Grossman, Patrick Ion, and Rodrigo De Castro, Facts about Erdős Numbers and the Collaboration Graph, <http://www.oakland.edu/~grossman/trivia.html>.
7. J. Kleinberg, The small-world phenomenon: An algorithmic perspective, *Proc. 32nd ACM Symposium on Theory of Computing*, 2000.
8. M. Mitzenmacher, A Brief History of Generative Models for Power Law and Log-normal Distributions, *Internet Math.* 1 (2003), no. 2.