On Generating Graphs with Prescribed Degree Sequences for Complex Network Modeling Applications

- Position Paper -

Milena Mihail and Nisheeth Vishnoi
College of Computing
Georgia Institute of Technology
mihail@cc.gatech.edu, nkv@cc.gatech.edu

Abstract

Graph models for real-world complex networks such as the Internet, the WWW and biological networks are necessary for simulation-based studies of a variety of problems. Currently, there is a fair amount of available data for such networks, and yet, the primitives of how these networks grow and evolve are far from understood. On the positive side, the available data suggest robust and persistent heavy tailed statistics, most notably on the degrees of the network topologies. Consequently, a practical way to generate network topologies that meet the observed data is the following degree-driven approach: First predict the degrees of the graph by extrapolation from the available data, and then construct a graph meeting the degree sequence and, possibly, additional constraints such as connectivity or "randomness", to name a couple. In particular, within the networking community, this is currently accepted as the most successful approach for modeling the topology of the Internet. And yet, despite the significance of the application area and the relative clarity of the problem, to date, all implementations are based on ad-hoc heuristics.

Constructing a simple graph that meets a given degree sequence is a classical problem in graph theory and theoretical computer science. This problem is intimately related to the theory of matchings, and hence, many generalizations of the problem can be also addressed in a strict theoretical framework.

In this paper we point out a range of theoretical primitives that are relevant to the degree-driven network generation approach. Some of these primitives can be readily adapted in practice, and enrich the output of existing network topology generators. More importantly, we formalize a few theoretical problems for which efficient aglorithms can greatly impact the application context. Since the practical applications involve tens of thousands of nodes, approximation algorithms become particularly important.

1 Introduction

There has been a recent surge of interest in complex real-world networks. These include *the WWW* [22, 4, 7, 12, 23, 24] where a node corresponds to a Web page and there is an edge between two nodes if there is a hyperlink between the corresponding pages, the *Internet at the level of Autonomous Systems* [16, 21, 27, 32, 28] where a node corresponds to a distinct routing administration domain (such as a University, a corporation, or an ISP) and an edge represents direct exchange of traffic between the corresponding domains, and *biological networks* [20] where, nodes correspond to genetic or metabolic building blocks (such as genes and proteins) and edges represent direct interactions between these blocks. Obtaining accurate graph models for such real-world networks is necessary for a variety of simulation-based studies.

A very robust and persistent characteristics of complex networks, including the WWW, the Internet and biological networks, is that their degree sequences follow heavy tailed statistics. By this we mean **(a)**the ith largest degree of the graph is proportional to $i^{-\alpha}$, with α approaching 1 from below, **(b)**the frequency of the ith smallest degree of the graph is proportional to $i^{-\beta}$, with β approaching 2 from above (see [16] for detailed Internet measurements, see [4, 12, 23, 24] for WWW measurements, and see also [1] for mathematical expolanation why $\beta \simeq 1+\frac{1}{\alpha}$). This is a sharp departure from the Erdös-Rényi random graph model where the degrees are exponentially distributed around the mean. Consequently, several papers have proposed plausible graph models, based on the syntactic notion of "preferential attachment" [4, 6, 24, 3, 11], and on the semantic notion of multiobjective optimization [15], for explaining this phenomenon. Despite the elegant principles of the above approaches, none of them predicts accurately all the observed measurements; in fact, none of these approaches attempts to explain the heavy tailed statistics on the high-end and the low-end of the degrees, **(a)** and **(b)** above, *simultaneously*. On the other hand, graph models for complex networks are often expected to pass strict performance requirements. For example, the networking community uses such graph models to simulate a wide range of network protocols [38, 16, 21, 26, 27, 31, 32, 35, 8], and hence the accuracy of the underlying topology model is very important.

Therefore, the following alternative approach for generating network topology models is considered. First predict the degrees of the graph to be generated by extrapolation from available data (for example, according to **(a)** and **(b)** above), and then generate a graph that satisfies the target degree sequence, and additional constraints, the first and most natural of which is connectivity. In the theory community this approach was first formalized in [2, 9] who especially addressed the connectivity issue. The main drawback of this approach is that yields graphs with a giant component for $\beta < 2$, which is outside the observed range mentioned above (in addition the generated graphs may contain loops and multiple edges). In the networking community this approach, typified by the Inet topology generator [21], is the method of choice. The current implementations of Inet uses the following heuristic

for constructing a connected graph that meets a predicted degree sequence: First it places a spanning tree to guarantee connectivity, and then it tries to match the remaining degrees "as much as possible" using a "preferential connectivity heuristic".

Generating a graph that meets a certain degree sequence is a classical problem in graph theory and theoretical computer science. In Section 2 we point out basic graph theoretic primitives that can be readily adapted in practice, and enrich the output of existing network topology generators. All graph theoretic primitives of Section 2 are well known, and the new point here is that these primitives have the described direct practical consequences. In Section 3 we formalize several extensions of the basic problem and explain their practical significance. These extensions include costs on links, evolutionary models, and directed graphs. The structure of all these problems arises from an intimate relation between the degree sequence problem and matching theory. For most of these extensions we outline polynomial time algorithms, however, the involved polynomials are of high degree, which makes the algorithms impractical (for example, todays' Internet topologies has approximately 15K nodes [30, 8]). In Questions Q1, Q2, Q3, Q4, Q5, Q6, and Q7 we formalize theoretical questions for which answers would yield algorithms efficient in practice. All the problems adressed in Section 2 are new. In Section 4 we discuss the potential of the Markov chain method for generating a random *connected* graph that meets a certain degree sequence.

2 Foundations of Degree-Based Graph Generation

Let n denote the number of nodes of the graph we wish to generate. Let $v_i, 1 \leq i \leq n$ denote the nodes and $d_1 \geq d_2 \geq \ldots \geq d_n$ denote the intended degrees of these nodes. We would like a simple undirected graph (i.e., without self-loops or multiple links) meeting the above degree sequence. In addition, we want the graph to be connected. In this section we review the classical Erdös-Gallai theorem [5, 14] and Havel-Hakimi construction [18, 19] for addressing this question. We point out that the connectivity requirement can be dealt with separately, and that the supporting theory allows great flexibily in the generated output graph. From the pactical point of view, these fundamental theoretical primitives can replace all ad-hoc heuristics of current implementations.

A sequence of degrees $d_1 \geq d_2 \geq \ldots \geq d_n$ is called *realizable* if and only if there exists a simple graph whose nodes have precisely this sequence of degrees. A straightforward necessary condition for a degree sequence to be realizable is that for each subset of the k highest degree nodes, the degrees of these nodes can be "absorbed" within the nodes and the outside degrees. Stated formally, for $1 \leq k \leq n-1$:

$$\sum_{i=1}^{k} d_i \leq k(k-1) + \sum_{i=k+1}^{n} \min\{k, d_i\} \tag{1}$$

The Erdös-Gallai theorem states that this necessary condition is also sufficient [5, 14].

The proof is inductive and provides the following natural construction algorithm [18, 19]. The algorithm is iterative and maintains the residual degrees of vertices. In each iteration, it picks an arbitrary vertex v and adds edges from v to d_v vertices of *highest* residual degree, where d_v is the residual degree of v. The residual degrees of the latter d_v vertices are updated appropriately. The significance of connecting with d_v highest degree vertices is that it ensures that condition (1) holds for the residual problem instance.

For example, the algorithm can start by connecting the highest degree vertex with d_1 other high degree vertices and obtain a residual degree sequence by reducing the degrees of these vertices by one, and repeat the same process until all degrees are satisfied (otherwise output "not realizable"). Alternatively, the algorithm can connect the lowest degree vertex d_n (resp. or a randomly chosen vertex d_i) with the d_n (resp. d_i) highest degree vertices, reduce their degrees and proceed as above.

Clearly the above algorithm runs in n iterations, each iteration invoking the degree of a vertex (and some book-keeping for maintaining residual degrees in sorted order). Thus the running time is very efficient, both in theory and in practice. In addition, since the sequence in which it picks vertices can be chosen, it provides the flexibility alluded to above. For example, when we start with higher degree vertices we get topologies that have very "dense cores", while when we start with low degree vertices we get topologies that have very "sparse cores".

The Erdös-Gallai condition (1) allows for additional flexibility, which results in topologies more closely resembling real data. The idea is to use the principle of *preferential attachment* for choosing the d_v vertices to which v will be connected, rather than the maximum degree d_v vertices. Thus, the d_v vertices can be chosen with probabilities proportional to their residual degrees. After each iteration, we need to ensure that condition (1) is satisfied by the residual graph (this part was automatic in case maximum degree vertices are chosen). If not, the probabilistic choice needs to be repeated. If it fails several times, we can go back to choosing maximum degree vertices.

Next, let us deal with the second requirement of obtaining a connected topology. If condition (1) is satisfied, then a necessary and sufficient condition to have a connected realization is that the graph contains a spanning tree [5], which means that the sum of the degrees must be at least $2(n-1)$. We can then construct a connected realization as follows. First construct a realization as stated above. If this graph turns out to be unconnected, then one of the connected components must contain a cycle. Let (u, v) be any edge in a cycle and let (s, t) be an edge in a different connected component. Clearly, the graph does not have edges between the pairs u, s and v, t. By removing the edges (u, v) and (s, t), and inserting the edges (u, s) and (v, t), we merge these two components. Note that the resulting graph still satisfies the given degree sequence. Proceeding in this manner, we can get a connected topology.

3 Extensions and Approximations for Costs, Constraints and Direction on Edges

3.1 Mincost Realizations

The first natural extension arises when links have costs. So let $d_1 \geq d_2 \geq \ldots \geq d_n$ be a realizable degree sequence, and suppose that there is a cost $c(i,j)$ associated with a link between v_i and v_j, $1 \leq i,j \leq n$, and since we consider undirected graphs we have $c(i,j) = c(j,i)$. We now want a *mincost* realization of the degree sequence, i.e., the total cost of the used links is minimized among all realizations.

We first observe that there is a polynomial-time algorithm to find a mincost realization by reduction to mincost perfect matching. However, this algorithm is not efficient in practice. Questions (Q1) and (Q2) outline theoretical questions whose answer(s) may yield practical algorithms.

A Polynomial-Time Algorithm via Matchings

For $\mathbf{d}=(d_1,\ldots,d_n)$ as above, let $M_{\mathbf{d}}$ be the following graph. For each $1 \leq i \leq n$, $M_{\mathbf{d}}$ contains a complete bipartite graph $H_i=(L_i,R_i)$, where $|R_i|=n-1$ and $L_i|=n-1-d_i$. The vertices of R_i are labeled so that there is a label for each $1 \leq j \leq n$ other than i; let us denote these labels by $\{u_{i,1},\ldots,u_{i,i-1},u_{i,i+1},\ldots,u_{i,n}\}$. In addition, for each $1 \leq i,j \leq n$ with $j \neq i$, $M_{\mathbf{d}}$ has an edge between $u_{i,j}$ and $u_{j,i}$. Now each perfect matching $\mathcal{M}$ of $M_{\mathbf{d}}$ gives rise to a unique realization $\mathcal{G}$ of $\mathbf{d}$ in the natural way: $\mathcal{G}$ has a link between v_i and v_j if and only if $\mathcal{M}$ contains the edge between $u_{i,j}$ and $u_{j,i}$. Similarly, each realization $\mathcal{G}$ of $\mathbf{d}$ is associated with $\prod_{i=1}^{n}(n-1-d_i)!$ perfect matchings of $M_{\mathbf{d}}$.

Now suppose that we assign costs to edges of $M_{\mathbf{d}}$ as follows: the cost of the edge between $u_{i,j}$ and $u_{j,i}$ is equal to the cost $c(i,j)$ of the link between v_i and v_j, for all $1 \leq i,j \leq n$. The costs of each edges of the bipartite graphs H_i is a fixed quantity λ. Now the total cost of the edges of each perfect matching $\mathcal{M}$ of $M_{\mathbf{d}}$ is of the form $\mathcal{C}+\lambda\sum_{i=1}^{n}(n-1-d_i)!$, where the cost $\mathcal{C}$ is due to edges between $u_{i,j}$'s and $u_{j,i}$'s, and the factor $\lambda\sum_{i=1}^{n}(n-1-d_i)!$ is due to edges of the H_i's. Consequently, each perfect matching $\mathcal{M}$ of $M_{\mathbf{d}}$ of total cost $\mathcal{C}+\lambda\sum_{i=1}^{n}(n-1-d_i)!$ gives rise to a unique realization $\mathcal{G}$ of $\mathbf{d}$ of total cost $\mathcal{C}$ in the natural way described in the paragraph above. Similarly, each realization $\mathcal{G}$ of $\mathbf{d}$ of total cost $\mathcal{C}$ is associated with $\prod_{i=1}^{n}(n-1-d_i)!$ perfect matchings of $M_{\mathbf{d}}$, each one of cost $\mathcal{C}+\lambda\sum_{i=1}^{n}(n-1-d_i)!$. It can be now seen that a mincost perfect matching of $M_{\mathbf{d}}$ directly suggests a mincost realization of $M_{\mathbf{d}}$. Since finding a minimum cost perfect matching is in polynomial time [25, 29, 10], it follows that finding a mincost realization is also in polynomial time.

Open Questions

From the practical point of view, the drawback of the method described in the previous subsection is that the reduction graph $M_{\mathbf{d}}$ is of size $O(n^2)$. We are thus invoking mincost perfect matching algorithms for a graph with $O(n^2)$ vertices, with n in the order of 10K to 20K; this is practically nearly unacceptable. The hope in moving to more practical running times lies in the fact that the undelying graph has special structure, and that reasonable approximations [37] would be acceptable for all practical purposes. We thus raise the following question:

Q1: Find algorithms that solve the mincost realization problem for a network of size n, even in an approximate sense, even when costs are of special kind (metric, Euclidean, highly "clustered), even for special classes of degree sequences (which include sequences with heavy tailed statistics on the degrees), and have running times substantially lower than invoking mincost perfect matching for a graph with $O(n^2)$ vertices.

Now let us revisit the requirement that the realization is mincost and connected. For this case, we do not even know of a polynomial time algorithm. We thus raise the following question:

Q2: What is the complexity of finding a mincost connected realization for a network of size n? Is there an efficient algorithm, even in an approximation sense, even when costs are of special kind (metric, Euclidean, "highly clustered"), and even for special classes of degree sequences (which include sequences with heavy tailed statistics on the degrees)?

Remark: In practice, even partial anwers to the above questions are likely to be significant. For example, a good algorithm to solve (Q1) can be combined with a heuristic along the lines of the last paragraph of Section 2 and yield reasonable practical performance.

3.2 Restrictions on Links and Evolutionary Models

A second natural extension arises when cetrain links are not allowed. This can be thought of as a special case of mincost realizations, by assigning a very large cost to each disallowed link. We sketch this case separately because it has very special practical significance. In particular, we may think of two degree sequences $d_1 \geq d_2 \geq \ldots \geq d_n$ and $d'_1 \geq d'_2 \geq \ldots \geq d'_n$, with $d'_i \geq d_i$, $1 \leq i \leq n$, aiming to capture the fact that the second degree sequence is an evolution of the first one. Let $\delta_i = d'_i - d_i$, $1 \leq i \leq n$. Now if $\mathcal{G}$ is a realization of the first degree sequence, we may wish to construct a realization of the degree sequence $\delta_1, \ldots, \delta_n$ which does not use links present in $\mathcal{G}$, thus representing a certain *evolution* of $\mathcal{G}$ (note that the above notation the δ_i's are not necessarily non-increasing, but this does not have any impact on any of the algorithmic agruments). Now the following questions arise:

Open Questions

Q3: Given a degree sequence $d_1 \geq d_2 \geq \ldots \geq d_n$, and a set $\mathcal{E}$ of disallowed links over a set of n vertices, find algorithms that solve the realization problem, even in an approximate sense, even when the structure of $\mathcal{E}$ is of special kind, even for special classes of degree sequences (which include sequences with heavy tailed statistics on the degrees), and have running times substantially lower than invoking mincost perfect matching for a graph with $O(n^2)$ vertices.

Q4: Given degree sequences $d_1 \geq d_2 \geq \ldots \geq d_n$ and $d'_1 \geq d'_2 \geq \ldots \geq d'_n$, with $d'_i \geq d_i$, $1 \leq i \leq n$, construct respective realizations $\mathcal{G}$ and $\mathcal{G}'$, where $\mathcal{G}$ is a subsraph of $\mathcal{G}'$ (or decide that such realizations do not exist). What is the complexity of the above question? Are there efficient algorithms, even in an approximate sense, even for special classes of degree sequences (which include sequences with heavy tailed statistics on the degrees)?

3.3 Graphic Approximations of Non-Graphic Sequences

Let $f_1 \geq f_2 \geq \ldots \geq f_n$ be a non-realizable degree sequence (i.e., condition (1) fails and let $d_1 \geq d_2 \geq \ldots \geq d_n$ be a realizable degree sequence. Say that a vertex v_i has "deficit" if $f_i > d_i$ and let $f_i - d_i$ be this deficit. Say that a vertex v_i has "surplus" if $f_i < d_i$ and let $d_i - f_i$ be this surplus. Define $dist(\mathbf{f}, \mathbf{d})$ as the sum of the deficits and the surpluses.

Given a non-realizable degree sequence $\mathbf{f}$, it is natural to ask for a realizable degree sequence $\mathbf{d}$ which minimizes $dist(\mathbf{f}, \mathbf{d})$. For this problem, there is a polynomial time algorithm which invokdes maximum cardinality matching of a graph with $O(n^2)$ vertices. In particular, where $M_{\mathbf{f}}$ is the reduction graph related to $\mathbf{f}$, as described in Section 3.1, it can be verified that maximum cardinality matchings of $M_{\mathbf{f}}$ give rise to realizable "approximations" $\mathbf{d}$) of the non-realizable degree sequence $\mathbf{f}$ which minimize $dist(\mathbf{f}, \mathbf{d})$. It would be particularly elegant to find a direct (and faster) approximation algorithms, for example reminiscent of the Havel-Hakimi construction of Section 2.

Q5: Given a sequence of n integers $\mathbf{f}$ in non-increasing order, we wish to find a realizable degree sequence $\mathbf{d}$ that minimizes $dist(\mathbf{f}, \mathbf{d})$. Is there a natural extension of the Havel-Hakimi construction for this problem? Is there a generalization of the Erdös-Gallai condition (1)?

Remark: We can show that min-dist approximations of a non-realizable degree sequence have several interesting structural properties: The subgraph induced by surplus vertices form an independent set, subgraph induced by deficit vertices form a clique, and there are min-dist realizations that involve only surplus or only deficit vertices.

Q6: Given a sequence of n integers $\mathbf{f}$ in non-increasing order, and a realizable degree sequence $\mathbf{d}$, what are there other natural notions of "distance" between $\mathbf{f}$ and $\mathbf{d}$? Are there efficient algorithms "approximate" $\mathbf{f}$ by a realizable $\mathbf{d}$ of whose "distance" from $\mathbf{f}$ is small?

3.4 Directed Network Models and a Reduction to Flows

So far we dealt with undirected graphs. However, in various application contexts, there are strong semantics suggesting directed edges. In the WWW these semantics are transparent and they denote the direction of a hyperlink. In the Internet topology context these semantics denote certain "customer-provider" relationships. As opposed to the WWW, these relationships are opaque (i.e., pre-nagotiated between partners, but never explicitly advertized in the transparent parts of the network protocols), and considerable research effort has been put to "infer" these relationships from routing tables and traces [17, 34]. As expected, the degrees of the corresponding directed complex networks also follow heavy tailed statistics. In particular, the large in-degrees follow a Zipf law with exponent approaching 1 from below, and the small out-degrees follow a power-law with exponent approaching 2 from above (similar to (a) and (b) of Section 1). It is therefore natural to address the issue of generating a directed graph that satisfies, simultaneously a given degree sequence for in-degrees, and a given degree sequence for out-degrees (even though, to date, the correlation of in-degrees and out-degrees is not well understood).

Let $\mathbf{d_{in}} = (d_{in,1}, d_{in,2}, \ldots, d_{in,n})$ and $\mathbf{d_{out}} = (d_{out,1}, d_{out,2}, \ldots, d_{out,n})$ be sequences of integers (in no particular sorted order), with $\sum_{i=1}^{n} d_{in,i} = \sum_{i=1}^{n} d_{out,i}$. We wish to construct a directed graph on n nodes, such node v_i has $d_{in,i}$ incoming edges and $d_{out,i}$ outgoing edges, $1 \leq i \leq n$.

Consider the following network. There is a source s, a sink t, a set of nodes $L = \{l_1, \ldots, l_n\}$ and a set of nodes $R = \{r_1, \ldots, r_n\}$. There is a link of capacity one directed from each l_i to each r_j, for $1 \leq i, j \leq n$ and $i \neq j$. There a link of capacity $d_{out,i}$ directed from s to each l_i, for $1 \leq i, j \leq n$. Finally, there is a link of capacity $d_{in,i}$ directed from each r_i to t, for $1 \leq i, j \leq n$. We may now consider integral maximum flows from s to t. If there is a such a flow of value $\sum_{i=1}^{n} d_{in,i} = \sum_{i=1}^{n} d_{out,i}$, then the corresponding degree sequences are realizable, and the flow gives a directed graph that satisfies, simultaneously, in-degrees $d_{in,i}$ and out-degrees $d_{out,i}$.

Q7: Are there more efficienet algorithms (beyond max-flow on bipartite networks), for the maximum flow problem stated above?

4 Random Graphs and the Markov Chain Method

We now turn to the question of generating a *random* instance from the space of all possible connected graphs that realize a target degree sequence. In experiment, it has been observed that "random" such instances are excellent fits for several characteristics of complex network topologies [2, 28, 35] (albeit, all these experiments fall short of guaranteeing that the generated instances are either "correct" connected realizations of the target degree sequence, or "close" to random).

For any sequence of integers that has a connected realization, consider the following Markov chain. Start from a connected graph $\mathcal{G}_t$ with this degree sequence. Pick two edges at

random, say (u, v) and (x, y) with distinct endpoints. If (u, x) and (v, y) are not edges then consider a graph $\mathcal{G}'$ by removing the edges (u, v) and (x, y) and inserting the edges (u, x) and (v, y). Observe that $\mathcal{G}'$ still satisfies the given degree sequence. We further have to check whether $\mathcal{G}'$ is a connected graph. If it is connected then we perform the switching operation and let $\mathcal{G}_{t+1}$ be $\mathcal{G}'$. Otherwise we do not perform the switching operation and $\mathcal{G}_{t+1}$ remains $\mathcal{G}_t$. It follows from a theorem of Taylor [5, 36] that, using the above switching operation, any *connected graph* can be transformed to any other *connected graph* satisfying the *same degree sequence* (we note that the proof of Taylor's theorem is somewhat more involved than the corresponding fact for realizations *without the connectivity constraint*; the latter fact is straightforward). It now follows from standard Marckov chain theory [29, 33] that, in the limit, the above Markov chain will generate a random graph with the given degree sequence.

What is the mixing rate? Similar questions have been considered elsewhere [9] without the connectivity requirement, however, the crucial connectivity consideration has not been appeared before. In experiment we have found very fast convergence (less than 1 million steps for networks of up to 15K nodes resembling the real data of [30, 8]).

Remark: In general, the problem of rapid mixing for connected realizations is strictly harder than that of arbitrary realizations (there is a simple reduction from the latter problem to the former by forcing one vertex that is connected to all other vertices). However, special cases, such as trees, are still of interest.

References

[1] Adamic, L., "Zipf, Power-laws, and Pareto - a ranking tutorial", (2000), http://www.hpl.hp.com/shl/people/ladamic.

[2] Aiello, W., Chung, F.R.K. and Lu, L., "A Random Graph Model for Power Law Graphs", Proceedings of the Fourty-First Annual IEEE *Symposium on Foundations of Computer Science*, (FOCS 2000), pp. 171-180.

[3] Aiello, W., Chung, F.R.K. and Lu, L., "Random Evolution in Massive Graphs", Proceedings of the Fourty-Second Annual IEEE *Symposium on Foundations of Computer Science*, (FOCS 2001), pp. 510-519.

[4] Barabási, A.-L. and Albert, R., "Emergence of scaling in random graphs", Science 286 (1999), pp. 509-512.

[5] Berge, Claude, "Graphs and Hypergraphs", North Holland Publishing Company, 1973.

[6] Bollobás, B., Riordan, O., Spencer, J. and Tusnády, G., "The degree sequence of a scale-free random graph process", Random Structures and Algorithms,Volume 18, Issue 3, 2001, pp 279-290.

[7] Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., Tomikns, A. and Wiener, J., "Graph structure in the Web", Proc. 9th International World Wide Web Conference (WWW9)/Computer Networks, 33(1-6), 2000, pp. 309-320.

[8] Chen, Q., Chang, H., Govindran, R., Jamin, S., Shenker, S. and Willinger, W., "The Origin of Power Laws in Internet Topologies Revisited", Infocom 2002, to appear.

[9] Chung, F.R.K. and Lu, L., "Connected components in random graphs with given degree sequences", Available at http://www.math.ucsd.edu/ fan.

[10] Cook, W., "Matchings: Research Software", http://www.math.princeton.edu/ bico.

[11] Cooper, C. and Frieze, A., A general model for web graphs, *Proceedings of ESA*, 2001, pp 500-511.

[12] Dill, S., Kumar, R., McCurley, K., Rajagopalan, S., Sivakumar, D. and Tomkins, A., "Self-similarity in the Web", In Proceedings of International Conference on Very Large Data Bases, Rome, 2001, pp. 69-78.
[13] Dorogovtsev, S.N. and Mendes, J.F.F., "Evolution of Networks", *Advances in Physics*, to appear (2002). Available at http://www.fc.up.pt/fis/sdorogov.
[14] Erdös. P. and Gallai, T., "Graphs with Prescribed Degrees of Vertices", *Mat. Lapok, 11*, 1960.
[15] Fabrikant, A., Koutsoupias, E. and Papadimitriou, C.H. "Heuristically Optimized Tradeoffs", Available at http://www.cs.berkeley.edu/ christos.
[16] Faloutsos, M., Faloutsos, P. and Faloutsos, C., "On Power-law Relationships of the Internet Topology", In Proceedings *Sigcomm* 1999, pp 251-262.
[17] Gao, L., "On Inferring Autonomous System Relationships in the Internet", *IEEE Glogal Internet*, 2000.
[18] Hakimi, S.L., "On the Realizability of a Set of Integers as Degrees of the Vertices of a Graph", *SIAM J. Appl. Math. 10*, 1962.
[19] Havel, V., "A Remark on the Existence of Finite Graphs", *Caposis Pest. Mat. 80*, 1955.
[20] H. Jeong, B. Tombor, R. Albert, Z.N. Oltvai, and A.-L. Barabasi, "The large-scale organization of metabolic networks", Nature 407, 651-654 (2000). si
[21] Jin, C., Chen, Q. and Jamin, S., "Inet: Internet Topology Generator", University of Michigan technical Report, CSE-TR-433-00. Available at http://irl.eecs.umich.edu/jamin.
[22] Kleinberg, J., "Authoritative sources in a hyperlinked environment", Proc. 9th ACM-SIAM Symposium on Discrete Algorithms, 1998. Extended version in Journal of the ACM 46(1999).
[23] Kumar, R., Rajagopalan, S., Sivakumar, D. and Tomkins, A., "Trawling the web for emerging cyber-communities", *WWW8/Computer Networks*, Vol. 31, No 11-16, 1999, pp. 1481-1493.
[24] Kumar, R., Raghavan, P., Rajagopalan, S., Sivakumar, D., Tomkins, A. and Upfal, E., "Stochastic models for the Web graph", *Proceedings of the 41st IEEE Symposium on Foundations of Computer Science*, (FOCS 2000), pp 57-65.
[25] Lovász, L., and Plummer, M.D., *Matching Theory*, Academic Press, New York, 1986.
[26] Medina, A., Lakhina, A., Matta, I. and Byers, J., BRITE: Universal Topology Generation from a User's Perspective. Technical Report BUCS-TR2001 -003, Boston University, 2001. Available at http://www.cs.bu.edu/brite/publications.
[27] Medina, A., Matta, I. and Byers, J., "On the origin of power laws in Internet topologies", *ACM Computer Communication Review*, vol. 30, no. 2, pp. 18–28, Apr. 2000.
[28] Mihail, M., Gkantsidis, C., Saberi, A. and Zegura, E., "On the Semntics of Internet Topologies", Georgia Institute of Technology, Technical Report GIT-CC-0207.
[29] Motwani, R. and Raghavan, P., *Approximation Algorithms*, Cambridge University Press, 1995.
[30] NLANR, National Laboratory for Applied Networking Research, Routing Data, Supported by NSF, http://moat.nlanr.net/Routing/rawdata.
[31] Palmer, C. and Steffan, J., "Generating network topologies that obey power laws", In Proceedings of *Globecom* 2000.
[32] Radoslavov, P., Tangmunarunkit, H., Yu, H. and Govindan, R., "On Characterizing Network Topologies and Analyzing Their Impact on Protocol Design", USC-CS-TR-00-731, http://citeseer.nj.nec.com/radoslavov00characterizing.html, March, 2000.
[33] Sinclair, A., *Algorithms for Random Generation and Counting: A Markov Chain Approach*, Springer-Verlag, 1997.
[34] Subramanian, L., S. Agarwal, J. Rexford, and R.H. Katz, "Characterizing the Internet Hierarchy from Multiple Vantage Points", Infocom 2002, to appear.
[35] Tangmunarunkit, H., Govindan, R., Jamin, S., Shenker, S., Willinger, W., "Network Topologies, PowerLaws, and Hierarchy", U.S.C. Tech. Report (http://www.isi.edu/ hongsuda/resume.html), May 2001.
[36] Taylor, R., "Constrained Switchings in Graphs", *SIAM J. ALG. DISC. METH.*, Vol. 3, No 1, 1982, pp 115-121.
[37] Vazirani, V., *Approximation Algorithms*, Springer-Verlag, 2001.
[38] Zegura, E., Calvert, K., and Donahoo, M., "A quantitative comparison of graph-based models for Internet Topology", *IEEE/ACM Transactions on Networking*, 1997.