Proceedings of the 2009 Winter Simulation Conference
M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin, and R. G. Ingalls, eds.

A STRUCTURAL APPROACH TO THE TEMPORAL MODELING OF NETWORKS

Isabel Beichl
Brian Cloteaux

Mathematical and Computational Sciences Division
National Institute of Standards and Technology
100 Bureau Drive, Stop 8910
Gaithersburg, MD 20899-8910, U.S.A

ABSTRACT

Simulation of many dynamic real world systems such as the Internet and social networks requires developing dynamic models for the underlying networks in these systems. Currently, there is a large body of work devoted towards determining the underlying mechanisms that create these networks, but the resulting models have not realistically captured many of the important structural characteristics when compared with real world examples. Towards creating more realistic dynamic models, we propose a method of structurally constructing models of an evolving network. We then conduct a series of computational experiments in modeling the evolution of the autonomous system (AS) topology of the Internet to test the effectiveness of our approach.

1 INTRODUCTION

During the past decade, starting with Barabási and Albert (1999), there have been a number of papers devoted to describing the relationships between entities in a system using networks with power-law distributed degree sequences. In a network, the number of edges connected to a node is called the *degree* of the node and the set of all the degrees in a graph is called its *degree distribution*. For a degree sequence to be power-law distributed means that the probability that a node has k adjacent edges is $P(k) \sim k^{-\alpha}$ for some $\alpha > 1$. This distribution in a graph produces a few nodes with very high degree (often called *hub nodes*) and a large number of low degree nodes.

The importance of networks, especially those having power law distributions, for modeling applications lies in the number of areas in which they are found. These networks have been shown to arise naturally in systems of both biological (Watts and Strogatz 1998, Jeong et al. 2000, Williams and Martinez 2000) and social (Amaral et al. 2000) interactions. They also appear in many engineered systems such as the power grid (Watts and Strogatz 1998), the Internet (Faloutsos, Faloutsos, and Faloutsos 1999), and software components (Potanin et al. 2005). Thus, realistic models of these types of interactions need to reflect their power law distribution.

While there have been a number of recent papers proposing possible mechanisms for how these power-law networks are created, the resulting models from these proposals have not realistically captured many of the structural characteristics that are observed in real life networks. Thus, if we want to model the time progression of some network, we cannot, at this moment, accurately rely on using any of these mechanisms. In order to get around this lack of understanding of the underlying generating principle for making models, we propose a strictly structural approach.

A structural approach does not try to mimic how the network evolves, but rather it generates a random network with a series of characteristics that we choose as being important to our model. What we define as a temporal approach to structural modeling is simply the structural creation of a network model with the additional constraint that model must contain a core that reflects the network from an earlier period. In other words, we are attempting to model the evolution of a network over time but without specifying the generating mechanism. Towards this goal, we present a new algorithm for creating these types of temporal models, and then test our approach by modeling progressions of the autonomous system (AS) topology.

978-1-4244-5771-7/09/$26.00 ©2009 IEEE

2 TEMPORAL MODEL CONSTRUCTION

We first present a description of our modeling generating techniques. This includes a new approach for creating models with an approximate core and combines that with earlier work in static structural modeling.

2.1 Definitions

When we speak of a network in this paper, we are actually speaking of a simple connected graph. In order to be able to talk about our network modeling algorithms, we need to give a set of definitions of the terms that we will use later.

A graph $G = (N,E)$ has a node set N and an edge set E. To show that an edge set E (or a node set N) belongs to a graph G, we write $E(G)$ (or $N(G)$). The *degree sequence* of G is defined as $\omega = \{\omega_1, \omega_2, \ldots, \omega_{|N|}\}$ where the degree of $n_i \in N(G)$ is ω_i. To show that a degree sequence ω belongs to a graph G, we write $\omega(G)$. The set of all simple connected graphs with the degree sequence ω is represented as $\mathscr{G}(\omega)$. One operation on degree sequences that we will use is the following decrement operation

$$\left(\ominus_{i_1,\ldots,i_k}\omega\right)_i = \begin{cases} \omega_i - 1 & \text{for } i \in \{i_1,\ldots,i_k\} \\ \omega_i & \text{otherwise} \end{cases}$$

The operator $\ominus$ decrements from the degree sequence ω at the indices $i_1,\ldots,i_k$. An example of this operator is for the degree sequence $\omega = (5,3,2,2)$, then $\ominus_{2,4}\omega = (5,2,2,1)$.

2.2 Creating a Model Containing an Approximate Subgraph

Constructing a simple undirected graph with certain properties is a classic problem in graph theory and computer science. The problem of creating a model with a given degree sequence and having a given subgraph embedded in it has been considered by Mihail and Vishnoi (2002). In their paper, Mihail and Vishnoi showed that it is possible to construct a graph having a degree sequence and with a given subgraph in polynomial time. Unfortunately, their algorithm is more of a theoretical result than a practical one, since it contains large run-time factors which makes it impractical for all but extremely small graphs. Thus they state that it is an open problem to find an fast and practical algorithm to construct these types of graphs.

We do not solve this problem, but rather we show that it is easy and fast to construct models with a given degree sequence and that "approximately" contains a given subgraph. What we define as approximately containing a subgraph is a graph that contains almost all the edges for a given subgraph.

Our implementation builds on a sequential algorithm for building random graphs with a given degree sequence that was given by Blitzstein and Diaconis (2006). Their algorithm constructs a graph by iteratively choosing a vertex i with smallest value in a degree sequence ω_i. The vertex i is then connected to ω_i random vertices which pass a graphicality test. In other words, the test creates a set of potential edges with the assurance that by adding any of these edges it does not prevent the remainder of the graph from being created (i.e. we reach a nongraphical degree sequence).

We modify the Blitzstein and Diaconis algorithm by ensuring that if a potential edge is from our desired subgraph, it is automatically added to the graph as we create it. If there are no edges from the subgraph to add, then the algorithm randomly picks an edge from the set of potential edges. These extra checks are necessary since, in general, if we simply start with the core graph and then simply add random edges to it, we have a high probability of reaching a nongraphical degree sequence. By using the framework of Blitzstein and Diaconis, we ensure that this cannot happen and thus that the algorithm will terminate with an appropriate model. The Blitzstein and Diaconis algorithm with our modification is detailed in Algorithm 1.

Our implementation of this algorithm is a modification of code originally provided by Blitzstein (2007). We converted his original code from R into C++, and added a series of optimizations and modifications.

2.3 Using Optimization to Create Models

An alternative approach of model building was suggested by Beichl and Cloteaux (2008a). That approach is to generate network models with a given degree sequence ω and some metric value m_p with an error bound $\varepsilon > 0$. Thus the graph we create has a metric value of $m(G) \in [m_p - \varepsilon, m_p + \varepsilon]$. The error bound is necessary since, in general, for a given metric value m_p and degree sequence ω, there is no guarantee that a graph G exists that meets those constraints.

In order to generate the model we first create an arbitrary graph with the given degree sequence (i.e. using an algorithm such as the Havel-Hakimi algorithm (Hakimi 1962); to ensure initial connectivity see Gkantsidis, Mihail, and Zegura (2003)).

```
Input: a degree sequence $\omega = \{\omega_1, \ldots, \omega_m\}$ and a subgraph $S$
Output: a graph $G$ with degree sequence $\omega$
Create new graph $G$ with $m$ nodes and no edges ;
while $\omega \neq 0$ do
    Choose the smallest $i$ where $\omega_i$ is a minimal nonnegative value;
    while $\omega_i > 0$ do
        Compute candidate list $J = \{j \neq i | (i,j) \notin G$ and $\ominus_{i,j}\omega$ is graphical $\}$;
        if there exists$j \in J$ where $\{i,j\} \in E(S)$ then
            Pick $j$;
        else
            Pick $j \in J$ with probability proportional to its degree;
        end
        Add $\{i,j\}$ to $G$ and update $\omega$ to $\ominus_{i,j}\omega$;
    end
end
return $G$;
```

Algorithm 1: The modified Blitzstein-Diaconis algorithm for the creation of a graph with a given degree sequence and an approximate subcore

Using the created graph as a starting point, we create a Markov chain of all the connected graphs with the degree sequence ω and then we direct a walk over that chain to find a graph with the desired metric value. To define the Markov chain, two connected graphs G_1, G_2 are adjacent if there exist the edges $(u,v),(x,y) \in E(G_1)$ and $(u,x),(v,y) \in E(G_2)$. In other words, we can transform the current graph to a new graph with the same degree sequence using the degree-preserving switch defined by $[(u,v),(x,y)] \rightarrow [(u,x),(v,y)]$. This degree switch is shown in Figure 1.

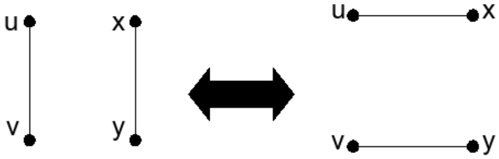


Figure 1: The degree preserving switch $[(u,v),(x,y)] \rightarrow [(u,x),(v,y)]$

To select the path to follow on the Markov chain, we minimize the difference between the metric value of the current graph and the desired value by picking switches using a form of simulated annealing called threshold acceptance (Dueck and Scheuer 1990). Threshold acceptance is an optimization technique that accepts any transition in the state space that does not increase the overall difference between the metric value m of the current graph and m_p by more than a defined threshold. Thus for any randomly chosen switch of two edges in a graph, we would accept the switch to produce a new graph G' if $|m(G') - m_p| < |m(G) - m_p| + t$. This threshold is non-negative and decreases to zero in time. We return the graph G' if $|m(G') - m_p| \leq \varepsilon$.

The major computational cost in this approach is ensuring that an edge switch does not produce a disconnected graph. Towards reducing the number of checks needed, we used the established heuristics of Viger and Latapy (2005) and Gkantsidis, Mihail, and Zegura (2003) for batching up a number of switches before checking connectivity. Previous work by Beichl and Cloteaux (2008b) used this approach to create network models based on the S-metric (we discuss the S-metric in Section 3.1.1).

It should be pointed out that we can optimize a graph using this method for any set of metrics. By optimizing over vectors of values and then using some metric function to compare the vectors, it allows arbitrary combinations of metrics to be used. Unfortunately, our experience is that as we add additional dimensions to optimize over, the convergence rate drastically drops. This explains why in our case study, which follow in the next sections, we generated our models using only the S-metric. Tests of optimizing over smaller graphs (around 700 nodes) using S-metric and clustering coefficient have shown substantial structural characteristics in most instances, but the time required to apply this type of optimization to larger models (in this case, networks with over 30,000 nodes) quickly becomes prohibitive.

Date	Num. of weeks
01/01/2007	0
01/08/2007	1
01/22/2007	3
02/19/2007	7
03/19/2007	11
06/11/2007	23
09/03/2007	35
11/26/2007	47
05/12/2008	71
10/27/2008	95

(a) Number of weeks from first date for AS instances

Date	Num. Nodes	Num. Edges
01/01/2007	29,167	136,888
01/08/2007	29,206	137,457
01/22/2007	29,356	139,310
02/19/2007	29,683	142,486
03/19/2007	30,050	145,865
06/11/2007	31,104	154,113
09/03/2007	32,008	162,497
11/26/2007	32,932	170,083
05/12/2008	34,775	187,730
10/27/2008	36,938	208,043

(b) Size of AS instances

Figure 2: These table show the selected dates and the size of those instances for our case study. All instances of the AS topology were retrieved from the UCLA Internet data.

2.4 A Hybrid Approach

A simple method to increase the quality of our models is to combine the two approaches above by first generating a graph with the desired degree sequence and the approximate subgraph, and then performing the optimization algorithm over all the edges not in the desired subgraph. This method captures the advantages of both approaches by having the desired subcore and at the same time having the structural characteristics we desire in a model.

3 MODELING THE EVOLUTION OF THE AUTONOMOUS SYSTEM TOPOLOGY

In order to test the effectiveness of these structural approaches, we examined how well we were able to model the evolution of the autonomous system (AS) topology over a two year period. We used the Internet topology database at UCLA (2009) (for information on how these topologies were collected, refer to the overview by Oliveira, Zhang, and Zhang (2007)).

We selected 10 topologies over a two year period as shown in Figure 2. These dates were selected to determine how accurately we could project the AS topology from the start date of January 1, 2007. Thus the dates we selected are more frequent near the beginning, and become more spaced out as we project farther into the future.

For our study we created three sets of models based on the sequential, optimization, and hybrid algorithms (Sections 2.2, 2.3, and 2.4 respectively). For the optimization and hybrid algorithms, we optimized using the S-metric value of the AS topology for a targeted date. In our computational experiments, we generated 50 models using each algorithm for each of the dates from January 8, 2007 to October 27, 2008 (see Figure 2). For each of the three algorithms, we computed the mean and 99% confidence levels for five structural metrics on each set of networks at each date.

As an additional point of comparison, we also included a data set based on the preferential attachment approach of Barabási and Albert (1999). In that data set, we started with the core graph of the starting date. We then added nodes sequentially which connected to r nodes in the graph. The value r was randomly chosen each time so that the average degree of the network matched the average of the real networks at the date we were modeling. Although this approach does not represent the state-of-the-art in producing networks using preferential attachment, it was included to give a reference to where the structural approach has shown superior characteristics to generating mechanisms.

3.1 Metric Description

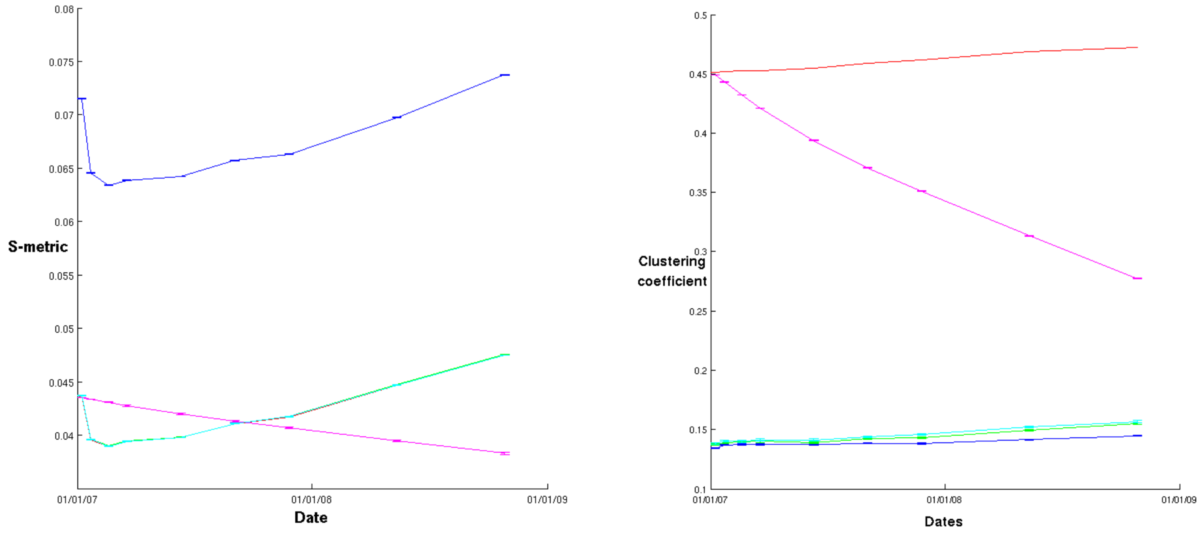
We now describe a series of metrics against which we measured our resulting models. In addition, for the S-metric, we also performed our optimization during model creation using that metric.

In the figures that follow, we demonstrate how well a series of network metrics were preserved by our modeling techniques. In these figures, the red line represent the actual values for the AS topologies. The legend for the remaining lines are blue for sequential approach (Section 2.2), cyan for the optimization method (Section 2.3), green for the hybrid approach (Section 2.4), and mauve for the preferential attachment algorithm. The error bars represent the .99 confidence level for the average of the points in the set.

3.1.1 *S*-metric

The *S*-metric was proposed by Li et al. (2005) as a measure of the interconnectedness between the hub nodes of a network. Li et al. showed that this metric is able to distinguish between many graphs having identical power law distributions.

In order to allow comparisons of networks of differing sizes, the definition of *S*-metric has a normalization factor so that the value is always from zero to one. While this normalization factor allows us to compare the values between different instances as the network evolves, its computation is not trivial. It can be slow and difficult to compute the normalization value used in the original paper. Thus for this paper, we instead normalize by an approximation of this factor given by Alderson and Li (2007). Throughout this paper, whenever we speak of the *S*-metric value of the network, we are actually using this approximation of the normalization factor rather than its true value.



(a) *S*-metric values

(b) Clustering coefficient

Figure 3: *S*-metric values and cluster coefficient

In Figure 3(a), we see how the *S*-metric value is preserved for our metrics. Since we created models for the optimization and hybrid approaches to converge to the actual AS values, the figure has all three lines laying over each other. One point we notice is that the shape of the *S*-metric is preserved for the first three algorithms, but is radically different for the preferential attachment case.

3.2 Clustering Coefficient

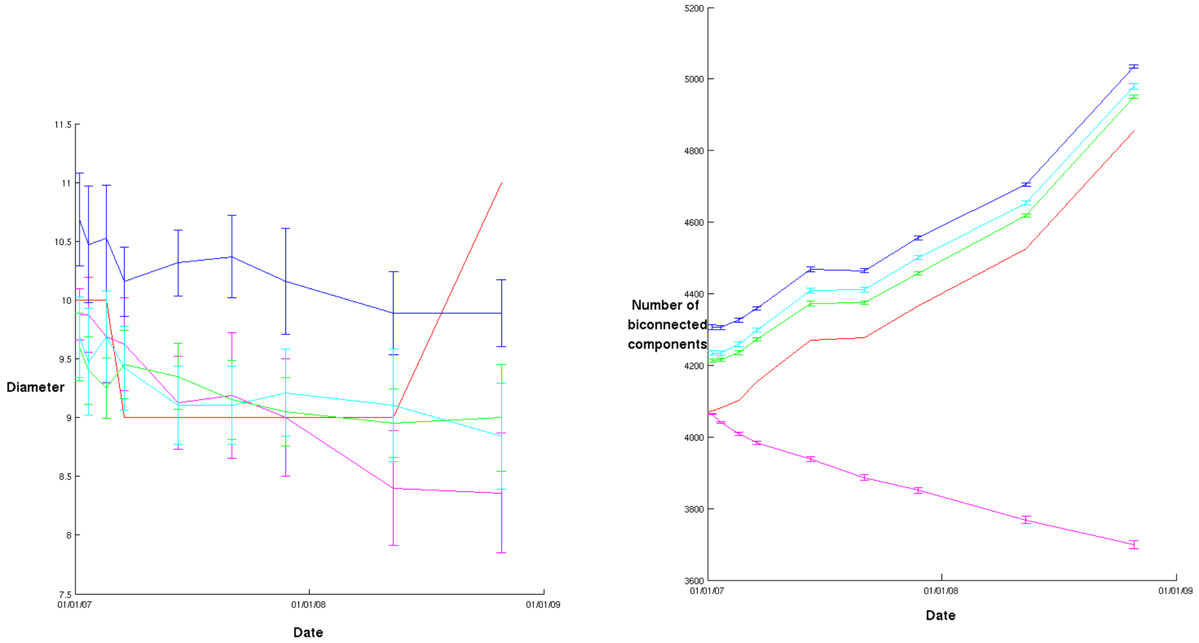
Clustering coefficient is a measure introduced by Watts and Strogatz (1998) to measure the cliquishness of a network or how locally connected a network is. In other words, for any node in the network it shows how likely the neighbors of that vertex are to be connected to one another. When we speak of the clustering coefficient of a network, we mean the average of all the local clustering coefficients for each of the nodes in that network.

When we compare how the clustering coefficient is preserved in Figure 3(b), we notice that the shape is captured well by our algorithms, but there are significant differences in the actual values. In contrast, the preferential attachment algorithm comes closer to the actual values, but is falling away very quickly. Thus in the short term, the preferential attachment model has better values, but as time progresses the other models would appear to become more accurate.

The large differences between the model and AS values are probably explained by the fact that we optimized the network using only the *S*-metric value. In our experience, optimizing only over the *S*-metric can dramatically affect the clustering coefficient values in the generated networks.

3.3 Diameter

The diameter of a graph G is the maximum length of all the shortest paths between any two nodes in G. Diameter is a rough measure of the expected size of paths in a network, since the diameter must be at least as large as the mean path distance. In other words, the smaller the diameter of a network, the smaller we expect the length of the path between any two nodes in the network to be.



(a) Diameter

(b) Biconnected components

Figure 4: Diameter and biconnected components

For the diameter, we see in Figure 4(a) a much larger variance than for the other metrics. The hybrid approach captures the diameter fairly accurately until AS topology takes a jump at the end of 2008. As also seen in Beichl and Cloteaux (2008b), optimizing over the S-metric does seem to capture the diameter fairly well.

3.4 Biconnected Components

A biconnected component in a graph is a maximal set of edges such that any two edges in the set are on some cycle in B. Since for any node in a biconnected component with size greater than one to be disconnected requires cutting at least two edges in the network, the number of biconnected components is a measure of how much edge redundancy a graph has. Computing the number of biconnected components can be accomplished in time linear to the number of edges in the graphs (Tarjan 1972).

When comparing the number of biconnected components in Figure 4(b), we see that the shape from the models captures the shape from the AS topology very well. In this instance, the hybrid approach gave a good approximation over the two year period.

3.5 Node Cover

The minimum node cover of a graph is a minimum set of nodes such that every edge in the graph is adjacent to at least one node in this set. The size of the minimum node cover is a measure of compactness of the network. We can think of this metric as measuring the smallest number of nodes we would need to monitor in a network to ensure the reliability of all the connections.

While the problem of finding a minimum node cover is NP-hard (Garey and Johnson 1990), there are efficient strategies for solving many specific instances. In particular, we used the kernelization techniques of Abu-Khzam et al. (2004) in order to simplify the problem. The idea of kernelization algorithms is to simplify the graph G to produce a smaller graph which is called the kernel graph. A constant k is also generated such that the minimum node cover of the original graph equals the minimum node cover of the kernel graph plus k. In the case of the graphs we examined, the exact minimal node cover for the resulting kernel graph was trivial to compute. In every instance, the resulting kernel was under 20 nodes and so could be solved exactly by brute force methods.

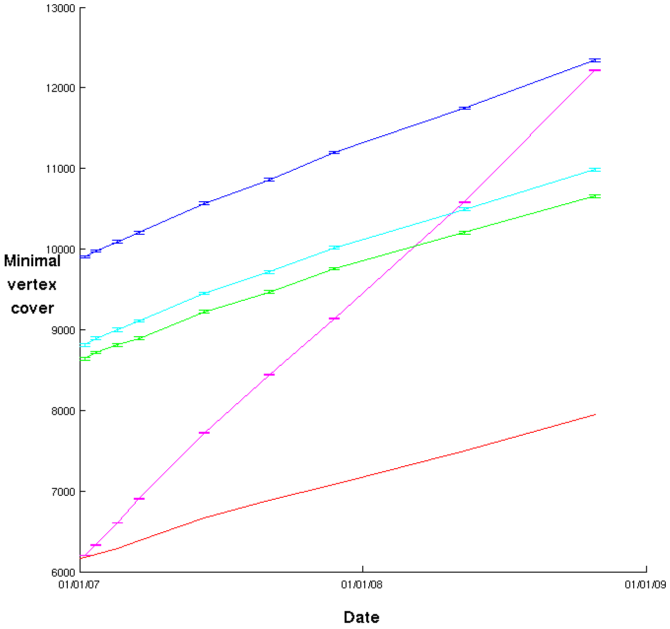


Figure 5: Minimal node cover

In Figure 5, we see that the modeling algorithms capture the correct shape of the AS topology, but there is a fairly large difference in the actual values. We speculate that this difference is more a result of optimizing using the S-metric, since it was shown in Beichl and Cloteaux (2008b) that S-metric does a poor job in reproducing minimal node cover.

4 CONCLUSIONS AND FUTURE WORK

Analysis of our plots shows that our hybrid structural approach seems to capture the correct shape of the AS topology for the various metrics we measured. In addition, for the most part the hybrid structural approach also tend to give the closest approximations to the values of the AS topology. Where there are large differences (such as clustering coefficient), it seems to be more of an artifact of optimizing the graph model based strictly on the S-metric. It was shown in Beichl and Cloteaux (2008b) that certain structural quantities are not captured well using the S-metric.

We believe that this approach is very promising for creating accurate temporal models. What is needed is a better understanding of which set of metrics provide better structural characteristics. As mentioned earlier, we have performed tests on much smaller networks by optimizing on both S-metric and clustering coefficient. This approach has shown great promise towards creating more accurate models. Unfortunately, the time needed for optimizing over these two constraints currently prevent it from being a practical method. A future area of study is to develop faster methods to allow optimization on greater sets of metrics.

ACKNOWLEDGMENTS

We would like to thank David Gilsinn and the anonymous referees for their many helpful comments for improving this paper. This work is a contribution of NIST, an agency of the US government, and is not subject to US copyright.

REFERENCES

Abu-Khzam, F. N., R. L. Collins, M. R. Fellows, M. A. Langston, W. H. Suters, and C. T. Symons. 2004. Kernelization algorithms for the vertex cover problem: Theory and experiments. In *Proceedings of the Sixth Workshop on Algorithm Engineering and Experiments and the First Workshop on Analytic Algorithmics and Combinatorics (ALENEX/ANALC)*, ed. L. Arge, G. F. Italiano, and R. Sedgewick, 62–69: SIAM.

Alderson, D. L., and L. Li. 2007, April. Diversity of graphs with highly variable connectivity. *Physical Review E* 75 (4): Art. No. 046102.

Amaral, L. A., A. Scala, M. Barthelemy, and H. E. Stanley. 2000, October. Classes of small-world networks. *Proceedings of the National Academy of Sciences of the United States of America* 97 (21): 11149–11152.

Barabási, A.-L., and R. Albert. 1999. Emergence of scaling in random networks. *Science* 286:509–512.

Beichl, I., and B. Cloteaux. 2008a. Generating network models using the *S*-metric. In *Proceedings of the International Conference on Modeling, Simulation and Visualization Methods*, ed. H. R. Arabnia and L. Deligiannidis, 159–164.

Beichl, I., and B. Cloteaux. 2008b. Measuring the effectiveness of the *s*-metric to produce better network models. In *Proceedings of the 2008 Winter Simulation Conference*, ed. S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, and J. W. Fowler, 1020–1028. Piscataway, NJ: Institute of Electrical and Electronics Engineers, Inc.

Blitzstein, J. 2007. Program for generating random graphs with prescribed degrees. http://www.people.fas.harvard.edu/~blitz/Site/Research_files/GraphAlgorithmR.txt. Accessed 10/12/2007.

Blitzstein, J., and P. Diaconis. 2006. A sequential importance sampling algorithm for generating random graphs with prescribed degrees. Submitted to *Annals of Applied Probability*.

Dueck, G., and T. Scheuer. 1990. Threshold accepting: a general purpose optimization algorithm appearing superior to simulated annealing. *Journal of Computational Physics* 90 (1): 161–175.

Faloutsos, M., P. Faloutsos, and C. Faloutsos. 1999. On power-law relationships of the internet topology. In *Proceedings of the 1999 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)*, 251–262. New York, NY, USA: ACM.

Garey, M. R., and D. S. Johnson. 1990. *Computers and intractability; a guide to the theory of NP-completeness*. New York, NY, USA: W. H. Freeman & Co.

Gkantsidis, C., M. Mihail, and E. Zegura. 2003. The Markov chain simulation method for generating connected power law random graphs. In *Proceedings of the Fifth Workshop on Algorithm Engineering and Experiments (ALENEX)*, ed. R. E. Ladner, 16–25: SIAM.

Hakimi, S. 1962. On realizability of a set of integers as degrees of the vertices of a linear graph. I. *Journal of the Society for Industrial and Applied Mathematics* 10 (3): 496–506.

Jeong, H., B. Tombor, R. Albert, Z. Oltvai, and A. L. Barabási. 2000, October. The large-scale organization of metabolic networks. *Nature* 407 (6804): 651–654.

Li, L., D. Alderson, J. C. Doyle, and W. Willinger. 2005. Towards a theory of scale-free graphs: Definition, properties, and implications. *Internet Mathematics* 2 (4): 431–523.

Mihail, M., and N. Vishnoi. 2002. On generating graphs with prescribed degree sequences for complex network modeling applications. In *Proceedings of the Third Workshop on Approximation and Randomized Algorithms for Communication Networks (ARACNE)*, 1–10.

Oliveira, R. V., B. Zhang, and L. Zhang. 2007. Observing the evolution of Internet AS topology. In *Proceedings of the 2007 conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM)*, 313–324. New York, NY, USA: ACM.

Potanin, A., J. Noble, M. Frean, and R. Biddle. 2005. Scale-free geometry in OO programs. *Communications of the ACM* 48 (5): 99–103.

Tarjan, R. 1972. Depth-first search and linear graph algorithms. *SIAM Journal on Computing* 1 (2): 146–160.

UCLA 2009. UCLA Internet topology collection. http://irl.cs.ucla.edu/topology. Accessed 2/12/2009.

Viger, F., and M. Latapy. 2005. Efficient and simple generation of random simple connected graphs with prescribed degree sequence. In *Proceedings of the Eleventh International Computing and Combinatorics Conference (COCOON)*, ed. L. Wang, Volume 3595 of *Lecture Notes in Computer Science*, 440–449: Springer.

Watts, D. J., and S. H. Strogatz. 1998. Collective dynamics of ‘small-world’ networks. *Nature* 393:440–442.

Williams, R. J., and N. D. Martinez. 2000, March. Simple rules yield complex food webs. *Nature* 404 (6774): 180–183.

AUTHOR BIOGRAPHIES

ISABEL BEICHL is a mathematician in the Mathematical and Computational Sciences division at the National Institute of Standards and Technology. She holds a PhD degree in Mathematics from Cornell University. Her email address is <isabel.beichl@nist.gov>.

BRIAN CLOTEAUX is a computer scientist in the Mathematical and Computational Sciences division at the National Institute of Standards and Technology. He holds a PhD degree in Computer Science from New Mexico State University. His email address is <brian.cloteaux@nist.gov>.