# A Basis for Systematic Analysis of Network Topologies

*Priya Mahadevan*[*] *Dmitri Krioukov*[†] *Kevin Fall*[‡] *Amin Vahdat*[§]

## Abstract

This paper presents a new, systematic approach to analyzing network topologies. We first introduce a series of probability distributions specifying all degree correlations within $d$-sized subgraphs of a given graph $G$. Using this series, we can quantitatively evaluate how close synthetic topologies are to $G$, construct graphs that accurately reproduce the values of commonly-used graph metrics of $G$, and provide a rigorous basis for capturing any future metrics that may be of interest. The $d = 0$ and $d = 1$ cases reduce to the known classical (Erdős-Rényi) random graphs and random graphs with prescribed degree distributions respectively. However, recent research shows that simply reproducing a graph's degree distribution is insufficient for capturing important properties of network topologies. Using our approach, we construct graphs for $d = 0, 1, 2, 3$ and demonstrate that these graphs reproduce, with increasing accuracy, important properties of measured and modeled Internet topologies. We find that the $d = 2$ case is sufficient for most practical purposes, while $d = 3$ essentially reconstructs the Internet AS- and router-level topologies exactly. Overall, the availability of a *systematic* method to analyze and synthesize topologies offers a significant improvement to the set of tools available to network topology and protocol researchers.

## 1 Introduction

Knowledge of network topology is crucial for understanding and predicting the performance, robustness, and scalability of network protocols and applications. Routing and searching in networks, robustness to random network failures and targeted attacks, the speed of virus spreading, and common strategies for traffic engineering and network management all depend on the topological characteristics of a given network.

[*]UCSD, pmahadevan@cs.ucsd.edu
[†]CAIDA, dima@caida.org
[‡]Intel Research, kevin.fall@intel.com
[§]UCSD, vahdat@cs.ucsd.edu

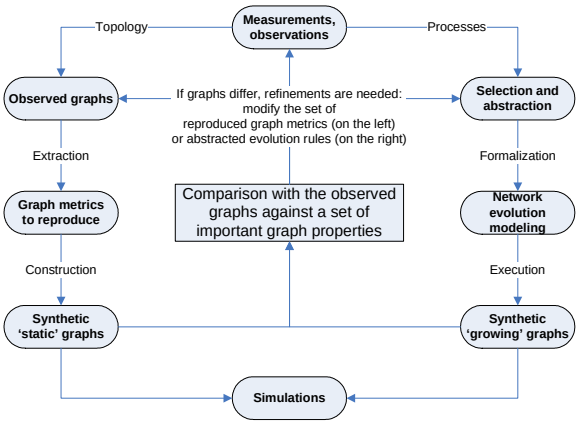



Figure 1: Methodologies of network topology research.

Research involving network topology, particularly Internet topology, generally investigates the following questions:

1. generation: can we efficiently generate ensembles of random but "realistic" topologies by reproducing a set of simple graph metrics?
2. simulations: how does some (new) protocol or application perform on a set of these "realistic" topologies?
3. evolution: what are the forces driving the evolution (growth) of the topology of a given network?

Figure 1 illustrates the methodologies used to answer these questions in its left, bottom, and right parts, respectively. Common to all of the methodologies is a set of practically-important graph properties used for analyzing and comparing sets of graphs at the center box of the figure. Many such properties have been defined and explored in the literature. We briefly discuss some of them in Section 2. Unfortunately, there are no known algorithms to construct random graphs with given values of most of these properties, since they typically characterize the global structure of the topology, making it difficult or impossible to algorithmically reproduce them.

This paper introduces a finite set of reproducible graph properties, the $dK$-*series*, to describe and constrain random graphs in successively finer detail. In the limit, these properties describe any given graph completely. In our model, we make use of probability distributions on the subgraphs of size $d$ in some given input graph. We call $dK$-*graphs* the sets of graphs constrained by such distributions. Producing a family of $0K$-graphs for a given input graph requires reproducing only the *average* node degree of the original graph, while producing a family of $1K$-graphs requires reproducing the original graph's node degree *distribution*. $2K$-graphs reproduce the *joint* degree distribution of the original graph as well—the probability that two nodes of degrees $k$ and $k'$ are connected. $3K$-graphs consider triples of nodes, and so forth. Generally, the set of $(d+1)K$-graphs is a subset of $dK$-graphs. That is, larger values of $d$ further constrain the number of possible graphs. As we shall see, generating $dK$-graphs becomes increasingly computationally difficult as $d$ increases.

A key contribution of this paper is to define the $dK$-graphs and the probability distributions we can employ for generating and analyzing network topology graphs. Specifically, we develop and implement new algorithms for constructing $2K$- and $3K$-graphs—algorithms to generate $0K$- and $1K$-graphs are already known.

Using a variety of measured and model Internet AS- and router-level topologies as input, we use our graph generators to show how the space of $dK$-graphs matching the measured property of the original graph becomes increasingly constrained as we increase $d$. Further, as we increase $d$, our randomly generated graphs capture an increasingly larger set of important graph properties proposed in the topology generation and analysis literature. We find that for the class of input graphs we consider, reproducing a graph's $3K$ distribution is sufficient to accurately reproduce *all* graph properties we have encountered so far.

While we cannot yet make claims about the generality of our approach, our initial experience suggests that the $dK$-series has the potential to deliver two key benefits. First, it can unify a variety of proposed graph metrics in the literature. Second, it enables the construction of random graphs matching complex graph properties, beyond the simple per-node properties considered by current topology generators.

## 2 Important Topology Metrics

In this section we outline a list of graph metrics that have been found important in the networking literature. This list is not complete, but we believe it is sufficiently diverse and comprehensive to be used as a good indicator of graph similarity in subsequent sections. Further, the accuracy with which we can reproduce important metrics is of practical benefit in its own right.

The *spectrum* of a graph is the set of eigenvalues of its Laplacian $\mathcal{L}$. The matrix elements of $\mathcal{L}$ are $\mathcal{L}_{ij} = \mathcal{L}_{ji} = -1/(k_i k_j)^{1/2}$ if there is a link between a $k_i$-degree node $i$ and a $k_j$-degree node $j$, and 0 otherwise. All the eigenvalues lie between 0 and 2. Of particular importance are the smallest non-zero and largest eigenvalues, $\lambda_1$ and $\lambda_{n-1}$, where $n$ is the graph size. These eigenvalues provide tight bounds for a number of critical network characteristics [6] including *network resilience* from [24] and *network performance* from [14], i.e., the maximum traffic throughput of the network.

The *distance distribution* $d(x)$ is the number of pairs of nodes at a distance $x$, divided by the total number of pairs $n^2$ (self-pairs included). This metric is a normalized version of *expansion* from [23]. It is also important for evaluating the performance of routing algorithms [13] as well as of the speed of the spread of viruses in a network.

*Betweenness* is the most commonly used measure of centrality, i.e., topological importance, both for nodes and links. It is a weighted sum of the number of shortest paths passing through a given node or link. As such, it estimates the potential traffic load on a node or link, assuming uniformly distributed traffic following shortest paths. Metrics such as *link value* in [24] or *router utilization* in [14] are directly related to betweenness.

The most widely known graph property is probably the *node degree distribution* $P(k)$, specifying the number of nodes of degree $k$ in a graph. The unexpected finding in [9] that the degree distributions in Internet topologies closely follow power laws was one of the main results behind the recent surge in topology research.

The *likelihood* $S$ [14] is the sum of products of degrees of adjacent nodes. It is linearly related to the *assortativity coefficient* $r$ suggested in [19] as a summary statistic of node interconnectivity: assortative (disassortative) networks are those where nodes with similar (dissimilar) degrees tend to be tightly interconnected. They are more (less) robust to both random and targeted removals of nodes and links. L. Li *et al.* uses $S$ in [14] as a measure of graph randomness to show that router-level topologies are not "very random": instead, they are the result of sophisticated engineering design.

*Clustering* $C(k)$ is a measure of how close neighbors of the average $k$-degree node are to forming a clique: $C(k)$ is the ratio of the average number of links between the neighbors of $k$-degree nodes to the maximum number of such links $\binom{k}{2}$. If two neighbors of a node are connected, then these three nodes form a triangle (3-cycle). Therefore, by definition, $C(k)$ is the average number of 3-cycles involving $k$-degree nodes. T. Bu and D. Towsley [3] employ clustering to estimate accuracy of topology generators. More recently, P. Fraigniaud [10] finds that a wide class of searching/routing strategies are more efficient on strongly clustered networks.

## 3 $dK$-series and $dK$-graphs

There are several problems with the graph metrics we have discussed thus far. First, they derive from a wide range of studies, and no one has established a systematic way to determine which metrics should be used in a given scenario. Second, there are no known algorithms capable of constructing graphs with desired values of most of the described metrics. Thus, while it is possible to determine whether two input graphs have, for example, similar clustering, it is currently not possible to generate graphs that precisely reproduce a required from of clustering. In the same vein, spectrum, distance distribution, and betweenness characterize global graph structure, while known graph-generating algorithms reproduce only local graph properties. Third, this list of metrics is incomplete. In particular, it cannot include any future metrics that may be of interest. Identifying such a metric might result in finding that known synthetic graphs do not match this new metric's value: moving along the loops in Figure 1 can thus continue forever.

To address these problems, we focus on establishing an enumerable set of related properties that can form the basis for any topological graph study. More precisely, for any graph $G$, we wish to identify a *series* of graph properties $\mathcal{P}_d$, $d = 0, 1, \ldots$, satisfying the following requirements:

1. *constructibility*: we can construct graphs having these properties;

2. *inclusion*: any property $\mathcal{P}_d$ subsumes all properties $\mathcal{P}_i$ with $i = 0, \ldots, d-1$: that is, a graph having property $\mathcal{P}_d$ is guaranteed to also have all properties $\mathcal{P}_i$ for $i < d$;

3. *convergence*: as $d$ increases, the set of graphs having property $\mathcal{P}_d$ "converges" to $G$: that is, there exists a value of index $d = D$ such that all graphs having property $\mathcal{P}_D$ are isomorphic to $G$.

We begin by establishing our construction of the properties $\mathcal{P}_d$, which we will call the *$dK$-series*. Our first two properties $\mathcal{P}_0$ and $\mathcal{P}_1$ are essentially a graph's average node degree and distribution of node degrees, respectively. While these properties have been relatively well-studied, the properties with larger values of $d$ have not. In the following sections, we demonstrate how to use our set of properties to both analyze and construct graphs, and demonstrate that graph generators based on this approach can effectively produce synthetic graphs that emulate measured graphs in terms of the common metrics of Section 2. The three properties of constructibility, inclusion, and convergence above give us confidence that our approach generalizes to additional graph metrics that may be proposed in the future.

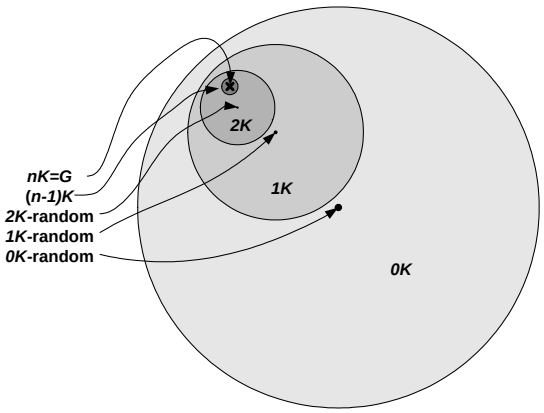


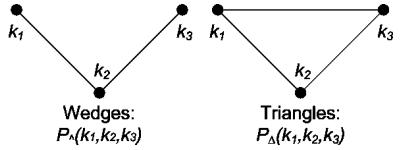

Figure 2: **The $dK$- and $dK$-random graph hierarchy.** The circles represent $dK$-graphs, whereas their centers represent $dK$-random graphs. The cross is the $nK$-graph isomorphic to a given graph $G$.

The most basic properties of a network topology characterize connectivity. The coarsest such property is the *average node degree* $\bar{k} = 2m/n$, where $n = |V|$ and $m = |E|$ are the numbers of nodes and links in a graph $G(V, E)$. Therefore, the first property $\mathcal{P}_0$ in our $dK$-series $\mathcal{P}_d$ is the average degree $\bar{k}$. In Figure 2 we schematically depict the set of all graphs with a given value of $\mathcal{P}_0$ as the $0K$-graphs, defining the largest circle. Generalizing, we adopt the term *$dK$-graphs* to represent the set of all graphs having a given value of property $\mathcal{P}_d$.

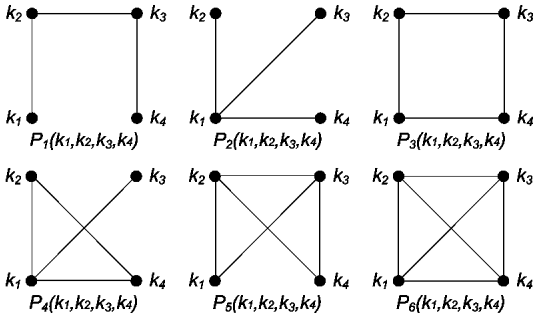
The $\mathcal{P}_0$ property tells us the average number of links per node, but it does not tell us the distribution of degrees across nodes. In particular, we do not know the number of nodes $n(k)$ of each degree $k$ in the graph. We define property $\mathcal{P}_1$ to capture this information: $\mathcal{P}_1$ is therefore the *node degree distribution* $P(k) = n(k)/n$. Knowing $\mathcal{P}_1$ implies knowing at least as much information about the network as knowing $\mathcal{P}_0$, but not vice versa. Using $P(k)$ we find $\mathcal{P}_0$ as $\bar{k} = \sum kP(k)$. $\mathcal{P}_1$ contains more information than $\mathcal{P}_0$, and it is therefore a more restrictive metric: the set of $1K$-graphs is a subset of the set of $0K$-graphs. Figure 2 illustrates this inclusive relationship by drawing the set of $1K$-graphs inside the set of $0K$-graphs.

Continuing on for the next value of $d = 2$, we note that the degree distribution tells us how many nodes of each degree are in the network, but it does not tell us how nodes of different degrees interconnect. That is, it does not provide any information on the total number $m(k, k')$ of links between nodes of degree $k$ and $k'$. We define the third property $\mathcal{P}_2$ in our series of properties as the *joint degree distribution* (JDD) $P(k_1, k_2) = m(k_1, k_2)\mu(k_1, k_2)/(2m)$, where $\mu(k_1, k_2)$ is 2 if $k_1 = k_2$ and 1 otherwise. By definition, the JDD describes degree correlations for *pairs* of connected nodes. Given $P(k_1, k_2)$, we can calculate $P(k)$, but not vice versa: $P(k) = (\bar{k}/k) \sum_{k'} P(k, k')$. Consequently, the set of $2K$-graphs is a subset of the $1K$-graphs. Therefore, in Figure 2 we depict the smaller $2K$-graph circle inside $1K$.

We can continue to increase the amount of connectivity information by considering degree correlations among greater numbers of connected nodes. To do this we must begin to consider the various geometries that are possible in interconnecting $d$ nodes. To define $\mathcal{P}_3$, we require the following two components: 1) *wedges*: chains of 3 nodes connected by 2 edges, called the $P_\wedge(k_1, k_2, k_3)$ component; and 2) *triangles*: cliques of 3 nodes, called the $P_\triangle(k_1, k_2, k_3)$ component:

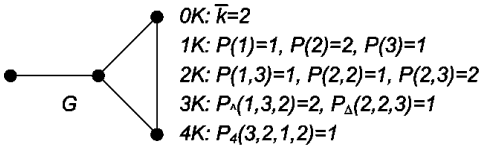



As two arrangements occur with differing frequencies among nodes of differing degree, we require a separate probability distribution for each configuration. For $\mathcal{P}_4$, we need the following six distributions:




where instead of indices $\wedge, \triangle$ we use for $d = 3$, we have all non-isomorphic graphs of size $4$ numbered by $1, \ldots, 6$. We note that the order of $k$-arguments generally matters, although we can permute any pair of arguments corresponding to pairs of nodes whose swapping leaves the graph isomorphic. For example: $P_\wedge(k_1, k_2, k_3) \neq P_\wedge(k_2, k_1, k_3) \neq P_\wedge(k_1, k_3, k_2)$, but $P_\wedge(k_1, k_2, k_3) = P_\wedge(k_3, k_2, k_1)$.

In the following figure, we illustrate properties $\mathcal{P}_d$, $d = 0, \ldots, 4$, calculated for a given graph $G$ of size $4$:




where for simplicity, values of all distributions $P$ are the total numbers of corresponding subgraphs, i.e., $P(2, 3) = 2$ means that $G$ contains 2 edges between 2- and 3-degree nodes.

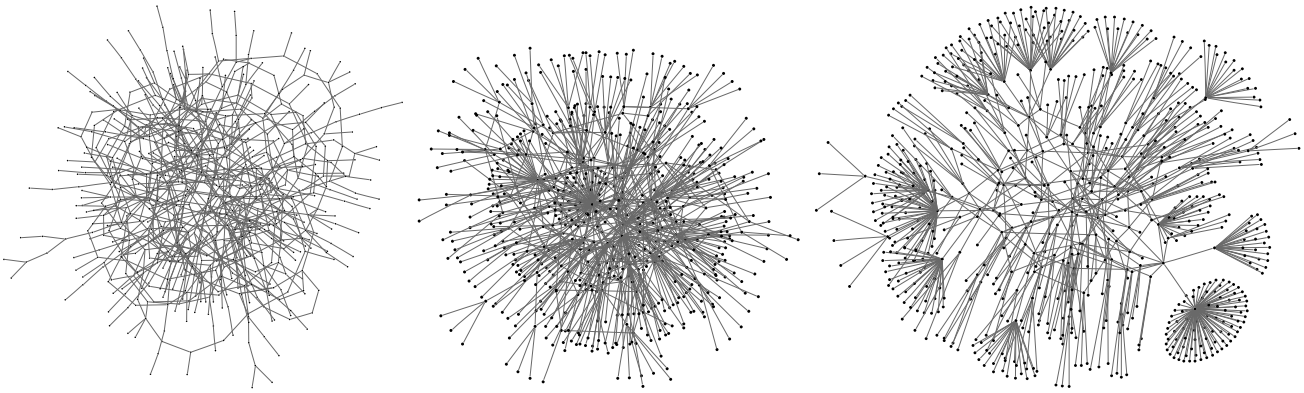
Generalizing, *we define the $dK$-series $\mathcal{P}_d$ to be degree correlations within non-isomorphic simple connected subgraphs of size $d$.* In other words, $\mathcal{P}_d$ tells us how groups of $d$-nodes with degrees $k_1, ..., k_d$ interconnect. Moving from $\mathcal{P}_d$ to $\mathcal{P}_{d+1}$ in describing a given graph $G$ is somewhat similar to including the additional $d + 1$'th term of the Fourier (time) or Taylor series representing a given function $F$. In both cases, we describe wider "neighborhoods" in $G$ or $F$ to achieve a more accurate representation of the original structure.

The $dK$-series definition satisfies the inclusion and convergence requirements described above. Indeed, the inclusion requirement is satisfied because any graph of size $d$, i.e., having $d$ nodes, is a subgraph of some graph of size $d + 1$. Convergence follows from the observation that in the limit of $d = n$, the set of $nK$-graphs contains only one element: $G$ itself. As a consequence of the convergence property, any topology metric we can define on $G$ will eventually be captured by $dK$-graphs with a sufficiently large $d$. Indeed, if two graphs are isomorphic, then the corresponding values of all their metrics are identical.

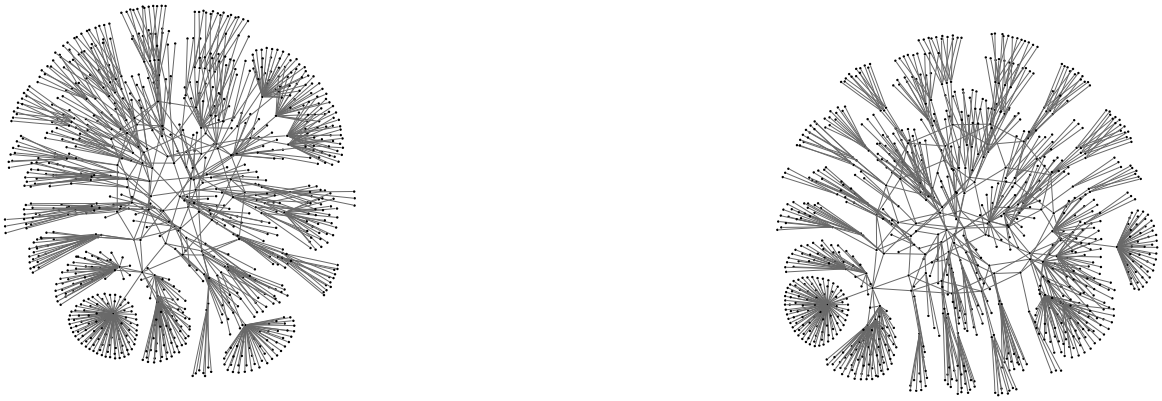
Hereafter, our main concerns with the $dK$-series become: 1) how well our first requirement, constructibility, is satisfied, and 2) how fast the convergence of $dK$-series takes place. We address these two concerns in Sections 4 and 5.

The reason for the second concern is that the complexity of $\mathcal{P}_d$ associated with the number of probability distributions required to fully define $\mathcal{P}_d$ grows quickly with $d$: see [22] for the number of non-isomorphic simple connected graphs of size $d$. Relative to the existing work on topology generators, limited to the maximum value of $d = 1$ [16, 26, 1] to the best of our knowledge, we present and implement algorithms for graph construction in the $d = 2$ and $d = 3$ cases. Having implemented these algorithms in Section 4, we find in Section 5 that convergence of the $dK$-series is fast: $2K$-graphs are sufficient for most practical purposes for the graphs we consider, while the $3K$-graphs are essentially identical to observed and modeled Internet topologies.

To motivate our ability to capture increasingly complex graph properties by increasing $d$, we present visualizations of $dK$-graphs generated using the $dK$-randomizing approach we will discuss in Section 4.1.4. Figure 3 depicts random $0K$-, $1K$-, $2K$- and $3K$-graphs matching the corresponding distributions of the HOT graph, a representative router-level topology from [14]. This topology is a particularly interesting case, because, to date reproducing router-level topologies using only degree distributions has proven difficult [14] since router-level topologies are more engineered and consequently "less random." However, a visual inspection of our generated topologies shows good convergence properties of the $dK$-series, with the $0K$-graph and $1K$-graph very far from the original HOT topology, the $2K$-graph much closer than the previous ones and the $3K$ graphs almost identical to the original. While the visual inspection is encouraging, we defer more careful comparisons to Section 5.



(a) $0K$-graph (b) $1K$-graph (c) $2K$-graph



(d) $3K$-graph (e) original HOT graph

Figure 3: **Picturizations of $dK$-graphs and the original HOT graph illustrating the convergence of $dK$-series.**

## 4 Constructing $dK$-graphs

There are several approaches for constructing $dK$-graphs for $d = 0$ and $d = 1$. We extended a number of these algorithms to work for higher values of $d$. In Section 4.1, we describe these approaches, their practical utility, and our new algorithms for $d > 1$. In Section 4.2, we introduce the concept of *$dK$-random graphs* that we will need to propose, in Section 4.3, a *$dK$-space exploration* methodology. We use this methodology to determine the lowest values of $d$ such that $dK$-graphs approximate a given topology with the required degree of accuracy.

### 4.1 $dK$-graph-constructing algorithms

We classify existing approaches to constructing $0K$- and $1K$-graphs into the following categories: *stochastic*, *pseudograph*, *matching*, and two types of *rewiring*: *randomizing* and *targeting*. We attempted to extend each of these techniques to general $dK$-graph construction. In this section, we qualitatively discuss the relative merits of each of these approaches before presenting a more quantitative comparison in Section 5.

#### 4.1.1 Stochastic

The simplest and most convenient for theoretical analysis is the stochastic approach. For $0K$, reproducing an $n$-sized graph with a given expected average degree $\bar{k}$ involves connecting every pair of $n$ nodes with probability $p_{0K} = \bar{k}/n$. This construction forms the classical (Erdős-Rényi) random graphs $\mathcal{G}_{n,p}$ [8]. It has recently been extended for $1K$ and $2K$ in [5] and [2, 7] respectively. In these cases, one first labels all nodes $i$ with their expected degrees $q_i$ drawn from the distribution $P(k)$ and then connects pairs of nodes $(i, j)$ with probabilities $p_{1K}(q_i, q_j) = q_i q_j/(n\bar{q})$ or $p_{2K}(q_i, q_j) = (\bar{q}/n)P(q_i, q_j)/(P(q_i)P(q_j))$ reproducing the expected values of $\mathcal{P}_1$ or $\mathcal{P}_2$, respectively.

In theory, we could generalize this approach for any $d$ in two stages: 1) *extraction*: given a graph $G$, calculate the frequencies of all (not only connected!) $d$-sized sub-

graphs in $G$, and 2) *construction*: prepare an $n$-sized set of $q_i$-labeled nodes and connect their $d$-sized subsets into different subgraphs with probabilities based on the calculated frequencies. In practice, we found the stochastic approach performs poorly even for $1K$ because of high degree variance. For example, nodes with expected degree 1 often wind up as degree 0 after completing the construction phase, leaving many small connected components.

#### 4.1.2 Pseudograph

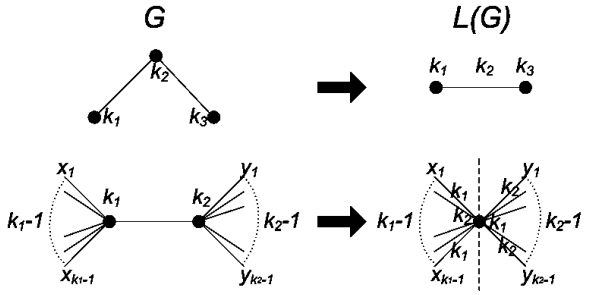
The pseudograph (also known as *configuration*) approach is probably the most popular and widely used class of graph-generating algorithms. It is at the core of well-established topology generators such as BRITE [16] and PLRG [1]. In its original form [18, 1], it applies only to the $1K$ case. As opposed to the stochastic approach, it reproduces a given degree distribution exactly, but does not necessarily construct simple graphs. That is, it may construct graphs with both ends of an edge connected to the same node (self-loops) and with multiple edges between the same pair of nodes (loops).

It works as follows: given the number of nodes, $n(k)$, of degree $k$, $n = \sum_{k=1}^{k_{\max}} n(k)$, first prepare $n(k)$ nodes with $k$ stubs attached to each node, $k = 1, \ldots, k_{\max}$, and then randomly choose pairs of stubs and connect them to form edges. To obtain a simple connected graph, remove all loops and extract the largest connected component.

We extended this algorithm to $2K$ as follows: given the number $m(k_1, k_2)$ of edges between $k_1$- and $k_2$-degree nodes, $m = \sum_{k_1,k_2=1}^{k_{\max}} m(k_1, k_2)$, first prepare a list of $m(k_1, k_2)$ disconnected edges with edge-ends labeled by values $k_1$ and $k_2$, $k_1, k_2 = 1, \ldots, k_{\max}$. Next, from the list of all edge-ends labeled with $k$, randomly select groups of $k$ edge-ends to create nodes having degree $k$, $k = 1, \ldots, k_{\max}$. While this pseudograph technique produced good results for $k = 2$, this algorithm unfortunately cannot be generalized easily for $d > 2$ because $d$-sized subgraphs overlap starting at $d = 3$. Such overlapping introduces a series of topological constraints, and we could not find a technique to preserve these constraints during the construction phase.

To demonstrate the overlap problem, we consider the simplest $d = 3$ case. We first note that the $3K$ pseudograph algorithm applied to graph $G$ is equivalent to the following modification of the $2K$ pseudograph algorithm applied to $G$'s line graph $L(G)$. Recall that a line graph $L(G)$ [20] is a graph whose vertices are edges of $G$ and two vertices of $L(G)$ are connected *iff* the two corresponding edges in $G$ are adjacent. Note that an edge between $k_1$- and $k_2$-degree nodes in $G$ (a $(k_1, k_2)$-edge) maps to a node of degree $k_1 + k_2 - 2$ in $L(G)$. We need more detailed degree information preserved in $L(G)$: we label every edge in $L(G)$ with degree triplets $(k_1, k_2, k_3)$ indicating that a pair of $(k_1, k_2)$- and $(k_2, k_3)$-edges are connected over their $k_2$-ends ($k_2$-degree node) in $G$. Every $(k_1, k_2, k_3)$-labeled edge has the two ends which we refer to as $(k_1, k_2)$- and $(k_3, k_2)$-ends, i.e., $k_2$ labels the middle of the edge. We thus map $(k_1, k_2, k_3)$-wedges in $G$ to $(k_1, k_2, k_3)$-edges in $L(G)$.

Simplifying our problem and assuming for a moment that $G$ has zero clustering (no triangles), we see that every $(k_1, k_2)$-edge in $G$ participates in $k_1 - 1$ of $(x_i, k_1, k_2)$-wedges and in $k_2 - 1$ of $(k_1, k_2, y_j)$-wedges, where $x_i$ and $y_j$, $i = 1, \ldots, k_1 - 1$, $j = 1, \ldots, k_2 - 1$, are some random degrees. In $L(G)$, this observation maps to the constraint that we need $k_1 - 1$ of $(x_i, k_1, k_2)$-edges and $k_2 - 1$ of $(k_1, k_2, y_j)$-edges to connect over their, respectively, $(k_2, k_1)$- and $(k_1, k_2)$-ends to form a $(k_1 + k_2 - 2)$-degree node:




Our $3K$ algorithm for $G$ becomes the $2K$ algorithm for $L(G)$ with the following modifications. Given the number $m_\wedge(k_1, k_2, k_3)$ of $(k_1, k_2, k_3)$-wedges in $G$, first prepare $m_\wedge(k_1, k_2, k_3)$ disconnected $L(G)$ edges labeled by $(k_1, k_2, k_3)$, and then randomly select groups of $p - 1$ edge-ends labeled by $(q, p)$ and $q - 1$ edge-ends labeled by $(p, q)$ to form $(p + q - 2)$-degree nodes.

Unfortunately, the above construction does not preserve one of the basic topological constraints that $k$-degree nodes in $G$ map to $k$-cliques in $L(G)$. Indeed, this constraint means that for all $k$, every $(\cdot, k, \cdot)$-labeled edge in $L(G)$ belongs to a $k$-clique. We see that our modification of the $2K$ pseudograph algorithm for $L(G)$ does not preserve this property, and we could not find an easy way to resolve this problem.

#### 4.1.3 Matching

The matching approach differs from the pseudograph approach in avoiding loops during the construction phase itself. In the $1K$ case, the algorithm works exactly as its pseudograph counterpart but skips pairs of stubs if both stubs are connected to the same node or if the nodes to which these stubs belong to are already connected.

We extended this approach to $2K$ as follows. We label each stub with the degree of the node it is attached to, so that by connecting $k_1$- and $k_2$-labeled stubs we create an edge between $k_1$- and $k_2$-degree nodes. We then make simple modifications to ensure we reproduce the required number $m(k_1, k_2)$ of $(k_1, k_2)$-edges. As in the $1K$ case,

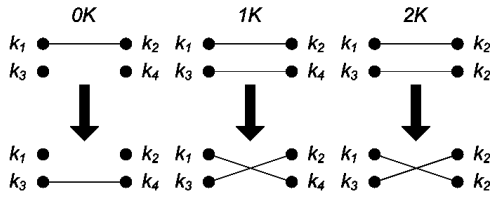



Figure 4: $dK$-**preserving rewiring for** $d = 0, 1, 2$.

we do not connect two stubs if they form a loop.

Unfortunately, such loop avoidance suffers from various forms of deadlock for both $1K$ and $2K$. The heuristics can end up in incomplete configurations when not all edges are formed, and the graph cannot be completed because there are no suitable stub pairs remaining that can be connected without forming a loop. We devised several techniques to deal with these problems.[1] With these additional techniques, we obtained good results for $2K$ graphs. Unfortunately, we could not generalize matching for $d > 2$ for essentially the same reason as in the pseudograph case. If $d = 3$, for example, then every edge is a part of many wedges and/or triangles, cf. the discussion and figure in the pseudograph subsection. Consequently, edge-forming decisions during graph construction are not independent. Each such decision simultaneously affects more than one counter of wedges and/or triangles. We found dealing with these combinatorial dependencies to be challenging.

#### 4.1.4 Rewiring

The rewiring approaches are generalizable to any $d$ and work well in practice. They involve $dK$-preserving rewiring as illustrated in Figure 4. The main idea is to rewire *random* (pairs of) edges such that the distributions corresponding to property $\mathcal{P}_d$ are preserved. For $d = 0$, we rewire a random edge to a random pair of nodes, thus preserving $\bar{k}$. For $d = 1$, we rewire two random edges as shown in the figure, which does not alter $P(k)$. If, in addition, there are at least two nodes of equal degrees adjacent to the different edges in the edge pair, then the same rewiring leaves $P(k, k')$ intact. Due to the inclusion property of the $dK$-series, $(d + 1)K$-rewirings form a subset of $dK$-rewirings for $d > 0$. For example, to preserve $3K$, we permit a $2K$-rewiring only if it also preserves the wedge and triangle distributions.

The *$dK$-randomizing rewiring algorithm* amounts to performing $dK$-preserving rewirings a sufficient number of times for some $dK$-graph. A "sufficient number" means enough rewirings for this process to lead to graphs that do not change their properties even if we subject them to additional rewirings. In other words, this rewiring process *converges* after some number of steps, producing random graphs having property $\mathcal{P}_d$. Even for $d = 1$, there are no known rigorous results proving how fast this process converges, but [11] shows that this process is an irreducible, symmetric and aperiodic Markov chain and demonstrates experimentally that it takes $O(m)$ steps to converge.

In our experiments in Section 5, we select the following strategy applicable for any $d$. We first calculate the number of possible initial $dK$-preserving rewirings. By "initial rewirings" we mean rewirings we can perform on a given graph $G$, to differentiate them from rewirings we can apply to graphs obtained from $G$ after its first (and subsequent) rewirings. We then subtract the number of rewirings that leave the graph isomorphic. For example, rewiring of any two $(1, k)$- and $(1, k')$-edges is a $dK$-preserving rewiring, for any $d$, and more strongly, the graph before rewiring is isomorphic to the graph after rewiring. We then multiply the resulting number by 10, and execute that many random rewirings. At the end of our rewiring procedure, we explicitly verify that randomization is indeed complete and the process has converged. To do that, we further increase the number of rewirings and check that all graphs' characteristics remain unchanged.

One obvious problem with our $dK$-randomizing technique is that it requires an original graph $G$ as input to construct the $dK$-random versions. It cannot start with the given distributions corresponding to property $\mathcal{P}_d$ to generate random graphs matching those distributions as is possible with the other approaches discussed above.

To address this limitation, we consider the inverse process of *$dK$-targeting $d'K$-preserving rewiring*, also known as *Metropolis dynamics* [17]. It incorporates the following modification to $d'K$-preserving rewiring: every rewiring step is accepted only if it moves the graph "closer" to $\mathcal{P}_d$. In practice, we can then employ targeting rewiring to construct $dK$-graphs with high values of $d$ by beginning with any $d'K$-graph where $d' < d$. Recall that we can always compute $\mathcal{P}_{d'}$ from $\mathcal{P}_d$ due to the inclusion property of $dK$-series. For instance, we can start with a graph having a given degree distribution ($d' = 1$) [25], and then move it toward a $dK$-graph via $dK$-targeting $1K$-preserving rewiring.

The definition of "close" above requires further explanation. We require a set of distance metrics (functions) that quantitatively differentiate two graphs based on the values of their distributions corresponding to $\mathcal{P}_d$. In our experiments, we use the sum of squares of differences between the existing and target numbers of subgraphs of a given type. For example, in the $d = 2$ case, we measure the distance between the given graph's JDD and the JDD of the current graph that we are rewiring by $\mathcal{D}_2 = \sum_{k_1,k_2} [m_{\text{current}}(k_1, k_2) - m_{\text{target}}(k_1, k_2)]^2$, and at each rewiring step, we accept the rewiring only if it decreases this distance. Note that the JDD distance is non-negative and equals zero only when reaching

[1] We omit discussion of these techniques for brevity and also since they grew fairly complex, especially for $2K$.

the target JDD. For $d = 3$, this distance $\mathcal{D}_3$ is a sum of squares of differences between the current and target numbers of wedges and triangles.

A potential problem with $dK$-targeting $d'K$-preserving rewiring is that it can be nonergodic, meaning that there might be no chain of $d'K$-preserving rewirings leading from the initial $d'K$-graph to the target $dK$-graph. In other words, we need to make sure that any two $d'K$-graphs are connected by a sequence of $d'K$-preserving rewirings.

To address this problem we note that the $d'K$-randomizing and $dK$-targeting $d'K$-preserving rewiring are actually two extremes of an entire family of rewiring processes. Indeed, let $\Delta\mathcal{D}_d = \mathcal{D}_{d,\text{after}} - \mathcal{D}_{d,\text{before}}$ be the difference of distance to the target $dK$-distribution computed before and after a $d'K$-preserving rewiring step. As with usual $dK$-targeting rewiring, we accept a rewiring step if $\Delta\mathcal{D}_d \leqslant 0$, but even if $\Delta\mathcal{D}_d \geqslant 0$, we also accept this step with probability $e^{-\Delta\mathcal{D}_d/T}$, where $T > 0$ is some parameter that we call *temperature* because of the similarity of the process to simulated annealing.

In the $T \to 0$ limit, this probability goes to 0, and we have the standard $dK$-targeting $d'K$-preserving rewiring process. When $T \to \infty$, the probability approaches 1, yielding the standard $d'K$-randomizing rewiring process. To verify ergodicity, we can start with a high temperature and then gradually cool the system while monitoring any metric known to have different values in $dK$- and $d'K$-graphs. If this metric's value forms a continuous function of the temperature, then our rewiring process is ergodic. Maslov *et al.* performed these experiments in [15] and demonstrated ergodicity in the case with $d' = 1$ and $d = 2$. In our experiments in Section 5, we always obtain a good match of all target graph metrics. Thus, we perform rewiring at zero temperature without further considering ergodicity. If however in some future experiments one detects the lack of a smooth convergence of rewiring procedures, then one should first verify ergodicity using the methodology we have just described.

For all the algorithms discussed above, we do not check for graph connectedness at each step of the algorithm since: 1) it is an expensive operation and 2) all resulting graphs always have a giant connected component (GCCs) with characteristics similar to the whole disconnected graph, as we shall see in Section 5.1.

## 4.2 $dK$-random graphs

No $dK$-graph-generating algorithm can quickly construct the set of *all* $dK$-graphs because: 1) such sets are too large, especially for small $d$; and, less obviously, 2) all algorithms try to produce graphs having property $\mathcal{P}_d$ while remaining *unbiased* (random) with respect to all other properties. One can check directly that the last characteristic applies to all the algorithms we have discussed above. Unfortunately, most construction algorithms result in non-uniform graph sampling: two different generated graphs can appear as the output of these algorithms with drastically different probabilities. Some $dK$-graphs have such a small probability of being constructed by certain algorithms that we can safely assume they never arise.

To illustrate this problem, we consider the simplest $\mathcal{P}_0$ stochastic construction, i.e., the classical random graphs $\mathcal{G}_{n,p}$. Using a probabilistic argument, one can show that the naturally-occurring $\mathcal{P}_1$ property (degree distribution) in these graphs has a specific form: binomial, which is closely approximated by the Poisson distribution: $P_{0K}(k) = e^{-\bar{k}}\bar{k}^k/k!$. Can the $0K$ stochastic algorithm produce a graph with a different $\mathcal{P}_1$, e.g., with a power-law $P(k) \sim k^{-\gamma}$? It can, but the probability of such an event is extremely low. Indeed, suppose $n \sim 10^4$, $\bar{k} \sim 5$, and $\gamma \sim 2.1$, so that the characteristic maximum degree is $k_{\max} \sim 2000$ (we chose these values to reflect measured values for Internet AS topologies). In this case, the probability that a $\mathcal{G}_{n,p}$-graph contains at least one node with degree equal to $k_{\max}$ is dominated by $1/2000! \sim 10^{-6600}$, and the probability that the remaining degrees simultaneously match those required for a power law is much lower.

It is now natural to introduce a set of graphs that correspond to the graphs most likely to be generated by $dK$-graph constructing algorithms. We call these *$dK$-random graphs*. These graphs reproduce the $\mathcal{P}_d$ property but are unbiased with respect to any other more constraining property. In this sense, the $dK$-random graphs are the *maximally random* or *maximum-entropy* $dK$-graphs. Our term *maximum entropy* here has the following justification. As we have just seen, $0K$-random graphs have the maximum-entropy value of property $\mathcal{P}_1$ since their node degree distribution is the distribution with the maximum entropy among all the distributions with a fixed average.[2] The $1K$-random graphs have the maximum-entropy value of property $\mathcal{P}_2$ since their joint degree distribution, $P_{1K}(k_1, k_2) = \tilde{P}(k_1)\tilde{P}(k_2)$, where $\tilde{P}(k) = kP(k)/\bar{k}$, is the distribution with the maximum joint entropy (minimum mutual information)[3] among all the joint distributions with fixed marginal distributions.[4]

The main point we extract from these observations is that in trying to construct $dK$-graphs, we generally obtain graphs from subsets of $dK$-graph sets. We call these sub-

[2]The entropy of a discrete distribution $P(x)$ is $H[P(x)] = -\sum_x P(x)\log P(x)$. Among all the discrete distributions with a fixed average, the binomial distribution maximizes entropy.

[3]The mutual information of a joint distribution $P(x, y)$ is $I[P(x, y)] = H[P(x)] + H[P(y)] - H[P(x, y)]$, where $P(x)$ and $P(y)$ are the marginal distributions.

[4]In reality, the last statement applies only to the class of all (not necessarily connected) pseudographs. Narrowing the class of graphs to simple connected graphs introduces topological constraints affecting the maximum-entropy form of $\mathcal{P}_2$.

Table 1: **The summary of $dK$-series.** The first column shows the property tag $dK$, $d = 0, \ldots, n$, for the property series $\mathcal{P}_d$ in the second columns. Index $d$ in $dK$ is the number of connected nodes with specified degree correlations. The third column lists the functions describing these correlations and explains the semantics behind our $dK$ notation: '$d$' in '$dK$' is the maximum *d*iameter of a $d$-sized graph and also the number of *d*egree arguments '$k$' in the correlation functions. In the $0K$, $1K$, and $2K$ case, these functions are, respectively, the average degree, degree distribution, and the joint degree distribution. In the $3K$ case, there are two independent functions specifying distribution of wedges and triangles. In the $4K$ case, there are six such functions, cf. Section 3, and so on until $d = n$, when the $dK$-series necessarily converge to a given graph $G$: property $\mathcal{P}_n$ just describes $G$ exhaustively. The fourth column illustrates the inclusion feature of the $dK$-series: property $\mathcal{P}_d$ contains more connectivity information about $G$ than $\mathcal{P}_{d-1}$, since $\mathcal{P}_d$ fully defines $\mathcal{P}_{d-1}$, but not vice versa. The fifth column lists probabilities of link existence between pairs of nodes in the stochastic $dK$-random graph constructions. Numbers $q$ are expected degrees distributed according to $P(k)$. The last column shows that, among all the $dK$-graphs, the $dK$-random graphs have specific, maximum-entropy forms of $\mathcal{P}_{d+1}$.

| Tag | Property symbol | Property value | $\mathcal{P}_d$ defines $\mathcal{P}_{d-1}$ | Edge existence probability in stochastic constructions | Max. entropy value of $\mathcal{P}_{d+1}$ in $dK$-random graphs |
|---|---|---|---|---|---|
| $0K$ | $\mathcal{P}_0$ | $\bar{k}$ | | $p_{0K} = \bar{k}/n$ | $P_{0K}(k) = e^{-\bar{k}}\bar{k}^k/k!$ |
| $1K$ | $\mathcal{P}_1$ | $P(k)$ | $k = \sum kP(k)$ | $p_{1K}(q_1, q_2) = q_1 q_2/(n\bar{q})$ | $P_{1K}(k_1, k_2) = k_1 P(k_1) k_2 P(k_2)/\bar{k}^2$ |
| $2K$ | $\mathcal{P}_2$ | $P(k_1, k_2)$ | $P(k) = (\bar{k}/k)\sum_{k'} P(k, k')$ | $p_{2K}(q_1, q_2) = (\bar{q}/n)P(q_1, q_2)/(P(q_1)P(q_2))$ | |
| $3K$ | $\mathcal{P}_3$ | $P_\wedge(k_1, k_2, k_3)$ $P_\triangle(k_1, k_2, k_3)$ | | | |
| ... | ... | ... | ... | ... | ... |
| $nK$ | $\mathcal{P}_n$ | $G$ | | | |

sets $dK$-random graphs and schematically depict them as centers of the $dK$-circles in Figure 2. These graphs do have the required values of property $\mathcal{P}_d$ and, consequently, of properties $\mathcal{P}_i$ with $i < d$, but they are unlikely to have required values of properties $\mathcal{P}_j$ with $j > d$ since these latter properties have specific, maximum-entropy values in such graphs.

## 4.3 $dK$-space explorations

Often we wish to analyze the topological constraints a given graph appears to obey. In other cases, we are interested in exploring the structure and diversity of the graphs we can generate given a particular set of properties $\mathcal{P}_d$.

If we are attempting to determine the minimum $d$ required to impose properties $\mathcal{P}_d$ upon generated graphs so that they appear similar to a given graph, we can start with a small value of $d$, generate compliant graphs, and measure their "distance" from the given graph. If the distance is too great, we can increase $d$ and repeat the process. On the other hand, if we want to explore the structural diversity among graphs having a certain property we can attempt to find the "extreme" graphs that are still constrained by $\mathcal{P}_d$.

We cannot construct all $dK$-graphs, so we need to use heuristics to generate some $dK$-graphs and adjust them according to a distance metric that draws us closer to the types of $dK$-graphs we seek. One such heuristic is based on the inclusion feature of the $dK$-series. Because all $dK$-graphs have the same values of distributions corresponding to $\mathcal{P}_d$ but not to $\mathcal{P}_{d+1}$, we look for simple metrics fixed by $\mathcal{P}_{d+1}$ but not by $\mathcal{P}_d$.

While identifying such metrics can be challenging for high $d$'s, we can always retreat to the following two simple extreme metrics:

- the correlation of degrees of nodes located at distance $d$ from each other; and
- the concentration of cliques of size $d + 1$.

These metrics are "extreme" in the sense that they correspond to the $(d + 1)$-sized subgraphs with, respectively, the maximum ($d$) and minimum (1) possible diameter.

We then try to construct $dK$-graphs with extreme values, e.g., the smallest or largest possible, for these (extreme) metrics. The $dK$-random graphs have the values of these metrics lying somewhere in between the extremes.

If the goal is to find the smallest $d$ that results in sufficiently constraining graphs, we can compute the difference between the extreme values of these metrics, as well as of other metrics we might consider. If this difference is too large, then the selected value of $d$ is not constraining enough so that we need to increase $d$. If the goal is to enumerate a large space of graphs that match a given property, then such $dK$-space exploration may be used to move beyond the relatively small circle of $dK$-random graphs.

To illustrate how this approach works in practice, we consider $1K$- and $2K$-space explorations. For $1K$, the simplest metric depending on $\mathcal{P}_2$ is any scalar summary statistics of the JDD, such as likelihood $S$ (cf. Section 2). To construct graphs with the maximum value of $S$, we can run a form of targeting $1K$-preserving rewiring that accepts each rewiring step only if it increases $S$. We can perform the opposite to minimize $S$. This type of experiment was at the core of recent work that led the authors of [14] to conclude that $d = 1$ was not constraining

enough for the topology they considered.

To perform space exploration for $2K$, we need to find simple scalar metrics depending on $\mathcal{P}_3$. Since $\mathcal{P}_3$ is actually two distributions, $P_\wedge(k_1,k_2,k_3)$ and $P_\triangle(k_1,k_2,k_3)$, we should have two independent scalar metrics. The *second-order likelihood* $S_2$ is one such metric for $P_\wedge(k_1,k_2,k_3)$. We define $S_2$ as the sum of the products of degrees of nodes located at the ends of wedges, $S_2 = \sum_{k_1,k_2,k_3} k_1 k_3 P_\wedge(k_1,k_2,k_3)$, so that any two graphs with the same $P_\wedge(k_1,k_2,k_3)$ have the same $S_2$. For the $P_\triangle(k_1,k_2,k_3)$ component, average clustering $\bar{C} = \sum_{k_1,k_2,k_3} k_1 P_\triangle(k_1,k_2,k_3)$ is an appropriate candidate. We note that these two metrics are also the two extreme metrics in the sense defined above: $S_2$ measures the properly normalized correlation of degrees of nodes located at distance 2, while $\bar{C}$ describes the concentration of 3-cliques. The $2K$-explorations amount then to performing the following two types of targeting $2K$-preserving rewiring: accept a $2K$-rewiring step only if it maximizes/minimizes: 1) $S_2$, or 2) $\bar{C}$.

# 5 Evaluation

In this section, we conduct a number of experiments to demonstrate the ability of the $dK$-series to capture important graph properties. We implemented all the $dK$-graph-constructing algorithms from Section 4.1, applied them to both measured and modeled Internet topologies, and calculated all the important topology metrics from Section 2 on the resulting graphs.

We experimented with three measured AS-level topologies, extracted from CAIDA's *skitter* traceroute [4], RouteViews' *BGP* [21], and RIPE's *WHOIS* [12] data for the month of March 2004, as well as with a synthetic router-level topology—the HOT graph from [14]. The qualitative results of our experiments are similar for the skitter and BGP topologies, while the WHOIS topology lies somewhere in-between skitter/BGP and HOT topologies. In the case of skitter, we will see that its degree distribution places significant constraints upon the graph generation process. Thus, even $1K$-random graphs approximate the skitter topology reasonably well. The HOT topology is at the opposite extreme: it is the least constrained; $1K$-random graphs approximate it poorly, and $dK$-series' convergence is slowest. We thus report results only for these two extreme cases, skitter and HOT. Our results represent averages over 100 generated graphs in each case, using the notation of Table 2.

## 5.1 Algorithmic Comparison

We first compare the different graph generation algorithms discussed in Section 4.1. All the algorithms give consistent results, except the stochastic approach, which exhibits high variance and connectivity issues discussed in Section 4.1.1. This conclusion immediately follows from Figure 5 and Tables 3 and 4 showing graph metric values for the different $2K$ and $3K$ algorithms described in Section 4.1.

Table 2: **Scalar graph metrics notations.**

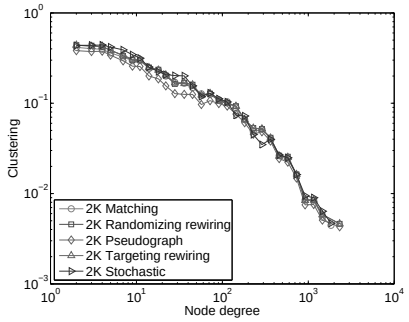
| Metric | Notation |
|---|---|
| Average degree | $\bar{k}$ |
| Assortativity coefficient | $r$ |
| Average clustering | $\bar{C}$ |
| Average distance | $\bar{d}$ |
| Standard deviation of distance distribution | $\sigma_d$ |
| Second-order likelihood | $S_2$ |
| Smallest eigenvalue of the Laplacian | $\lambda_1$ |
| Largest eigenvalue of the Laplacian | $\lambda_{n-1}$ |

Table 4: **Scalar metrics for $3K$-random HOT graphs generated using different techniques.**

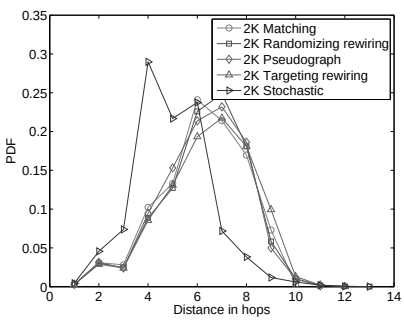
| Metric | $3K$-randomizing rewiring | $3K$-targeting rewiring | Original HOT |
|---|---|---|---|
| $\bar{k}$ | 2.10 | 2.13 | 2.10 |
| $r$ | -0.22 | -0.23 | -0.22 |
| $\bar{d}$ | 6.55 | 6.79 | 6.81 |
| $\sigma_d$ | 0.84 | 0.72 | 0.57 |

In our experience, we find that $dK$-randomizing rewiring is easiest to use. However, it requires the original graph as input. If only the target distribution associated with property $\mathcal{P}_d$ is available and if $d \leqslant 2$, we find the pseudograph algorithm most appropriate in practice. We note that our $2K$ version results in fewer loops and a larger giant connected component (GCC), than PLRG, its commonly-known $1K$ counterpart. In the skitter case for example, the $1K$ and $2K$ pseudograph algorithms generate graphs with GCC size equal to, on average, $98\%$ and $100\%$, respectively, of the original graph size, and with the number of edges after removing all loops in the graph equal to $98.5\%$ and $99.99\%$, respectively, of the number of edges in the original graph. This improvement is due to the additional constraints introduced by the $2K$ case. For example, if there is only one node of high degree $x$ and one node of another high degree $y$ in the original graph, then there can be only one link of type $(x,y)$. Our $2K$ modification of the pseudograph algorithm must consequently produce exactly one link between these two $x$- and $y$-degree nodes, whereas in the $1K$ case, the algorithm tends to create many such links. Since the original graph does not have pairs of 1-degree nodes connected to each other, our $2K$ generator, as opposed to $1K$, does not form these small 2-node CCs either.

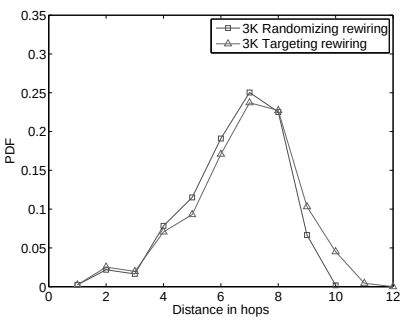
While the pseudograph algorithm is a good $2K$-random graph generator, we could not generalize it for $d \geqslant 3$, cf. Section 4.1.2. Therefore, to generate $dK$-random graphs with $d \geqslant 3$ when an original graph is unavailable, one has to use $dK$-targeting rewiring. The re-




(a) Clustering in skitter for different $2K$ algorithms




(b) Distance distribution in HOT for different $2K$ algorithms




(c) Distance distribution in HOT for different $3K$ algorithms

Figure 5: **Comparison of $2K$- and $3K$-graph-constructing algorithms.**

Table 3: **Scalar metrics for $2K$-random HOT graphs generated using different techniques.**

| Metric | Matching | $2K$-randomizing rewiring | Pseudograph | $2K$-targeting rewiring | Stochastic | Original HOT |
|---|---|---|---|---|---|---|
| $k$ | 2.22 | 2.18 | 2.19 | 2.18 | 2.87 | 2.10 |
| $r$ | -0.21 | -0.23 | -0.24 | -0.24 | -0.22 | -0.22 |
| $\bar{d}$ | 6.22 | 6.32 | 6.25 | 6.35 | 4.99 | 6.81 |
| $\sigma_d$ | 0.74 | 0.70 | 0.75 | 0.70 | 0.85 | 0.57 |

sults on $dK$-targeting rewiring in this section, i.e., in Figure 5 and Tables 3 and 4, are for graphs that we generate as follows. We first bootstrap the process by constructing $1K$-random graphs using the pseudograph algorithm and then apply to them $2K$-targeting $1K$-preserving rewiring to obtain $2K$-random graphs. To produce $3K$-random graphs, we apply $3K$-targeting $2K$-preserving rewiring to the $2K$-random graphs obtained at the previous step.

## 5.2 Topology Comparisons

We next test the convergence of our $dK$-series for the skitter and HOT graphs. We obtained all $dK$-random graphs in this study using the $dK$-randomizing rewiring strategy from Section 4.1.4, which we chose for its simplicity. The number of possible initial $dK$-randomizing rewirings is a good preliminary indicator of how constrained the $dK$-graph space is. We show these numbers for the HOT graph in Table 5. If we discard rewirings that obviously lead to isomorphic graphs, e.g., a rewiring of a 1-degree node from a $k$-degree node to another $k$-degree node, then the number of possible initial rewirings is even smaller. Note that this is a conservative estimate because we do not check for all possible graph isomorphisms, we only discard the obvious ones involving 1-degree nodes.

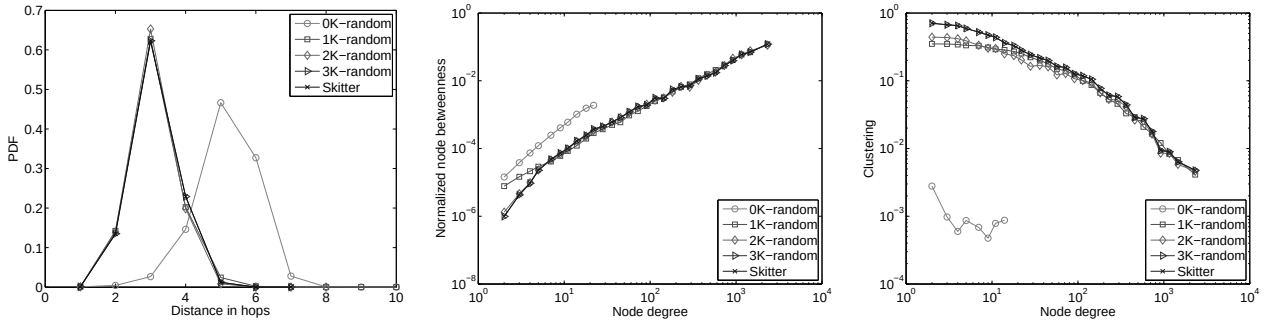
We compare the skitter topology with its $dK$-random counterparts, $d = 0, \ldots, 3$, in Table 6 and Figure 6. We report all the metrics calculated on the GCCs. Minor discrepancies between values of average degree $\bar{k}$ and $r$ result from GCC extractions. If we do not extract the

Table 5: **Numbers of possible initial $dK$-randomizing rewirings for the HOT graph.**

| $d$ | Possible initial rewirings | Possible initial rewirings, ignoring obvious isomorphisms |
|---|---|---|
| 0 | 435,546,699 | - |
| 1 | 477,905 | 440,355 |
| 2 | 326,409 | 268,871 |
| 3 | 146 | 44 |

Table 6: **Comparing scalar metrics for $dK$-random and skitter graphs.**

| Metric | $0K$ | $1K$ | $2K$ | $3K$ | skitter |
|---|---|---|---|---|---|
| $k$ | 6.31 | 6.34 | 6.29 | 6.29 | 6.29 |
| $r$ | 0 | -0.24 | -0.24 | -0.24 | -0.24 |
| $\bar{C}$ | 0.001 | 0.25 | 0.29 | 0.46 | 0.46 |
| $\bar{d}$ | 5.17 | 3.11 | 3.08 | 3.09 | 3.12 |
| $\sigma_d$ | 0.27 | 0.4 | 0.35 | 0.35 | 0.37 |
| $\lambda_1$ | 0.2 | 0.03 | 0.15 | 0.1 | 0.1 |
| $\lambda_{n-1}$ | 1.8 | 1.97 | 1.85 | 1.9 | 1.9 |




(a) Distance distribution (b) Betweenness (c) Clustering

Figure 6: **Comparison of $dK$-random and skitter graphs.**

GCC, then $\bar{k}$ is the same as that of the original graph for all $d = 0, \ldots, 3$, and $r$ is exactly the same for $d > 1$.

We do not show degree distributions for brevity. However, degree distributions match when considering the entire graph and are very similar for the GCCs for all $d > 0$. When $d = 0$, the degree distribution is binomial, as expected.

We see that all other metrics gradually converge to those in the original graph as $d$ increases. A value of $d = 1$ provides a reasonably good description of the skitter topology, while $d = 2$ matches all properties except clustering. The $3K$-random graphs are identical to the original for all metrics we consider, including clustering.

We perform the $2K$-space explorations described in Section 4.3 in order to check if $d = 2$ is indeed constraining enough for the skitter topology. We observe small variations of clustering $\bar{C}$, second-order likelihood $S_2$, and spectrum, shown in Table 7 and Figure 7. All other metrics do not change, so we do not show plots for them. We conclude that $d = 2$, i.e., the joint degree distribution, provides a reasonably accurate description of observed AS-level topologies.

The HOT topology is more complex than AS-level topologies. Earlier work argues that this topology cannot be accurately modeled using degree distributions [14]. We therefore selected the HOT topology graph as a difficult case for our approach.

A preliminary inspection of visualizations in Figure 3 indicates that the $dK$-series converge reasonably fast even for the HOT graph. The $0K$-random graph is a classical random graph and lacks high-degree nodes, as expected. It has almost nothing in common with the HOT graph. The $1K$-random graph has all the high-degree nodes we desire, but they are crowded toward the core, a property absent in the HOT graph. The $2K$ constraints start pushing the high-degree nodes away to the periphery, while the lower-degree nodes migrate the core, and the $2K$-random graph begins to resemble the HOT graph. The $3K$-random topology looks remarkably similar to the HOT topology.

Table 8: **Comparing scalar metrics for $dK$-random and HOT graphs.**

| Metric | $0K$ | $1K$ | $2K$ | $3K$ | HOT |
|---|---|---|---|---|---|
| $\bar{k}$ | 2.47 | 2.59 | 2.18 | 2.10 | 2.10 |
| $r$ | -0.05 | -0.14 | -0.23 | -0.22 | -0.22 |
| $\bar{C}$ | 0.002 | 0.009 | 0.001 | 0 | 0 |
| $\bar{d}$ | 8.48 | 4.41 | 6.32 | 6.55 | 6.81 |
| $\sigma_d$ | 1.23 | 0.72 | 0.71 | 0.84 | 0.57 |
| $\lambda_1$ | 0.01 | 0.034 | 0.005 | 0.004 | 0.004 |
| $\lambda_{n-1}$ | 1.989 | 1.967 | 1.996 | 1.997 | 1.997 |

Of course, visual inspection of a small number of randomly generated graphs is insufficient to demonstrate our ability to capture important metrics of the HOT graph. Thus, we compute the different metric values for each of the $dK$-random graph and compare them with the corresponding metric values of the original HOT graph. In Table 8 and Figures 8 and 9 we see that the $dK$-series converges more slowly for HOT than for skitter. Note that we do not show clustering plots because clustering is almost zero everywhere: the HOT topology has very few cycles; it is almost a tree. The $1K$-random graphs yield a poor approximation of the original topology, in agreement with the main argument in [14]. Both Figures 3 and 9 indicate that starting with $d = 2$, low- but not high-degree nodes form the core: betweenness is approximately as high for nodes of degree $\sim 10$ as for high-degree nodes. Consequently, the $2K$-random graphs provide a better approximation, but not nearly as good as it was for skitter. However, the $3K$-random graphs match the original HOT topology *almost exactly*. We thus conclude that the $dK$-series captures the essential characteristics of even particularly difficult topologies, such as HOT, by sufficiently increasing $d$, in this case to 3.

In general, more complex topologies may require developing algorithms for generating $dK$-random graphs for $d > 3$. However, for the class of Internet graphs that we have considered, $d = 3$ appears to be sufficient to reproduce a broad range of important graph properties.

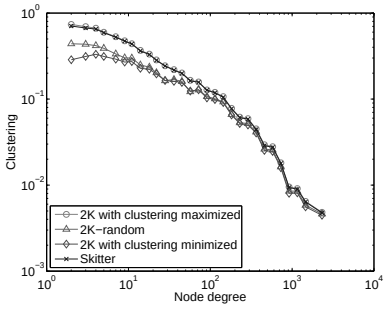



Figure 7: **Varying clustering in $2K$-graphs for skitter.**

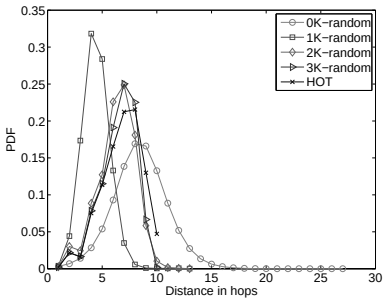



Figure 8: **Distance distribution for $dK$-random and HOT graphs**

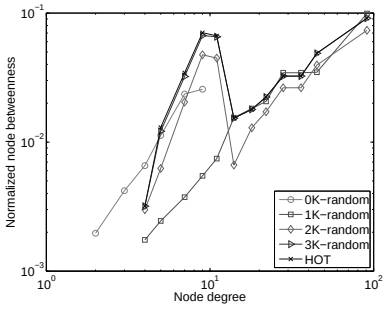



Figure 9: **Betweenness for $dK$-random and HOT graphs**

Table 7: **Scalar metrics for $2K$-space explorations for skitter.**

| Metric | Min $C$ | Max $C$ | Min $S_2$ | Max $S_2$ | $2K$-random | Skitter |
|---|---|---|---|---|---|---|
| $\bar{k}$ | 6.29 | 6.29 | 6.29 | 6.29 | 6.29 | 6.29 |
| $r$ | -0.24 | -0.24 | -0.24 | -0.24 | -0.24 | -0.24 |
| $\bar{C}$ | 0.21 | 0.47 | 0.4 | 0.4 | 0.29 | 0.46 |
| $\bar{d}$ | 3.06 | 3.12 | 3.12 | 3.10 | 3.08 | 3.12 |
| $\sigma_d$ | 0.33 | 0.38 | 0.37 | 0.36 | 0.35 | 0.37 |
| $\lambda_1$ | 0.25 | 0.11 | 0.11 | 0.1 | 0.15 | 0.1 |
| $\lambda_{n-1}$ | 1.75 | 1.89 | 1.89 | 1.89 | 1.85 | 1.9 |
| $S_2/S_2^{\max}$ | 0.988 | 0.961 | 0.955 | 1.000 | 0.986 | 0.958 |

## 6 Discussion and Future Work

While we feel our approach to topology analysis holds significant promise, a number of important avenues remain for further investigation. First, one must determine appropriate values of $d$ to carry out studies of interest. Our experience to date suggests that $d = 2$ is sufficient to reproduce most metrics of interest and that $d = 3$ faithfully reproduces all metrics of interest that we are aware of for Internet-like graphs. It also appears likely that $d = 3$ will be sufficient for small-world graphs generally. This issue is particularly important because the computational complexity of producing $dK$-graphs grows rapidly with $d$. Studies requiring large values of $d$ may limit the practicality of our approach.

A second important question concerns the discrete nature of our model. For instance, we are able to reproduce $1K$ and $2K$ distributions but it is not meaningful to consider reproducing $1.4K$ distributions. Consider a graph property, $X$, not captured by $1K$ but successfully captured by $2K$. It could turn out that the space of $2K$-random graphs that match the desired space overconstrains the set of graphs reproducing $X$. That is, while $2K$-graphs do successfully reproduce $X$, there may be other graphs that also match $X$ but are not $2K$-graphs.

A limitation fundamental to our approach is that we seek to reproduce important characteristics of a *given* network topology. We cannot use our methodology to discover fundamental laws governing the evolutionary growth of a particular network. Rather, we are restricted to observing changes in degree correlations in Internet graphs over time, and then generating graphs that match such degree correlations. However, to some extent the goals of reproducing important characteristics of a given set of graphs and discovering laws governing evolution are complimentary. For example, the availability of a large number of randomly generated graphs that match important characteristics of a time-series of original graphs representing, e.g., the Internet, may be particularly helpful in extracting common evolutionary traits and "patterns of mutation."

Directions for future work all stem from the observation that the $dK$-series is actually the simplest basis for statistical analysis of correlations in complex networks. We can incorporate any kind of technological constraints into our constructions. If working with a router-level topology, for example, we recognize that there is some dependency between the number of interfaces a router has (node degree) and their average bandwidth (betweenness/degree ratio), cf. [14], then we can simply adjust our rewiring-based algorithms from Section 4.1.4 to not accept rewirings violating this dependency. In other words, we can always consider ensembles of $dK$-random graphs subject to various forms of external constraints imposed by specifics of a given network.

Another promising avenue for future work derives from the observation that abstracting real networks as undirected graphs might be losing too much detail for cer-

tain tasks. In particular, both nodes and links can be of different types in reality. For example, in the AS-level topology case, the link types can represent business AS relationships, e.g., customer-provider, peering, etc. For a router-level topology, we can label links with bandwidth, latency, etc., and nodes with router manufacturer, geographical information, etc. Keeping such *annotation* information for nodes and links can be also useful for other types of networks, e.g., biological, social, etc. Degrees of nodes and degree pairs of edges then become the simplest form of possible annotations. We can obviously generalize the $dK$-series approach to study networks with more sophisticated forms of annotations, in which case the $dK$-series would describe correlations among different types of nodes connected by different types of links within $d$-sized geometries. Given the results of our experiments showing how constraining values of $d = 2$ and $d = 3$ are and recognizing that including annotations would introduce significant additional constraints to the space of $dK$-graphs, we believe that $2K$-random annotated graphs could provide appropriate descriptions of observed networks in a variety of settings.

## 7 Conclusions

Over the years, a number of important graph metrics have been proposed to compare how closely the structure of two arbitrary graphs match. Such metrics are employed by networking researchers involved in topology construction and analysis, and by those interested in protocol and distributed system performance. Unfortunately, there is limited understanding of which metrics are appropriate for a given setting and, for most proposed metrics, there are no known algorithms for generating graphs that reproduce the target property.

This paper defines a series of graph structural properties that can be employed in a more systematic approach to dealing with network topologies than the current set of comparatively *ad-hoc* metrics. The properties $\mathcal{P}_d$, $d = 0, \ldots, n$, comprising the $dK$-series, define a collection of distributions describing the correlations of node degrees. By increasing the value of $d$ in the series, it is possible to capture more complex properties of a given graph and, in the limit, a sufficiently large value of $d$ fully characterizes a given graph. We show that $0K$- and $1K$-graphs are two known graph classes corresponding to special instances of our properties: average node degree and degree distribution, respectively. $2K$-graphs capture relationships among pairs of nodes of given degrees; $3K$-graphs capture such relationships among node triples of given degrees, and so on. We present the first algorithms for constructing graphs matching the target properties $\mathcal{P}_2$ and $\mathcal{P}_3$, and sketch an approach for extending the algorithm to arbitrary $d$.

Along the way, we have discovered interesting trade-offs in choosing the appropriate value of $d$ to compare two graphs or to generate random graphs with a given property $\mathcal{P}_d$. As we increase $d$, the set of randomly generated graphs that match property $\mathcal{P}_d$ becomes increasingly constrained and the resulting graphs are increasingly likely to reproduce a variety of metrics of interest. At the same time, the algorithmic complexity associated with generating the graphs increases sharply. Thus, we present a methodology where practitioners choose the smallest $d$ that captures essential graph characteristics for their study. For the graphs that we consider, including comparatively complex Internet AS-level and router topologies, we find that $d = 3$ is sufficient to reproduce the original graph adequately and to capture all graph properties proposed in the literature known to us.

With this paper, we are releasing the source code for our analysis tools to measure an input graph's property $\mathcal{P}_d$ and our generator able to produce random graphs matching a given $\mathcal{P}_d$ for $d < 4$. We hope that our methodology will enable a more rigorous and consistent method of comparing topology graphs and also enable protocol and application researchers to test system behavior under a suite of randomly generated yet appropriately constrained and realistic network topologies.

## References


[1] W. Aiello, F. Chung, and L. Lu. A random graph model for massive graphs. In *ACM STOC*, 2000.

[2] M. Boguna and R. Pastor-Satorras. Class of correlated random networks with hidden variables. *Physical Review E*, 68:036112, 2003.

[3] T. Bu and D. Towsley. On distinguishing between Internet power law topology generators. In *IEEE INFOCOM*, 2002.

[4] CAIDA. Macroscopic topology AS adjacencies. `http://www.caida.org/tools/measurement/skitter/as_adjacencies.xml`.

[5] F. Chung and L. Lu. Connected components in random graphs with given degree sequences. *Annals of Combinatorics*, 6:125–145, 2002.

[6] F. K. R. Chung. *Spectral Graph Theory*, volume 92 of *Regional Conference Series in Mathematics*. American Mathematical Society, Providence, RI, 1997.

[7] S. N. Dorogovtsev. Networks with given correlations. `http://arxiv.org/abs/cond-mat/0308336v1`.

[8] P. Erdős and A. Rényi. On random graphs. *Publicationes Mathematicae*, 6:290–297, 1959.

[9] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the Internet topology. In *ACM SIGCOMM*, pages 251–262, 1999.

[10] P. Fraigniaud. A new perspective on the small-world phenomenon: Greedy routing in tree-decomposed graphs. In *ESA*, 2005.

[11] C. Gkantsidis, M. Mihail, and E. Zegura. The Markov simulation method for generating connected power law random graphs. In *ALENEX*, 2003.

[12] Internet Routing Registries. http://www.irr.net/.
[13] D. Krioukov, K. Fall, and X. Yang. Compact routing on Internet-like graphs. In *IEEE INFOCOM*, 2004.
[14] L. Li, D. Alderson, W. Willinger, and J. Doyle. A first-principles approach to understanding the Internets router-level topology. In *ACM SIGCOMM*, 2004.
[15] S. Maslov, K. Sneppen, and A. Zaliznyak. Detection of topological patterns in complex networks: Correlation profile of the Internet. *Physica A*, 333:529–540, 2004.
[16] A. Medina, A. Lakhina, I. Matta, and J. Byers. BRITE: An approach to universal topology generation. In *MASCOTS*, 2001.
[17] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. Equation-of-state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087, 1953.
[18] M. Molloy and B. Reed. A critical point for random graphs with a given degree sequence. *Random Structures and Algorithms*, 6:161–179, 1995.
[19] M. E. J. Newman. Assortative mixing in networks. *Physical Review Letters*, 89(20):208701, 2002.
[20] E. Prisner. *Graph Dynamics*. Longman, Harlow, 1995.
[21] University of Oregon RouteViews Project. http://www.routeviews.org/.
[22] N. J. A. Sloane. Sequence A001349. The On-Line Encyclopedia of Integer Sequences. http://www.research.att.com/projects/OEIS?Anum=A001349.
[23] H. Tangmunarunkit, J. Doyle, R. Govindan, S. Jamin, W. Willinger, and S. Shenker. Does AS size determine AS degree? *ACM Computer Communication Review*, October 2001.
[24] H. Tangmunarunkit, R. Govindan, S. Jamin, S. Shenker, and W. Willinger. Network topology generators: Degree-based vs. structural. In *ACM SIGCOMM*, pages 147–159, 2002.
[25] F. Viger and M. Latapy. Efficient and simple generation of random simple connected graphs with prescribed degree sequence. In *COCOON*, 2005.
[26] J. Winick and S. Jamin. Inet-3.0: Internet topology generator. Technical Report UM-CSE-TR-456-02, University of Michigan, 2002.