A Qualitative Comparison of Power Law Generators

Javier Martin Hernandez, Tom Kleiberg, Huijuan Wang and Piet Van Mieghem
Delft University of Technology
Faculty of Electrical Engineering, Mathematics & Informatics (EWI)
Mekelweg 4, 2628 CD Delft
The Netherlands
Email: {J.M.Hernandez, T.Kleiberg, H.Wang, P.VanMieghem}@ewi.tudelft.nl

Abstract

Several topology generator algorithms have been proposed to match the heavy-tailed behavior of the real Internet AS-level topology. Although evaluation studies have been presented for each topology generator, each of these studies assumes a different framework. Therefore, it is difficult to compare the differences between the topology generator algorithms. The objective of this paper is to fill this gap by comparing representative families of topology generators.

Keywords: Network Topology, Internet, Power Law, Graphs, Algorithms.

1. INTRODUCTION

The first widely used class of Internet topology generators was developed by Waxman [23]. The Waxman algorithm is a variant of the Erdös-Rényi random graph [15] based on the Euclidean distance. Later research claimed that the real Internet topology does not obey a random-structure, but instead possesses some kind of hierarchy, for instance a differentiation between transit and stub nodes. As a consequence, structural generators as Transit-Stub [25], Tiers [26], and GT-ITM[1] appeared. These structural generators were considered *valid* until the appearance of a seminal paper by Faloutsos *et al.* [22] in 1999. In that paper, the nodal degree of the Internet AS-level topology was shown to closely obey a power law. The graphs generated by the structural generators do not exhibit this power law behavior, turning them into deficient Internet topology models [18]. The work by Faloutsos *et al.* [22] fueled the development of a new family of generators, such as the Barabási-Albert (BA)[2] and Power Law Random Graph (PLRG)[1]. In this paper we confine our scope to this last generation of power law algorithms.

All the algorithms have an evocative [21] approach to the problem in common: the algorithms can reproduce a metric of interest, but do not capture the underlying causal mechanisms. A well-understood network metric (in most cases the nodal degree distribution) is usually chosen, then an algorithm that matches the metric is developed. This approach presents several problems. First, it is hard to choose the single metric. Second, a method that matches the chosen metric, often does not fit other metrics of interest.

Power Laws and scale-freeness are two important concepts. A random variable $X \geq 0$ is said to possess a power law if the probability density function (*pdf*) obeys

$$f_X(x) = cx^{-\beta} \ (x > 0, \beta > 1) \tag{1}$$

where c is a normalization constant, and β is the power law exponent. The corresponding power law density function is scale-free because it satisfies

$$f_X(ax) = g(a) f_X(x) \tag{2}$$

An increase by a factor a in the scale or units by which one measures x does not change the overall density $f_X(x)$, except for a multiplicative scaling factor. In the remainder of this paper, we will be using the notions of scaling and power law distribution interchangeably.

Earlier work on comparison of Internet topology generators can be found in [8], [10] and [11]. In each of these papers, the authors test existing generators against a new generator they propose themselves using custom sets of metrics. Here, we use the same framework for evaluating all the generators to avoid these somewhat subjective results and ensure a fair comparison.

This paper is organized as follows. In Section 2., we present the power law algorithms: PLRG, Havel-Hakimi (in short Havel), Takao, BA, BA rewired (BA-r) and Generalized Linear Preference (GLP). In Section 3., we introduce the metrics used to compare the output graphs: degree distribution, average hopcount, clustering coefficient, assortativity and spectrum of the adjacency matrix. Further in Section 4., we present the qualitative analysis of the different algorithms under study. Finally, we conclude in Section 5. For a more detailed version of this paper, the reader is referred to [13].

2. TOPOLOGY GENERATORS

We can classify the existent Internet topology generators into two families: the curve fitting family and the preferential attachment family.

[1] http://www3.cc.gatech.edu/projects/gtitm/

The **curve fitting family** generators use an explicit scale-free degree distribution $\mathcal{D} = \{d_1, d_2, ..., d_N\}$ (the *curve*). Given $\mathcal{D}$, the algorithms interconnects the set of $\mathcal{N}$ nodes such that the resultant graph $G(N, L)$ with N nodes and L links has degree distribution $\mathcal{D}$. Generators of this family are PLRG, Havel-Hakimi and Takao.

The **preferential attachment family** combines the ideas of network growth and preferential attachment. The generator starts with a small, fully connected graph and divides its growth into time-steps. At every time-step, one node is added to the network. Then, m links are added between the new node and randomly chosen nodes, where the probability to attach to a node is proportional to the degree of the latter. Generators of this family we consider are BA, BA-r and GLP.

Finally, we also consider the **hybrid** generator inet3 that uses both curve fitting and preferential attachment.

2.1. Degree Distribution Generation

The performance of the PLRG, Havel-Hakimi and Takao algorithms highly depends on the provided input degree sequence $\mathcal{D} = \{d_1, d_2, ..., d_N\}$. Each nodal degree d_j is a random variable with power law distribution (1). All the degrees are assumed to be independent, thus ignoring the basic law [16] of the degree $\sum d_j = 2L$ that eventually correlates all nodal degrees. Since the process followed to generate degree sequences is identical for PLRG, Havel and Takao, their probability density functions present the same behavior (while other parameters may differ).

2.2. PLRG

The PLRG algorithm [1] first assigns *stubs* to the N nodes in the graph obeying the calculated degree sequence $\mathcal{D}$. Then the algorithm randomly connects nodes by matching the free stubs. In case that $\sum_{i=1}^{N} d_n$ is an odd number, one stub remains unassigned and is deleted.

The produced graph may not be connected and may contain self-loops and duplicate links. In our simulations, we delete self-loops, merge duplicate links, and extract the giant component. The connected components theory [17] states that for values of β between $2 < \beta < \beta_0$ ($\beta_0 = 3.478$ as derived in [1]) the random graph has a.s. a giant component and the size of the second largest component is $O(\log N)$. For $1 < \beta < 2$ the second largest component is a.s. of size $O(1)$.

2.3. Havel-Hakimi

The behavior of the Havel-Hakimi algorithm is deterministic: given a degree sequence $\mathcal{D} = \{d_1, d_2, ..., d_N\}$, the resultant graph G is always the same. The original Havel algorithm was designed to check whether a degree sequence is graphical[2] using the Havel-Hakimi theorem [9]. The algorithm is based on the following theorem.

Theorem 2..1 *(Havel-Hakimi theorem) Let* $\mathcal{D} = \{d_1, d_2, ..., d_N\}$ *be a sequence of non-negative integers with* $d_1 \geq d_2 \geq ... \geq d_N$ *. Let* $\mathcal{D}'$ *be a sequence* $\{d'_1, d'_2, ..., d'_N\}$ *obtained from* $\mathcal{D}$ *by setting* $d'_1 = 0$, $d'_i = d_i - 1$ $(i = 2, .., d_1 + 1)$, *and* $d'_j = d_j (j = d_1 + 2, ..., n)$. *Then* $\mathcal{D}$ *is graphical if* $\mathcal{D}'$ *is graphical.*

In words, $\mathcal{D}$ is graphical if the following sequence is graphical: replace d_1 by 0 and subtract 1 from the next d_1 degrees:$\{d_2, d_3, ..., d_{d_1+1}\}$. This algorithm ensures that the high degree nodes will be connected with the high degree nodes.

2.4. Takao

The Takao algorithm was introduced by Takao Asano in [19]. Similar to PLRG and Havel, the Takao algorithm requires an explicit degree sequence as input. The Takao algorithm is deterministic, such that for degree sequence $\mathcal{D}$ the algorithm will always produce the same graph.

The main feature of the Takao and Havel algorithms is their speed, which is in the order of $O(\sum_{i=1}^{N} d_i/2)$ provided $\mathcal{D}$ is graphical. The Takao algorithm first checks whether the given degree sequence is graphical by applying Theorem 2..2. Similar to Havel, the connectivity information is obtained from recursive application of Theorem2..2, but the connectivity algorithms are completely different. The Takao theorem is defined as follows:

Theorem 2..2 *(Takao theorem) Given a degree sequence* $\mathcal{D} = \{d_1, d_2, ...d_N\}$ *of positive integers where* $n > d_1 \geq d_2 \geq ... \geq d_n > 0$. *Let* $h = d_n$, $x = min\{j | d_j = d_h\}, y = max\{j | j \leq n-1, d_j = d_h\}$. *Describe* $\mathcal{C} = \{c_1, c_2, ...d_{N-1}\}$ *as a sequence of positive integers where* $c_1 \geq c_2 \geq ... \geq c_{n-1}$, *and*

$$c_i = \begin{cases} d_i - 1, & \text{if } 1 \leq i \leq x-1 \text{ or } y-h+x \leq i \leq y \\ d_i, & \text{if } x \leq i \leq y \text{ or } y+1 \leq i \leq n-1 \end{cases}$$

Then $\mathcal{D}$ *is graphical if* $\mathcal{C}$ *is graphical.*

In words, at each step we are removing from $\mathcal{D}$ the node with the lowest degree h, and subtracting d_h units from the set of highest degrees.

[2] A degree sequence $\mathcal{D}$ is called graphical if it is possible to draw at least one graph with degree sequence $\mathcal{D}$.

2.5. Barabási-Albert

The two main concepts behind the BA algorithm are the linear preferential attachment and Yule's process. The combination of a growth process with preferential attachment has been proven theoretically [4] and empirically [2] to lead to scale invariant distributions. The algorithm starts with creating a small, fully connected core of m_0 nodes. Then, the BA algorithm incrementally constructs a topology by continuously adding nodes. Each new node is connected to m randomly chosen nodes. The probability that the new node will be connected to a node i is given by

$$\Pr[X = i] = \frac{d_i}{\sum_{\forall j} d_j} \tag{3}$$

The rewiring variant BA-r [3] separates the growing process into three main events: addition of one node, addition of m links, and rewiring of m links taken from the complete set of links in the network. Each time-step, one of these operations is chosen randomly with probabilities p, q and $1 - p - q$, respectively.

2.6. GLP

GLP is a variant of the BA model. Chen *et al.* pointed out in [20] that in the real Internet, new ASs have a stronger preference to connect to high degree ASs than predicted by the linear preferential model. To achieve a stronger preference for high degree nodes, the probability that a new node will be connected to the node i is adapted [8] to non-linear preferential attachment

$$\Pr[X = i] = \frac{d_i - \theta}{\sum_{\forall j}(d_j - \theta)} \tag{4}$$

The implementation of BA, BA-r and GLP algorithms have been retrieved from the BRITE project[3].

2.7. Inet3

Inet3 [10] is based on empirical data extracted from the Oregon Route-Views project[4]. Thus, the algorithm is based on the BGP Autonomous System (AS) topology.

Inet3 follows the next sequential steps to generate the final topology: first compute the number of months t that would take the 1997's Internet to reach N nodes (exponential growth of the number of nodes is assumed). Second, compute the new frequency (and rank) distributions using the calculated t. The degree distribution is calculated through the pdf $f_X(x) = e^{at+b}x^{-S}$ where a and b are known constants (extracted from Oregon Route-Views). For the 2% higher degree nodes, apply instead the ccdf formula $F_X(x) = e^{ct+d}x^{-R}$. Third, assign degree 1 to $m\%$ of the nodes. Fourth, form a spanning tree with nodes of degree higher than 1, creating G. Fifth, attach nodes with degree 1 to G using linear preferential attachment (3). Finally match the remaining nodes with G using linear preferential attachment.

Inet3 depends on the empirical constants a, b, c, d, S, R and m. The values for the constants have been extracted from November 1997's Internet snapshot[5].

[3] http://www.cs.bu.edu/brite/

[4] http://www.routeviews.org

3. TOPOLOGICAL CHARACTERISTICS

In this Section, we introduce the metrics used to analyze and compare the different topology generators.

3.1. Degree Distribution

Applying logarithms to both sides of (1) we obtain the equation of a line with slope β.

$$\log(y) = \log c - \beta \log x \tag{5}$$

The simplest way to empirically obtain β is by performing a linear regression of $f_X(x)$ when plotted on a log-log scale, but the frequency distribution plot is known to underestimate β [21]. Even worse, plotted pdfs have tendency to falsely suggest that a scaling behavior exists. Figures 1a-1d in [21] demonstrate how an exponential pdf can be interpreted mistakenly as *scale-free*. The use of the ccdf solves the classification problem, as it clearly discriminates between exponential and power law distributions. In conclusion, ccdf plots are more reliable than the frequency degree plots. From here on, we only use ccdf plots.

3.2. Hopcount

The hopcount or path length between two nodes is a characterizing property of a graph. It is defined as the minimal number of distinct links that forms a path connecting two given nodes. The mean hopcount is the mean of the shortest path lengths connecting each node $n \in \mathcal{N}$ to all other nodes in G.

Generally, a topology is assumed to be either completely regular or completely random. But many biological, technological and social networks lie somewhere between these two extremes. Systems can be highly clustered, like regular lattices, yet have small average hopcount [24], like random graphs. These are called *small-world* networks, by analogy with the small-world phenomenon [24]. The average hopcount distribution is an important tool to distinguish between *small world* and large diameter graphs, like lattices [16].

[5] Currently, the latest version of Inet generator is *Inet-3.0*.

3.3. Assortativity

The assortativity coefficient $R(G)$ was introduced by Newman in [14] as

$$R(G) = \frac{\sum_{(i,j)\in\mathcal{L}} d_i d_j - \left(\sum_{i\in\mathcal{N}} \frac{1}{2} d_i^2\right)^2 / L}{\sum_{i\in\mathcal{N}} \frac{1}{3} d_i^3 - \left(\sum_{i\in\mathcal{N}} \frac{1}{2} d_i^2\right)^2 / L} \qquad (6)$$

Assortative mixing ($R > 0$) is defined as "a preference for high-degree nodes to attach to other high-degree nodes" and disassortative mixing ($R < 0$) as the converse, where "high-degree nodes attach to low-degree ones". Assortative and disassortative mixing patterns indicate a generic tendency to connect to similar or dissimilar peers, respectively.

If we plot the *average degree of the neighbors of a given node* i versus the node degree d_i we obtain a graphical representation of the assortativity coefficient. The slope of the linear fit is directly related to $R(G)$ (see Section 4.4. for examples).

3.4. Clustering Coefficient

The clustering coefficient $c_G(i)$ characterizes the density of connections in the environment of a node i. A node's clustering coefficient is commonly defined as the ratio of the number y of links connecting the d_i neighbors of i, over the maximum possible $\frac{1}{2} d_i(d_i - 1)$.

$$c_G(i) = \frac{2y}{d_i(d_i - 1)} \text{ provided that } d_i \geq 2 \qquad (7)$$

The clustering coefficient $c_G(i)$ expresses local robustness in the graph and thus has practical implications: the higher the local clustering, the more interconnected are its neighbors. The clustering coefficient for the whole graph C_G is defined as the average of the clustering coefficient for all the nodes with degree higher than 1.

3.5. Spectrum of the Adjacency Matrix

The adjacency matrix A of a graph G with N nodes is an $N \times N$ matrix with elements $a_{ij} = 1$ only if (i, j) is a link of G, otherwise $a_{ij} = 0$. We assume bidirectional links, if there is a link from i to j ($a_{ij} = 1$) then there is a link from j to i ($a_{ji} = 1$) for any $j \neq i$. Moreover, we exclude self-loops ($a_{jj} = 0$) or multiple links between two nodes i and j. The spectrum of G is the set of eigenvalues of its adjacency matrix A.

The spectrum of the adjacency matrix is an important global characteristic of a topology. It yields tight bounds for a wide range of graph characteristics, such as distance-related parameters, expansion properties, and values related to separator problems estimating graph resilience under node/link removal. Faloutsos *et al.* [22] showed that the larger eigenvalues of the Internet AS-level correspondent adjacency matrix

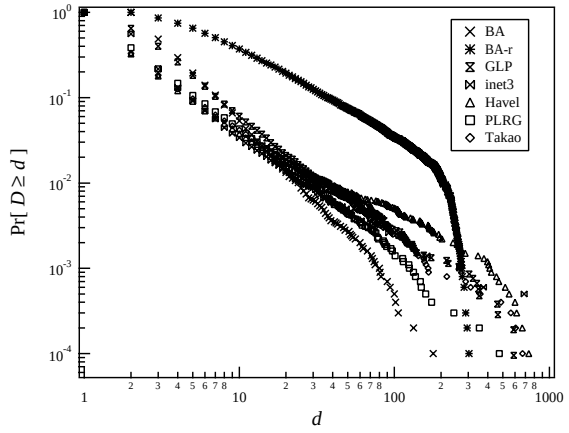


Figure 1. The ccdf curves for the 7 algorithms. Apart from BA-rewire, each curve closely follows a straight line, which expresses the Power Law property.

follow an empirical power law, such that

$$f_\lambda(x) \simeq x^{-\delta} \qquad (8)$$

Separately, Dorogovstev *et al.* [6] showed that the tail parameter β of the power law degree distribution, and the tail parameter δ for the highest eigenvalues are directly related through

$$f_\lambda(x) \simeq x^{1-2\beta} \qquad (9)$$

Eq. (8) and (9) together provide a relation between the tail exponent of the degree distribution, and that of the spectrum.

4. COMPARING GENERATORS BY SIMULATION

4.1. Methodology

All generated networks consist of $N = 3050$ nodes. The main reason for this number is that inet3 requires at least 3037 nodes, which is the number of nodes in 1997 Internet AS topology. To generate the degree sequences $\mathcal{D}$, we choose $\beta = 2.18$ as tail exponent. This value of β is in agreement with results for the Internet AS-level topology [8] and [12]. The tail exponent β appears to be constant over time [10].

4.2. Degree Distribution

Figure 1 plots the degree distribution ccdf for all the topology generators, on log-log scale. Figure 7 in the Appendix contains separate diagrams for the degree distribution ccdf of each algorithm. The linear slopes in the ccdf plot indicate that the obtained distributions indeed follow a power law.

For each data set, we calculate the linear regression to get β. Additionally, we calculate the Pearson's r linear correlation coefficient and the coefficient of variation ($C_v = \sigma^2/\mu$)

	ccdf β	r	C_V
PLRG	2.47	0.99	35.3
Havel	2.05	0.97	21.9
Takao	2.27	0.99	38.1
inet3	2.21	0.99	31.3
BA	2.96	0.99	31.0
BA-r	2.40	0.87	13.8
GLP	2.34	0.99	25.5

Table 1. Linear fit of β for the ccdf data set, Pearson's r and coefficient of variation C_V. The correlation coefficients lie above 0.97 in all the cases, except BA-r.

of the degree distribution. The first parameter estimates the goodness of the linear fits, where the second quantifies the grade of variability. The numerical results are summarized in Table 1.

Figure 1 illustrates how the algorithms correctly match a power law distribution. However, the tail exponents deviate considerably from the exponent of the real Internet AS topology, $\beta \approx 2.18$ [6]. The maximum degree differs for each generator. This can better be observed in Figure 7. While for BA the maximum degree is around 200, GLP has a higher maximum degree of 600. The reason for this is the non-linear preferential attachment of GLP as opposed to the linear preferential attachment of BA and BA-r. On the other hand, BA has more low-degree nodes.

BA-r[6] follows a power law distribution up to $d_i \approx 100$. However, the degree distribution becomes exponential for higher degrees, which explains the small coefficient of variation. The exponential tail can be ascribed to the rewiring process. By randomly rewiring links, high degree nodes are more likely to loose a link, such that high degree nodes are losing degrees. The PLRG algorithm generates topologies with isolated nodes, duplicate links and self-loops. After these irregularities have been solved (deleting isolated nodes, merging duplicate links and deleting self-loops) the remaining topology has an obvious deficiency in the degree distribution, as several low degree nodes are removed from the graph. Consequently, the tail exponent β is affected.

4.3. Hopcount

Figure 2 shows the histogram of the hopcount distribution for all the proposed generators. The average hopcount generally lies under 5 hops, thus the networks tend to exhibit the small world property.

In [17] and [5] it is shown that, for large N, the average

[6]The family of Barabási-Albert generators does not have nodes with degree 1, since during the generation process each new node starts with initial degree $m = m_0 = 2$.

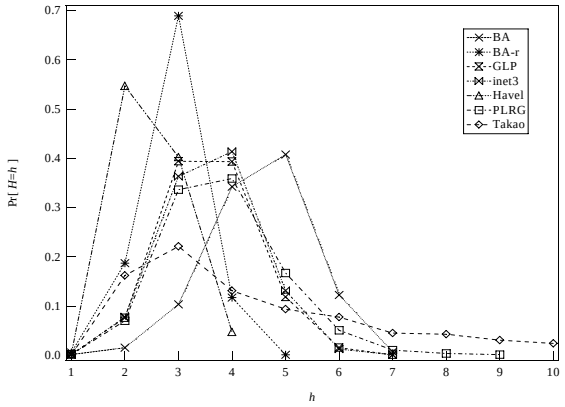


Figure 2. Hopcount histogram for all the considered algorithms. Takao represents a notable exception, as it interconnects long chains of nodes, making unappropriate use of the input degree sequence.

hopcount of power law graphs is approximated by,

$$E[H_N] = 2\frac{\log\log N}{|\log(\beta-2)|}(1+o(1)) \qquad (10)$$

yielding for $N = 3050$ and $\beta = 2.18$ that $E[H_N] \approx 2.42$.

Table 2 presents the numerical results of the simulations and compares them with (10). The results present remarkable differences even for algorithms belonging to the same family. For instance, BA with a mean value of 4.5 lies far from BA-r with a mean of 2.4. This difference is due to the random rewiring process. Watts and Strogatz demonstrated in [24] that randomly rewiring creates shortcuts in a network, hence decreasing the average hopcount. GLPs non-linear preferential attachment also decreases the average hopcount as compared to BA, because it creates densely connected cores through which all nodes can be reached within a few hops. The differences in the hopcount between the generators (of the same family) shows that minor modifications in the algorithm have a strong effect on the network properties.

Havel and BA-rewire show a very narrow distribution that centers around only two hops. Havel's algorithm systematically interconnects high-degree nodes with high-degree nodes, creating a highly interconnected core, through which almost all nodes can be reached within 3 hops.

Takao interconnects long chains of nodes with low degree, which is an artifact of the algorithm. The direct consequence is a dramatic increase of the average and maximum hopcount.

4.4. Assortativity

Figure 3 plots the average degree of the neighbors of a node with degree d for all the algorithms. Figure 7 in the Appendix contains separate diagrams of Figure 3 for each algorithm. Table 2 presents the numerical results. As observed in Table 2,

	theory	μ	σ^2	max	R(G)
PLRG	5.51	3.8	0.83	9	-0.13
Havel	1.39	2.4	0.54	4	-0.38
Takao	3.18	7.3	5.19	243	-0.23
inet3	2.66	3.6	0.72	7	-0.18
BA	102	4.5	0.77	7	-0.01
BA-r	4.54	2.9	0.36	5	-0.01
GLP	3.86	3.6	0.72	7	-0.11

Table 2. The four left rows show the theoretical mean, empirical mean, variance, and maximum value of the hopcount for the studied generators. Takao's maximum hopcount clearly exceeds the others. The theoretical value for BA is not correctly approximated, as Eq. (7) is only valid in the range $2 < \beta < 3$. The last column shows the average assortativity coefficient $R(G)$ of the considered algorithms.

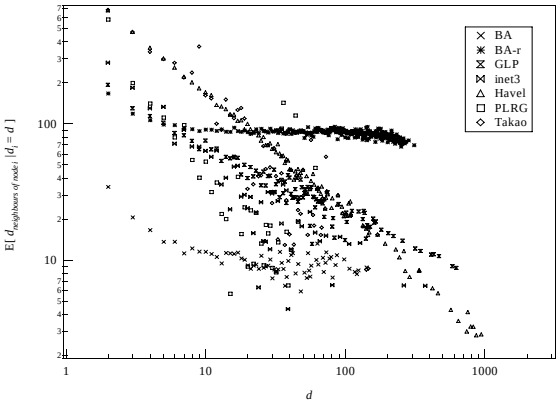


Figure 3. Average degree of the neighbors (y-axis) given a degree d (x-axis).

the assortativity of inet3, PLRG, Havel, and Takao is clearly negative, which corresponds to disassortative behavior. This effect is represented as a negative slope in Figure 3, meaning that low degree nodes tend to connect to high degree nodes, and conversely, high degree nodes tend to interconnect with lower degree nodes.

The assortativity coefficient for the Internet AS level graph is found [14] as $R(Internet) \approx -0.189$, a significant disassortative behavior. In comparison to the Internet, the deterministic generators considered are strongly disassortative, while the preferential attachment family topologies are weakly disassortative. As an extension of BA, GLP improves the assortativity.

In accordance with other studies [8], the degree of the neighbors for BA and BA-r is close to constant, which means that there is no correlation between the degree of a node and that of its neighbors. As an extension of BA, GLP improves the assortativity. Because in GLP, the tendency of the nodes to attach to higher degree nodes is stronger (non-linear preferential attachment), a few highly connected nodes will arise which are then connected to many low degree nodes.

Intuitively we might infer that the assortativity of Havel is positive, as the algorithm uses the lemma "*High degree connects to high degree*". But the empirical results ($R(G)_{HAVEL} = -0.38$) contradict this interpretation. The essential reason for this apparent conflict is that $R(G)$ is normalized against a set of graphs containing self-loops and isolated nodes. So Havel could be thought of as disassortative when compared with *all* graphs [21].

Empirical data shows that 20% of the PLRG nodes do not belong to the Giant Component, and thus they are removed. As these erased nodes represent a significant fraction of nodes with low degree, subtracting them from the graph implies that the higher degree nodes are *losing* low degree connectivity. This produces an increase on the *average degree of neighbors* for the high degree nodes**.**

4.5. Clustering Coefficient

Figure 4 shows the clustering coefficient for the evaluated generators, plotting a vertical compound of all the histograms. The real Internet AS-level graph clustering coefficient is $C_{Internet} \approx 0.46$ [8]. Figure 4 shows that none of the proposed generators closely matches the Internet's clustering coefficient. Table 3 shows a detailed summary of the obtained results.

BA, BA-r, GLP and inet3 present low variance in comparison to the rest of the algorithms, where Havel and Takao posses higher variance and irregular distributions (right side of Figure 4).

The clustering coefficient of Barabási-Albert is very low ($\mu_{BA} = 0.012$), which implies that BA is organized as a tree or a star. The rewiring process in BA-r interconnects random nodes, which affects the tree structure and thus increases the clustering coefficient. The rewiring probability $1 - p - q$ has a direct effect on the clustering coefficient, as pointed out earlier in [24]. GLPs non-linear preferential attachment increases the tendency of new nodes to attach to higher degree nodes. Hence when a new node is added to the network, it is more likely, in comparison to BA, that the nodes to which it attaches have a high degree. Since, these two nodes have a high degree, the probability that they are connected is also high, such that a *triangle* is created between the three nodes, which increases the clustering coefficient. The clustering coefficient, as defined in Section 3.4., is also equal to the ratio between the number of triangles that contain node i and the number of triangles that could possibly exist if all neighbors of i were interconnected. A higher number of triangles results in a higher overall clustering coefficient. Hence, the clustering coefficient of GLP is expected to be higher than BA. The clustering coefficient for inet3 ($\mu_{inet3} = 0.1255$)

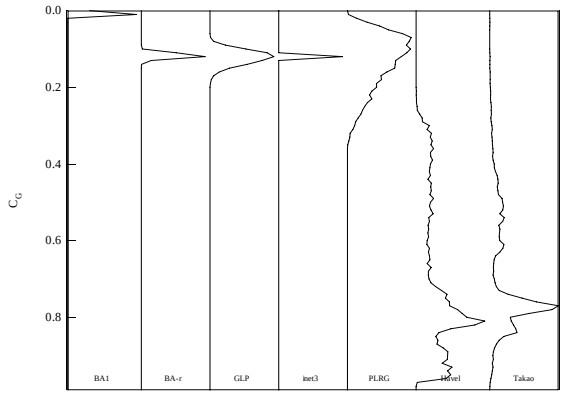


Figure 4. Vertical compilation of the histograms of clustering coefficients. From left to right: BA, BA-r, GLP, inet3, PLRG, Havel and Takao. While some algorithms present widely spread histograms (Havel and Takao), others present zero variance (BA and inet3).

appears constant with no variance.

Both Havel and Takao generate graphs with high C_G. The recursive nature of these algorithms systematically interconnects the neighbors of a node. Yet, the average hopcount of Havel and Takao strongly differs. The difference arises because the Takao algorithm interconnects *long* tails of nodes with low degree. These tails dramatically increase the average hopcount, but still preserve a central clustered core.

As the average hopcount for Takao is very large, this algorithm does not exhibit the small world property. This effect indicates that a power law degree sequence is not a warranty to obtain a small world graph [21].

The BA algorithm presents a very small clustering coefficient, implying that the network also does not exhibit the small world property [24]. PLRG, BA-r, GLP and Havel do have the small world property, as they combine a low average hopcount with high clustering coefficients when compared [16] to the random graph $C_{Gp(N)} \approx 10^{-3}$.

4.6. Spectrum

As pointed by Faloutsos *et al.* [22], an eigen exponent δ exists, such that the spectral density decays as a power law (8) for *large* eigenvalues. However, the linear regression is non trivial, as it depends strongly on what we consider a *large* eigenvalue. This problem is illustrated in Figure 6.

There is a rich literature on the eigenvalues of graphs and their relation with topological properties. The eigenvalues of random graphs tend to the *semicircle law* [6] [16], whereas for scale-free networks the spectrum resembles a bell shape representing a tree-like graph [7]. Moreover, the shape of the spectrum at lower eigenvalues is directly related to the randomness used by the algorithms to generate the topologies:

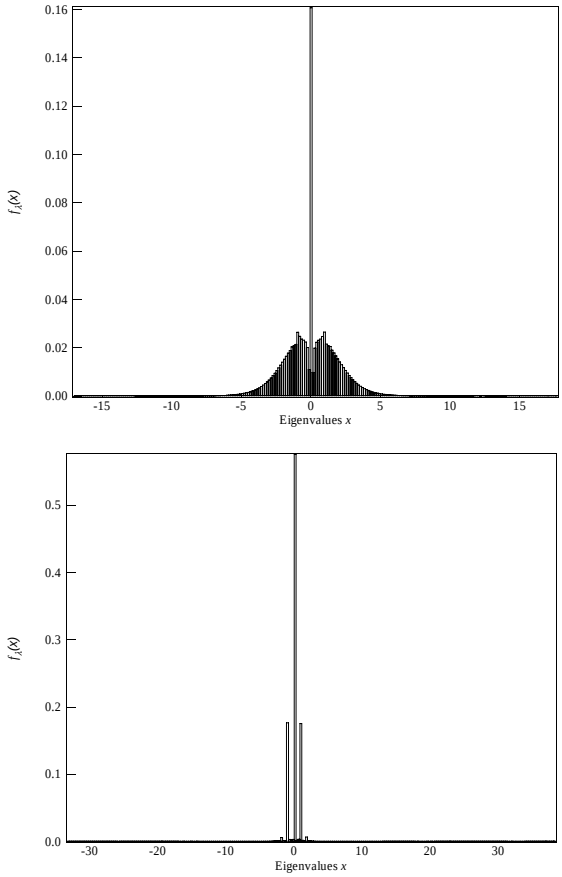


Figure 5. Top: The BA spectrum. The bell shape of the lower eigenvalues reveals a stochastic topology generator. Bottom: The Havel spectrum. The spectrum of the deterministic algorithms presents very sharp shapes; approximately 87% of the spectral density is concentrated in 3 eigenvalues.

stochastic topology generating algorithms create graphs that present bulk-shape forms for small eigenvalues [7]. This behavior can be observed for BA (top of Figure 5), BA-rewire, GLP and PLRG. On the other hand, the deterministic algorithms Havel (bottom of Figure 5) and Takao concentrate their densities in the zero-eigenvalue λ_0. The Inet3 spectrum is qualitatively between the deterministic and random behavior.

Real Internet AS-level graphs present power law features in their spectrum, as indicated by Eq. (8). The generators under study correctly emulate this scaling behavior: the spectrum decays as a power law for large eigenvalues. Figure 6 illustrates an example of this fitting process with GLP.

Table 3 shows the relation between the tail exponent of the spectrum δ and that of the degree distribution β. The difference is expressed as $|\Delta| = \delta - 2\beta + 1$.

Although BA-r, inet3 and Havel follow the scaling of Eq. (8), they do not behave like the Internet ($|\Delta|_{Internet} \approx 0$),

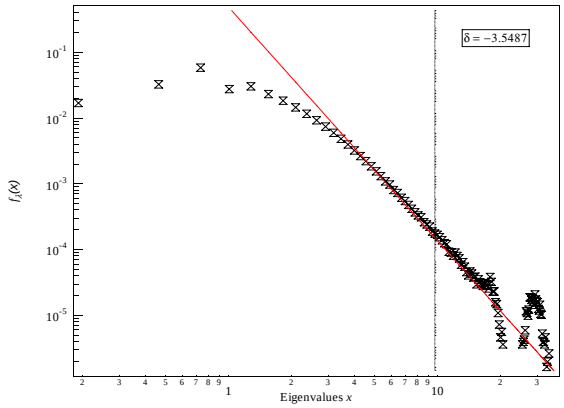


Figure 6. Righthand side of the spectrum of GLP on log-log scale. The line indicates the linear fit. The range of the data to fit is somewhat subjective, since there exists no clear definition on what are considered 'large' eigenvalues. Here we only consider eigenvalues greater than 10.

	μ	σ^2	β	δ	$\lvert\Delta\rvert$
PLRG	0.139	0.070	2.47	3.95	0.01
Havel	0.698	0.190	2.05	1.88	1.22
Takao	0.687	0.150	2.27	3.55	0.01
inet3	0.126	0	2.20	2.39	1.01
BA	0.012	0.002	2.95	4.78	0.12
BA-r	0.123	0.005	2.40	1.69	2.11
GLP	0.127	0.018	2.34	3.54	0.14

Table 3. The first two columns show the mean and standard deviation for the clustering coefficient. The last three columns show a comparison between the tail exponents of the spectrum (δ) and that of the degree distribution (β). Dorogovstev *et al.* [6] found that the empirical relation $\delta = 2\beta - 1$ holds for the real Internet AS graph.

because the measured difference does not match Eq. (9).

5. CONCLUSIONS

The metrics analyzed here are a grasp of the full range of characteristics that can be computed, but they seem sufficient to differentiate between all the algorithms. On the other hand, one single metric is still not enough to classify an algorithm. Table 4 summarizes the simulation results.

The curve fitting family can be split in two groups. The first group contains Havel and Takao. This group shows poor results: a highly variable clustering and large hopcounts. This eliminates them as Internet AS topology generators. From this results, we can conclude that the degree distribution does not provide enough insight into the Internet AS topology. The second group, PLRG, behaves close to the Internet AS in all the proposed metrics.

	Degr.	**Hopc.**	**Ass.**	**Clus.**	**Spect.**
Internet	**2.18**	**2.42**	**-0.19**	**0.46**	**0**
PLRG	2.47	3.8	-0.13	0.14	0.01
Havel	2.05	2.4	-0.38	0.70	1.22
Takao	2.27	7.3	-0.23	0.69	0.01
BA	2.96	4.5	-0.01	0.01	0.12
BA-r	2.40	2.9	-0.01	0.12	2.11
GLP	2.34	3.6	-0.11	0.13	0.14
inet3	2.21	3.6	-0.18	0.13	1.01

Table 4. Summary of the simulation results. The columns compare the algorithms degree distribution tail exponent, average hopcount, assortativity coefficient, clustering coefficient and the spectrum deviation Δ versus the real Internet AS topology.

The preferential attachment family misses the disassortative behavior of the Internet AS topology. GLP equals and improves BA in all the results, especially inserting disassortativity through the non-linear preferential attachment.

Inet3 starts as a good algorithm showing good results in degree distribution, hopcount and assortativity. But the static clustering coefficient and the spectrum results classify this algorithm as non appropriate.

We may summarize as follows:

- **PLRG** does not deviate considerably from the (estimated) real Internet AS results in any of its results, but the number of nodes in the output graph is not static.
- **Havel** design purpose is to check if a degree sequence is graphical. Thus Havel topologies are far from reality.
- **Takao** was designed to interconnect any graphical degree sequence in a short time. In the connectivity process unrealistic long chains of nodes are created.
- **BA**, the lack of input parameters makes it useless to generate real-Internet topologies. It still remains as an interesting toy model.
- **BA-r**, the simulations provide a defect in the tail of the degree distribution as it decays exponentially. This invalidates the model.
- **GLP** improves Barabási-Albert in many parameters. However, the algorithm needs many inputs, thus it depends highly on real Internet AS-level measurements.
- **inet3**, trying to mimic the power law degree distribution (using constant parameters), provides undesirable side effects such as the static clustering coefficient.

As GLP matches almost all the Internet AS parameters equally or better than PLRG and inet3, we conclude that GLP is the most suitable Internet AS topology generator.

REFERENCES

[1] W. Aiello, F. Chung, L. Lu. *A Random Graph Model for Massive Graphs*. 2000.

[2] A.-L. Barabási, R. Albert. *Emergence of Scaling in Random Networks*. Science vol. 286, October 1999.

[3] A.-L. Barabási, R. Albert. *Topology of Evolving Networks: Local Events and Universality*. Physical Review vol. 85 n24.

[4] B.Bollobas, O. Riordan, J. Spencer, and G. Tusanady, *The degree sequence of a scale-free random graph process*, Random Structures and Algorithms, 18, pp 279-290, 2001.

[5] F.Chung, L.Lu, *The average distance in a random graph with a given expected degrees*, PNAS 99 (2002) 15879-15882.

[6] S.N. Dorogovstev, A.V. Golstev, J.F.F.Mendes, A.N.Samuknin. *Spectra of complex networks*. Physical Review E 68 046109, 2003.

[7] I.J. Farkas, I. Derenyi, A.-L. Barabási, T. Vicsek. *Spectra of "real-world" graphs: Beyond semicircle law.* Physical Review E 64 026704, 2001.

[8] T. Bu and D. Towsley, *On Distinguishing between Internet Power Law Topology Generators*, in Proc. Infocom 2002, June 23–27, New York.

[9] S.L. Hakimi. *On the realizability of a set of integers as degrees of the vertices of a graph*. SIAM J. Appl. Math, 10, 1962.

[10] J. Winick, S. Jamin. *Inet3.0: Internet Topology Generator*. University of Michigan Technical Report CSE-TR-456-02.

[11] D. Magoni, J-J. Pansiot, *Analysis and Comparison of Internet Topology Generators*. Networking 2002, LNCS 2345, pp. 364-375, 2002.

[12] B. Zhang, R. Liu, D. Massey, L. Zhang. *Collecting the Internet AS-level Topology*. http://irl.cs.ucla.edu/topology/.

[13] J. Martin Hernandez, T. Kleiberg, H. Wang , P. Van Mieghem, A Qualitative Comparison of Power Law Generators, Delft University of Technology, report20061115, 2006.

[14] M.E.J. Newman, *Assortative Mixing in Networks*, Physical Review Lett. 89, 208701 (2002)

[15] P. Van Mieghem, *Paths in the simple Random Graph and the Waxman Graph*, Probability in the Engineering and Informational Sciences (PEIS), vol. 15, pp. 535-555, 2001.

[16] P. Van Mieghem, *Performance Analysis of Communications*, Cambridge University Press, 2006.

[17] G. Hooghiemstra, R. van der Hofstad and P. Van Mieghem, *Random graphs with finite variance degrees*, Random Structures Algorithms, 26, 76-123, 2005.

[18] H. Tagmunarunkit, R. Govindan, S. Jamin, *Network topology Generators: Degree-based vs. Structural*, SIGMCOMM'02, August 19-23, 2002.

[19] Takao Asano. An *O(n log log n) Time algorithm for Constructing a Graph of Maximun Connectivity with Prescribed degrees*. Journal of Computer and System Sciences, 51, 503-510 (1995).

[20] Q. Chen, H. Chang, R. Govindan, S. Jamin, S. J. Shenker, and W. Willinger, *The origin of power laws in Internet topologies revisited*, in IEEE INFOCOM, 2002.

[21] L. Li, D. Anderson, J. C. Doyle, W. Willinger. *Towards a Theory of Scale-Free Graphs: Definition, Properties and Implications*, 2005.

[22] M. Faloutsos, P. Faloutsos, C. Faloutsos. *On Power-Law relationships of the Internet Topology*, 1999.

[23] B.M. Waxman. *Routing of multipoints connections. IEEE Journal of selected areas in communication*, 6(9):1617-1622, December 1988.

[24] D. Watts, S. Strogatz. *Collective Dynamics of 'small world' networks*. NAture, 393, 440-442, 1998.

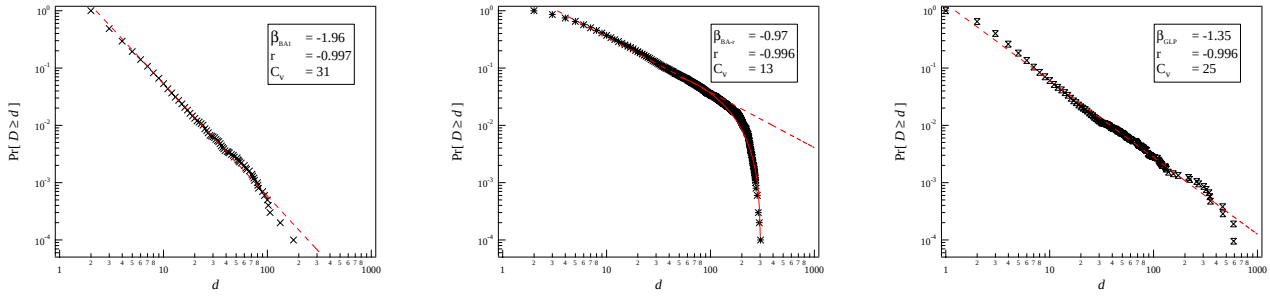
[25] K. Calvert, M. Doar, and E. Zegura. *Modelling Internet Topology*. IEEE Communications Magazine. June 1997.

[26] M. Doar. *A better model for generating test networks*. In *Proeeding of IEEE Global Telecimmunications Conference* (GLOBECOM), November 1996.

A ACKNOWLEDGEMENTS

This work is funded by the STW foundation Netherlands as part of the Network Dynamics and QoS project (DTC.6421) and by the Netherlands Organization for Scientific Research (NWO) under project number 643.000.503.

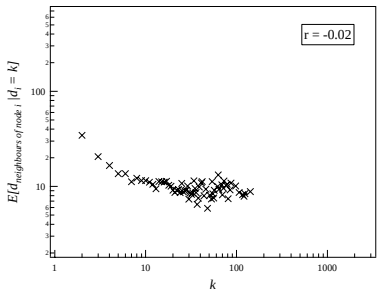
B FIGURES



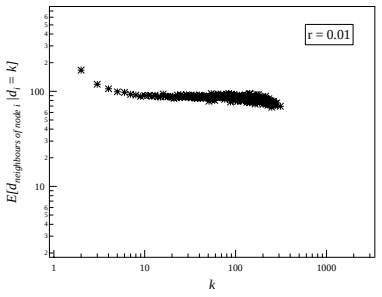
(a) CCDF BA

(b) CCDF BA-r

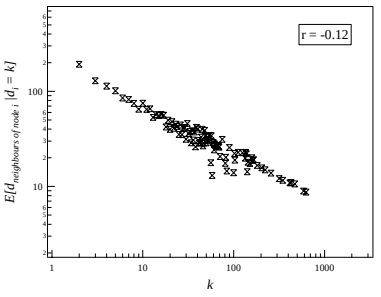
(c) CCDF GLP



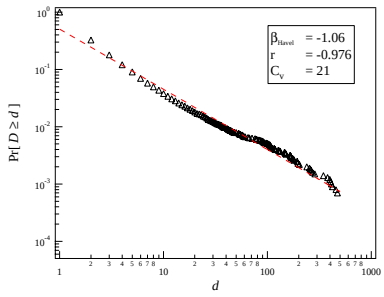
(d) Avg. Degree neighbors BA



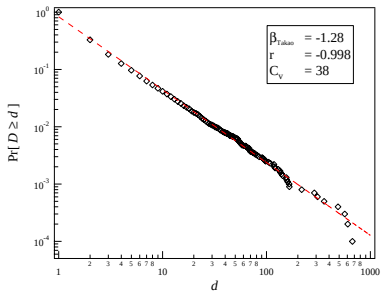
(e) Avg. Degree neighbors BA-r



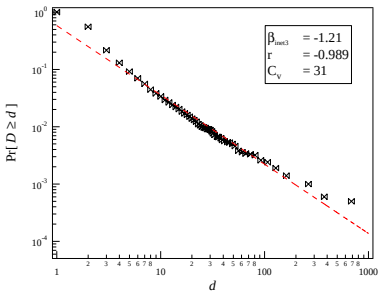
(f) Avg. Degree neighbors GLP



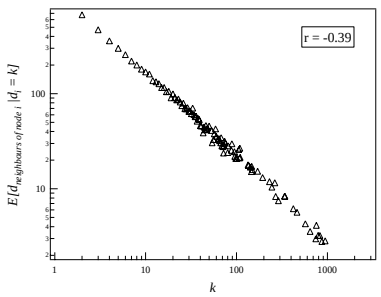
(g) CCDF Havel-Hakimi



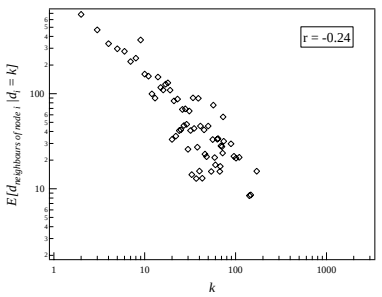
(h) CCDF Takao



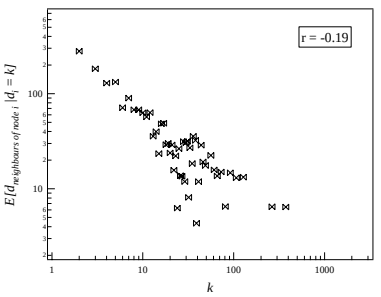
(i) CCDF inet3



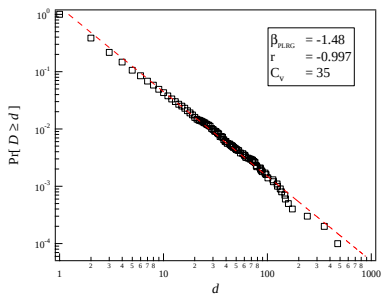
(j) Avg. Degree neighbors Havel-Hakimi



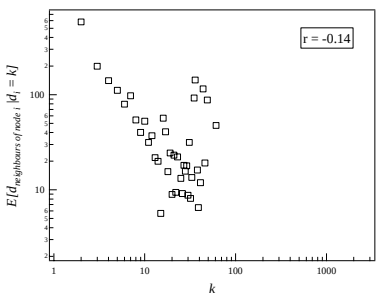
(k) Avg. Degree neighbors Takao



(l) Avg. Degree neighbors inet3



(m) CCDF PLRG



(n) Avg. Degree neighbors PLRG

Figure 7. The separate CCDF diagrams for the seven algorithms together with the diagrams showing the correlation between the average degree of a node's neighbors and its own degree.