Deciding on the type of the degree distribution of a graph (network) from traceroute-like measurements

Xiaomin Wang

October 23, 2011

Abstract

The degree distribution of the Internet topology is considered as one of its main properties. However, it is only known through a measurement procedure which gives a biased estimate. This measurement may in first approximation be modeled by a BFS (Breadth-First Search) tree. We explore here our ability to infer the type (Poisson or power-law) of the degree distribution from such a limited knowledge. We design procedures which estimate the degree distribution of a graph from a BFS or multi-BFS trees, and show experimentally (on models and real-world data) that our approaches succeed in making the difference between Poisson and power-law degree distribution and in some cases can also estimate the number of links. In addition, we establish a method, which is a diminishing urn, to analyze the procedure of the queue. We analyze the profile of the BFS tree from a random graph with a given degree distribution. The expected number of nodes and the the expected number of invisible links at each level of BFS tree are two main results that we obtain. Using these informations, we propose two new methodologies to decide on the type of the underlying graph.

Key words: Topology of Internet, BFS

Contents

Chapter 1

Introduction

Contents

1.1 Complex networks

Complex networks from the real world, modeled as graphs, appear in many contexts, such as metabolic networks, protein interactions or topology of the brain in biology [Alb05, JTA+00, SBH02], relationships or exchanges of information in society [HD03, LENA+01, NWS02, WF94], references or co-authoring in citation analysis and occurrence in linguistic networks [Bir08, PNFB05].

In computer science, we can also cite many examples, like web graphs (hyperlinks between pages, see for instance [15], [BV04a, BV04b, Bro00, KKR+99]), or data exchanges (in peer-to-peer systems, e-mail, etc, see for instance [HKLFM04, LBGL05, LBLG04, VKMVS04]). The Internet may also be seen as a graph at several levels [DF07]: Autonomous Systems (AS), routers and links between them, or Internet Protocol (IP) hops between interfaces for instance. For a decade, these graphs have been at the core of an intense research activity [DF07, FFF99, GMZ03, KCC+07, MP01a, RsA04, WAD09] aimed at a better understanding and management of the Internet, which plays a crucial role in our society.

Most graphs from the examples cited above have some nontrivial statistical properties. The term *nontrivial* indicates that the properties of real-world graphs are significantly different from those of model graphs, such as lattices, rings and random graphs[1]. The main such properties [DGM08] are:

- Small density[2]: A *dense* graph is a graph in which the number of links is close to the maximal number of links. The opposite, a graph with far fewer links, is a *sparse* graph. The distinction between sparse and dense graphs is rather vague, and depends on the context, but most graphs met in practice are clearly sparse, with a number of links of the order of the number of nodes.

- Giant component: A giant component is a connected subgraph that contains the majority of all nodes [MR98]. For random graphs, it is not guaranteed that there always exists a giant component [Bol84, ER60], but for real-world graphs most nodes tend to be connected.

- Small diameter[3] and small average distance between nodes: This property sometimes leads to call real-world graphs as *small-world graphs*. In other words, between any two nodes, there is a short path (which is not true for lattices) [AJB99, BR04, CL04, NSW01].

[1]The concept *random graphs* varies according to the context. In this introduction, it refers to the graphs from the ER model [Bol01], *i.e.* graphs chosen at random among the ones with a given number of nodes and links

[2]For undirected simple graphs, the graph density is defined as: $D = \frac{2|E|}{|V|(|V|-1)}$, where $|E|$ is the number of links and $|V|$ is the number of nodes.

[3]The diameter of a graph is the length of the longest shortest path between any two nodes of this graph.

- Heterogeneous degree distribution: The degree distribution of a graph is the fraction P_k of nodes with degree k, for all k. A degree distribution is *homogeneous*, if all the values are close to the average, like in Poisson and Gaussian distributions; a degree distribution is *heterogeneous*, if there is a huge variability between degrees, with several orders of magnitude between them. When a distribution is heterogeneous, it makes sense to try to measure this heterogeneity rather than the average value, which then has little meaning. In some cases, this can be done by fitting the distribution by a power-law, *i.e.* a distribution of the form $p_k \propto k^{-\alpha}$. The degree distribution of real-world graphs often is well fitted by a power-law distribution with an exponent between 2 and 3 [CSN09].

- Strong clustering coefficient[4]: Several variants of this notion coexist [WS98, Hua06], their common goal being to capture the fact that nodes in a graph tend to form groups of strongly connected subgraph. In real-world networks, this is much more present than in random graphs.

Despite this knowledge of some common properties of most real-world graphs, understanding their structures and their evolutions is still challenge. Researchers have studied these key questions in the last years and they focus therefore on four main research areas: measurement, analysis, modeling and algorithmics of these networks [Lat07].

Measurement. In order to study complex networks like the Internet, the Web, social networks or biological networks, first of all one has to explore them. However, most real-world graphs are not directly available: data (information about nodes and links) are collected by some measurement procedures. How to perform the measurement is itself a challenge which depends on the various cases. In addition, obtained views are in general *partial* because of the enormous size of the considered cases. Such partial views reveal some properties in a *biased* form. Therefore, it is necessary to study the bias and try to correct it, either by improving the procedure, or correcting the results.

Analysis. Given a real-world graph, the first step is to describe its structure (static view) and evolution (dynamic view). This is done using statistical notions and/or structural ones, aiming at capturing the key features of the graph. This topic has led to an important stream of studies [AJB00, BA99, BGLL08, CSWN00, CEAH00, WF94]. The definition of such notions is however not trivial, as well as the evaluation of their relevance and the interpretation of obtained descriptions. The main method currently is to compare the real-world complex networks to random graphs in similar classes.

[4]Two versions of this measure exist: the global and the local. The global version was designed to give an overall indication of the clustering in the network, whereas the local gives an indication of the embeddedness of single nodes.

Modeling. In order to explain the nature of observations, to develop mathematically rigorous results and to conduct appropriate simulations, it is important to capture observed properties in models of real-world complex networks. This is generally done by the random sampling of graphs in a specific class of graphs generated by some explicit construction [Bol01, MR95, BAJ99, GLM06, NWS02]. We then obtain artificial graphs similar to the real ones regarding the selected properties.

Algorithmics. Finally, the study of very large graphs naturally calls for algorithmics for two reasons. First, the context of real networks raises original algorithms questions (such as community detection), which did not exist before. In addition, solutions to common classical algorithmic problems (such as the calculation of the diameter) are no longer applicable because of the size of the considered graphs. On the other side, the properties encountered in practice may be used to improve the efficiency of algorithms on real networks.

1.2 The case of the Internet

In the case of the Internet, the most frequently used measurement tool is *traceroute* [7], available on most operating systems[5]. It is a computer network diagnostic tool for displaying the route (path) followed by packets and measuring transit delays across an Internet Protocol (IP) network[6]. Each route consists of a monitor where traceroute is launched, a destination that is an arbitrary IP address that we indicate as a parameter of traceroute and a set of intermediate nodes. An image of the topology of the Internet is then obtained by merging a great number of such routes. If we increase the number of monitors, some properties will be better estimated [DAHB+06, GLM06, GL05].

A property of high interest in the map of the Internet is its *degree distribution*, *i.e.* the fraction P_k of nodes with k links, for all k: it may have a strong influence on the robustness of the network [AJB00, MLG09, CEAH00, CSWN00, KW08, AB02], on protocol design [MP01b], and on spreading of information [BBCS05, RsA04]. Moreover, it is often claimed that these degree distributions may deviate significantly from what classical models assume [AJB00, FFF99, RsA04, WAD09], which leads to an intense activity on modeling issues [FKP02, ALWD05].

However, the degree distribution of the Internet topology (at any of the levels cited above) is not readily available: one only has access to *samples* of these graphs, obtained through measurement procedures which are intricate, time and resource

[5]On Microsoft Windows operating systems it is named *tracert* [14]. Windows NT-based operating systems also provide *PathPing* [5], with similar functionality. Variants with similar functionality are also available, such as *tracepath* on Linux installations [12, 4]. For Internet Protocol Version 6 (IPv6) the tool sometimes has the name *traceroute6* [13].

[6]If we focus on AS network, we just need and accurate IP-to-AS mapping. Some researchers work on this topic in order to develop a scalable and accurate AS-level traceroute tool [MRWK03].

consuming, and far from complete. Even more important, these samples are *biased* by the measurement procedure, which may have a strong influence on the observed degree distribution [DAHB+06, AKCM05, LBCX03, WAD09, CM04, PR04].

As a consequence, the current situation regarding the degree distribution of the Internet is unclear [WAD09, LAWD04, KW08, LM08]. The relevance of obtained samples regarding the degree distribution observed from them is far from being established. In particular, there is a controversy on whether the Internet topology may have a homogeneous (typically Poisson), or heterogeneous (typically power-law) degree distribution [KW08, WAD09]. In order to obtain an answer to this question, the most widely used approach currently is to conduct larger and larger measurements, in the expectation that these will lead to accurate observations [LM08, SS05, CHK+09]. However, this may be a dead end: the degree distribution may be intrinsically biased by the measurement process [AKCM05, LBCX03] and in practice it may depend much on the sample size [LM08].

1.3 This thesis

We explore in this thesis a completely different approach: we consider a simple model of the Internet topology measurements and try to derive the type of the degree distribution of the underlying graph from this limited observation. Our basic goal therefore is to answer the following question: given the limited information obtained from measurement, does the underlying topology more likely have a heterogeneous (typically power-law) or a homogeneous (typically Poisson) degree distribution?

In many cases (traceroute measurements, BGP tables, and AS-level traceroute, typically), the measurement process may be approximated by a BFS (Breadth First Search) tree[7] from a given node of the network. Indeed, the Internet measurements mostly consist in sets of *routes* (i.e. paths in the considered topology) going from a monitor to a set of targets, collected from as many monitors as possible. Since each route is modeled as a shortest path and since one may expect routes to have long common prefixes, the view from each monitor may be approximated by a BFS tree. Although this is a rough approximation, in the lack of a widely accepted and better solution, it has been used in many occasions [LBCX03, OML08, AKCM05, VBD+07, DAHB+06]. We will use it in this thesis too.

Finally, we focus in this thesis on deciding on the type of the degree distribution (homogeneous or heterogeneous) of graphs from traceroute-

[7] In graph theory, breadth-first search (BFS) is a graph search algorithm [CLRS09] that begins at the root node and explores all the neighboring nodes. Then for each of those nearest nodes, it explores their unexplored neighbor nodes, and so on, until all nodes are covered.

like measurements (modeled by BFS), using both experimental and analytic methods.

1.4 Contributions

The following chapters of this thesis are organized as follows.

Chapter 2 describes some basic notations and concepts that we will use in this thesis.

Chapter 3 describes the basic rebuilding procedure which reconstructs a graph similar to the original one from one of its BFS trees. Comparing the distribution of the rebuilt graph and the distribution in theoretical view, we may decide on the type, either Poisson or power-law, in condition that the number of links is known. According to different strategies of selection, either random or preferential, two rebuilding strategies are proposed: RR and PP. The procedure is composed with re-adding the same number of links that are invisible in BFS tree. In order to avoid changing the diameter or the average distance a lot, we add links only among the allowed positions. The validity of our strategies is verified with random model graphs. Our strategy succeeds in deciding the type of the underlying graphs.

Chapter 4 develops the procedure of Chapter 3, while we no longer require the condition the number of links. We choose the number of links m from a wide range, then for each m and for each type we test the KS distance between the theoretical distribution and the distribution of the corresponding rebuilt graph. In those triple of $(type, m, KS)$, the minimum KS is chosen as the result of estimate. For deciding type, our strategy always runs well. But for estimating the number of links, a bias always presents for random model graphs.

Chapter 5 adapts our method with several monitors of BFS trees. While with several complete BFS trees, we first merge these BFS trees into one graph and the allowed positions are the intersection of the allowed positions of each BFS tree. Then RR or PP strategies are applied with the merged graph. First, we test the probability of link-detection on random model graphs. And we observe that about 10 roots are sufficient to cover a large part of links, so we choose the number of monitor no more than 20 in practice. Using several BFS trees, we observe a more exact result than using a single BFS tree. The estimate of the number of links is very close to the underlying one.

Chapter 6 contributes to some formal analysis on the profiles of BFS tree by using generating functions and configuration model. The procedure of BFS use a FIFO queue as the underlying data structure, which can be modeled as a diminishing urn problem. With the help of generating function and some technique of PDE, we may have an explicit expression of the solution. To get a more intuitive result, we compute two expectations: (1) the expected number of nodes at each level in a BFS tree; (2) the expected number of invisible links at each level in a BFS tree.

Chapter 7 presents two applications by using the analytic results of Chapter 6. Using the expected number of invisible links, we refine RR and PP strategies and we propose two improved versions: RRIL and PPIL strategies. Using the expected number of nodes at each level, the estimation is conducted by comparing the detected "node vector" and the "theoretical node vectors". By comparing node vectors, we need only a bounded BFS tree rather than a complete BFS tree. But the results on real-world graphs show that these two applications are not stable ones.

Chapter 8 gives the comparison of different methods and the suggestion of the future works.

Chapter 2

Preliminaries

Contents

2.1 Graphs and degree distributions

In this section, we present some basic concepts and notations for graphs and degree distributions, which are used in all this thesis.

2.1.1 Basic concepts and notations

A *graph* is an ordered pair $G = (V, E)$ comprising a set V of *vertices* or *nodes* together with a set E of *edges*, *lines* or *links*; in this thesis we use the notations: *nodes* and *links*. Links are 2-element subsets of V (i.e. a link is composed of two nodes.), and the relation is represented as an unordered pair. This type of graphs may be named precisely as *undirected* and *simple*. More generally, E may be a multi-set of unordered pairs of (not necessarily distinct) nodes; G is then called a *multigraph* or *pseudograph*.

The nodes belonging to a link are called the *ends* or *endpoints* of the link. A node may exist in a graph and not belong to any link.

The sets V and E are usually taken to be finite, and many of the well-known results are not true (or are rather different) for infinite graphs because many of the arguments fail in the infinite case. The *order* of a graph is $n = |V|$ (its number of nodes). A graph's *size* is $m = |E|$ (its number of links). The *degree* of a node is the number of links that are connected to it, where a link that connects to the node at both ends (a loop) is counted twice. For a link $\{u, v\}$, graph theorists sometimes use the notation uv.

An important property of a graph is its *degree sequence*.

Definition 1. Degree sequence $\{d_j\}$*: the degree sequence of an undirected graph is the sequence of integers* d_j *with* d_j *equal to the number of nodes of degree* j *for all* $j \geq 0$*.*

For example, a degree sequence $(0, 3, 1, 1, 2)$ means that there is no node with degree 0, three nodes with degree 1, one with degree 2, one with degree 3 and two with degree 4. If a degree sequence is given, it is easy to count the number of links $|E| = \frac{\sum_j j d_j}{2}$, half of sum of all degrees. Note that the only constraint for a degree sequence is that the sum $\sum_j j d_j$ must be an even number.

Example 1. *Given a degree sequence* $(0, 0, 2, 2)$*, that is two nodes with degree* 2 *and two nodes with degree* 3*, Figure 2.1 shows two possible graphs.*

In practice, the degree distribution is sometimes used in place of the degree sequence.

Definition 2. Degree distribution $\{a_j\}$*:* a_j *is the fraction of nodes in the graph with degree* j*. Thus if there are* n *nodes in total and* d_j *of them have degree* j*, we have* $a_j = d_j/n$*. In this thesis* P^G *denotes the node degree distribution of a graph* G*, and* P_j^G *is a synonym of* a_j*.*

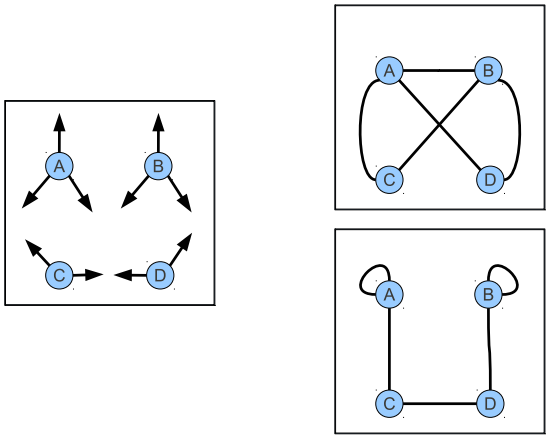

Figure 2.1: Two possible constructions of a graph.

The generating function of a degree distribution is defined as $g(z) = \sum_j a_j z^j$. We then obtain some direct results. For instance, the sum of degree distribution must be 1, *i.e.* $g(1) = 1$[1]; and the average degree is expressed as $\delta = g'(1) = \sum_j j a_j$.

2.1.2 Special distributions

We present have some typical degree distributions often used in the domain of complex networks. The frequently used types in this thesis are *Regular*, *Poisson* and *Power-law*. In each case, we give the explicit a_j and the respective generating function $g(z) = \sum_j a_j z^j$:

- Regular r: $a_r = 1$ and $g(z) = z^r$. Some lattices and rings have a regular degree distribution.
- Poisson λ: $a_j = \frac{e^{-\lambda}\lambda^j}{j!}$ and $g(z) = e^{\lambda(z-1)}$, where λ, the parameter of Poisson, is a positive real value (the average of considered values), see Figure 2.2, left.
- Power-law α: $a_j = Cj^{-\alpha}$ and $g(z) = C\sum_j j^{-\alpha} z^j$, where C is the coefficient of normalization $C = \frac{1}{\sum_j j^{-\alpha}}$ and α is a constant parameter of the power-law distribution known as the *exponent* or *scaling parameter*, see Figure 2.2, center.
- Power-law α with exponential cutoff β: $a_j = Cj^{-\alpha}\exp\left(-\frac{j}{\beta}\right)$. Such distributions are also called truncated power-law, see Figure 2.2, right.

[1]Generating functions provide an efficient way to deal with degree distribution [NSW01].

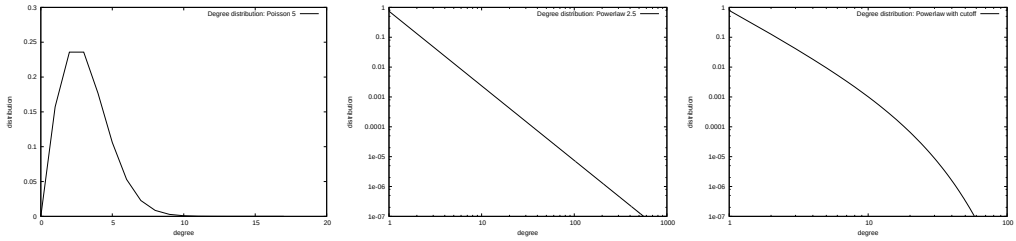

Figure 2.2: Left: distribution Poisson 5; Center: distribution of power-law 2.5, Right: distribution of power-law 2.5 with cutoff 10.

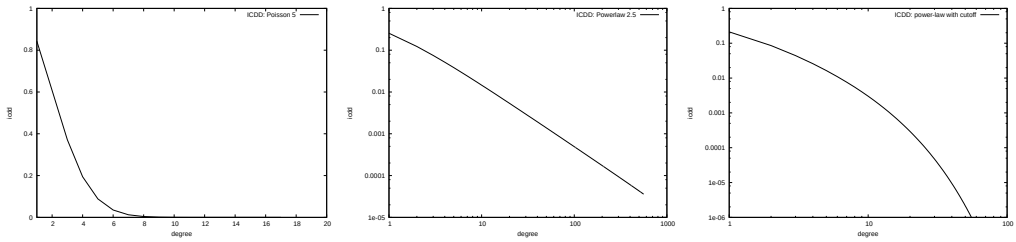

Figure 2.3: Left: ICDD of Poisson 5; Center: ICDD of power-law 2.5; Right: ICDD of power-law 2.5 with cutoff 10.

For readability reasons, in particular in the case of empirical heterogeneous distributions, it is often more convenient to plot Inverse Cumulative Degree Distribution (ICDD), which is in bijection with the original distribution.

Definition 3. ICDD *Inverse Cumulative Degree Distribution: given a distribution* $\{a_j\}$*, the corresponding ICDD is* $ICDD(j) = 1 - \sum_{k=1}^{j} a_k$.

All ICDD are monotone decreasing from 1 to 0. See examples in Figure 2.3.

2.1.3 Comparing two distributions

Comparing distributions is a challenge in itself, for which no general automatic procedure is commonly accepted. To perform this, we will use two classical statistical tests, Kolmogorov-Smirnov test [EDJ+08] and the Statistical Distance [Bas89], defined as:

Definition 4. Kolmogorov-Smirnov test*:*
KS test is the maximum difference of the cumulative values of two distributions P *and* Q*:* $KS(P,Q) = max_i|\sum_{k=1}^{i}(P_k - Q_k)|$.

Definition 5. Statistical Distance*:*
The SD gives the sum of the absolute difference between two degree distributions P *and* Q*:* $SD(P,Q) = \sum_{k=1} |P_k - Q_k|$.

Example 2. *Let us consider for instance the two distributions:*
$P = (0.1, 0.1, 0.2, 0.3, 0.1, 0.2)$ *and*
$Q = (0.3, 0.1, 0.1, 0.2, 0.2, 0.1)$
As shown in Table 2.1, $KS(P, Q) = 0.2$ *the fourth row at* $i = 1$ *or* $i = 2$ *and*

Table 2.1: Computation of KS and SD

j	1	2	3	4	5	6
P_j	0.1	0.1	0.2	0.3	0.1	0.2
Q_j	0.3	0.1	0.1	0.2	0.2	0.1
$\lvert P_j - Q_j \rvert$	0.2	0	0.1	0.1	0.1	0.1
$\lvert \sum_{k=1}^{j} (P_k - Q_k) \rvert$	0.2	0.2	0.1	0	0.1	0

$SD(P, Q) = 0.6$ *is the sum of the third row.*

KS value and the SD value are between 0 and 2. They both have their own advantages and disadvantages, and provide complementary information: KS test may be seen as a worst case, and the SD as an average one.

2.1.4 Random graphs

ER model

In graph theory, Erdös Rényi (ER) model [Bol01, ER60] is one that sets a link between each pair of nodes with equal probability, independently of the other links, also noted as $G(n, p)$ model. Each link is included in the graph with probability p.

A graph in $G(n, p)$ has on average $\binom{n}{2}p$ links. The distribution of the degree of any particular node is binomial:

$$P(deg(v) = k) = \binom{n-1}{k} p^k (1-p)^{n-1-k}$$

where n is the total number of nodes in the graph.

Configuration model

To deal precisely with a random graph, sometimes a multiple one or one with self-loop links (links started and terminated at the same node), with a given degree sequence $\{d_j\}$, we use the *configuration model* [Bol01]: for each node of degree j, we create j copies, and then the links of the graph are determined by two copies according to a uniformly random matching among all these copies. In the process of matching, two copies associated by a link mutually call *partner*.

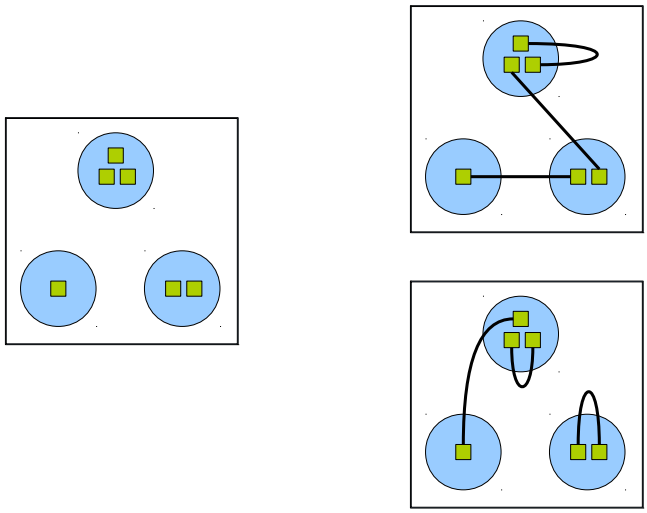

Figure 2.4: The configuration model with a given degree sequence $(1, 1, 1)$, namely one node with degree 1, one with degree 2 and one with degree 3. The left part corresponds the given degree sequence and the right part are two examples (but not all) of random matching among these copies. The generated graph may have multiple links and self-loop links.

Example 3. *See Figure 2.4. The round points represent nodes and the square points represent copies. Here, the configuration model with a given degree sequence* $(1, 1, 1)$*, namely one node with degree* 1*, one with degree* 2 *and one with degree* 3*. The left part of figure corresponds to the given degree sequence* $(1, 1, 1)$ *and the right part consist in two examples (but not all) of random matching among these copies. Attention: the generated graph may have multiple links and self-loop links.*

Given a degree distribution, the procedure of a graph generator consists of two main steps: (1) generate a random sequence from a specified distribution, parameters [EHP00], see also [1]. (2) generate random graphs from a sequence [VL05], see also [2].

2.2 State-of-the-art

Only a few previous works deal with a theoretical approach to the problem of bias in traceroute-like measurements. They confirm the previous experimental work, which showed that this bias, although it decreases when the size of the measurement growes, has a crucial impact on observations [GLM06, GL05].

In this section, we review the main theoretical contributions published on this topic, first show the necessary assumptions to idealize the union of traceroute

paths as a BFS tree (see Subsection 2.2.1). The degree distribution of a BFS tree has been proved a biased one (see Subsection 2.2.2).

2.2.1 Measurement from m monitors to k destinations [DAHB+06]

Many Internet measurement projects build Internet maps by collecting data from multiple monitors [9, 10, 11, 8, 3]. The general strategy consists in acquiring a partial view of the network from each monitor and merging these views in order to get a presumably accurate global map. This provides a map, which remains incomplete but may be very large. Properties of such maps are supposed to be reasonable approximations of the properties of the actual Internet.

More formally, let us consider a sparse undirected graph $G = (V, E)$ with nodes $V = \{v_1, v_2, ...v_n\}$ and links E. Then let us define a set of monitor nodes $S = \{i_1, i_2, ...i_{n_s}\}$ and a set of target nodes $T = \{j_1, j_2...j_{n_t}\}$ describing the deployment of n_s monitors and n_t targets. For each monitor-target pair, we compute a shortest path connecting them and merge all obtained paths. We call this process a (n_s, n_t)-traceroute measurement and denote the obtained graph by $G' = (V', E')$. It is a partial view (a subgraph) of G.

Unfortunately, even with a graph merging from a great number of traceroute paths, the estimate of the degree distribution is still biased. First we check from which factors this bias originates and we focus on two basic problems: node-detection and link-detection.

The main purpose is to compute the probability of link-detection and node-detection which are functions of n_s, n_t and the topology of the underlying graph.

First, we introduce some notations:

- $\sigma_i^{(l,m)}$: that takes the value 1 if the node i belongs to the path between nodes l and m, and 0 otherwise.
- $\sigma_{i,j}^{(l,m)}$: that takes the value 1 if the link ij belongs to the path between nodes l and m, and 0 otherwise.
- $\delta_{i,j}$: Kronecker's delta is a function of two variables i and j, usually integers, which is 1 if they are equal and 0 otherwise.
- ρ_S: the density of monitors $\rho_S = \frac{n_s}{n}$.
- ρ_T: the density of targets $\rho_T = \frac{n_t}{n}$.
- ϵ: the density of probes $\epsilon = \frac{n_s n_t}{n}$.
- $\langle ... \rangle$: denotes the average.
- b_i: node betweenness, the number of shortest paths between pairs of nodes in the network that pass through node i, $b_i = \sum_{l \neq m \neq i} \left\langle \sigma_i^{(l,m)} \right\rangle$.

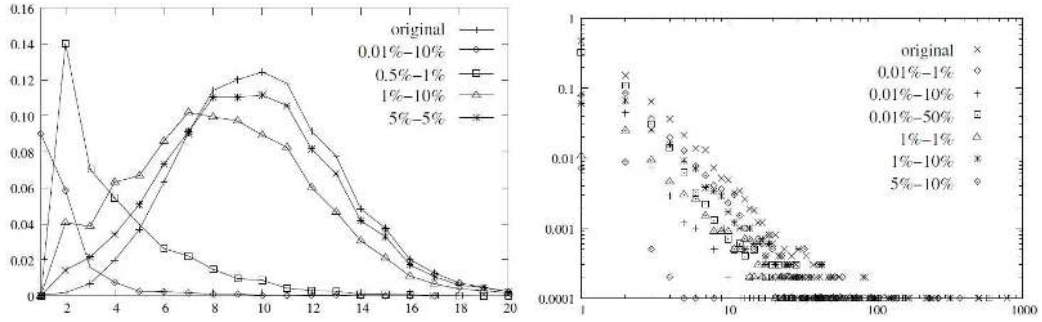

Figure 2.5: Left: Poisson graph (from ER model) with average degree 10; Right: Power-law graph (from MR model) with exponent 2.5. Both graphs have 10000 nodes. The notation $x\% - y\%$ means that $\rho_S = x$ and $\rho_T = y$.

- $b_{i,j}$: edge betweenness, the number of shortest paths between pairs of nodes in the network that pass through link ij, $b_{ij} = \sum_{l \neq m} \left\langle \sigma_{i,j}^{(l,m)} \right\rangle$.

Theorem 1. *[DAHB+06] If the situation of the measurement satisfies the condition $\rho_S \rho_T \ll 1$, then the average discovery probability of a link is*

$$E(\pi_{i,j}) \sim 1 - \exp(-\rho_S \rho_T b_{ij}) \tag{2.1}$$

Theorem 2. *[DAHB+06] If the situation of the measurement satisfies the condition $\rho_S \rho_T \ll 1$, then the average discovery probability of a node is*

$$E(\pi_i) \sim 1 - \exp(-\rho_S \rho_T b_i) \tag{2.2}$$

The betweenness (see also [Bra01, Fre77]) gives a measure of the amount of all-to-all traffic that goes through a link or a node, if the shortest path is used as the metric defining the optimal path between pairs of nodes, and it can be considered as a non-local measure of the *centrality* of a link or a node in the graph.

The edge betweenness has values between 2 and $n(n-1)$. If the densities of monitors and targets are small but finite in the limit of large n, it is clear that links with low betweenness have $\langle \pi_{i,j} \rangle \sim O(N^{-1})$. This fact readily implies that in real situations the discovery process is generally not complete, a large part of low betweenness links being not discovered, and that the network sampling is made progressively more accurate by increasing the density of probes ϵ.

According to Theorem 1, a great fraction of links cannot be discovered while $\epsilon \ll 1$. This fact implicates that the degree distribution under the same condition is biased. In [GLM06], there are simulation results on degree distribution.

For ER graphs with low average degree Poisson, as shown in Figure 2.5: Left, the obtained degree distribution converges quite slowly. For configuration model with power-law, although lines that fit the degree distribution coincide not well, they are really parallel ones. As we know, the parallel lines in a log-log scale plot

must have the same exponent α. In both cases, in order to obtain an enough accurate sampling, at least 5% nodes should be taken as monitors and targets. Unfortunately, the number of monitors are strictly constrained by social reasons, only a relative small number of monitors can be handled in a real simulation.

2.2.2 Degree distribution of BFS trees [AKCM05]

The degree distribution of a BFS tree from a configuration model graph is proved to be a power-law, in the limit of parameters of regular graphs and Poisson graphs. In general case, we have:

Theorem 3. *Let G be a random multi-graph with a sober [2] degree distribution $\{a_j\}$, and assume that G is connected. Let T be a BFS tree on G, and let A_j^{BFS} be the number of nodes of degree j in T . There exists a constant $\eta > 0$ such that with high probability [3], $|A_j^{BFS} - a_j^{BFS} n| < n^{1-\eta}$ for all j, where*

$$a_{m+1}^{BFS} = \sum_i a_i \left[\int_0^1 i t^{i-1} \binom{i-1}{m} p_{vis}(t)^m \left(1 - p_{vis}(t)\right)^{i-1-m} dt \right]$$

$$p_{vis}(t) = \frac{1}{\sum_j j a_j t^j} \sum_k k a_k t^k \left(\frac{\sum_j j a_j t^j}{\delta t^2} \right)^k$$

More concisely, if $g(z) = \sum_0^\infty a_j z^j$ is the generating function of the degree distribution $\{a_j\}$ of G, then a_j^{BFS} is the coefficient of z^j in

$$g^{BFS}(z) = z \int_0^1 g' \left[t - \frac{(1-z)}{g'(1)} g' \left(\frac{g'(t)}{g'(1)} \right) \right] dt \tag{2.3}$$

The requirement of sober ensure that the graph G is connected with a high probability. This assumption is reasonable in practice, because the isolated nodes have not much sense in our research. Then we give three examples: regular graphs, Poisson graphs and power-law graphs.

Regular graphs

Using Equation (2.3), we derive the observed degree distribution of a BFS tree from a random regular graph G_r:

Lemma 1. *The generating function for a r-regular graph is $g(z) = z^r$, the expected degree distribution of BFS T is given by:*

$$a_{m+1}^{BFS} = \frac{\Gamma(r)\,\Gamma\left(m + \frac{1}{r-2}\right)}{\Gamma\left(r + \frac{1}{r-2}\right)(r-2)\,m!} \tag{2.4}$$

[2] A degree distribution $\{a_j\}$ is *sober* if $a_j = 0$ for $j < 3$, and there exist constants $\alpha > 2$ and $C > 0$ such that $a_j < C j^{-\alpha}$ for all j. In fact, if the tested graphs is guaranteed to be connected, the assumption of sober is not necessary.

[3] a sequence of events ϵ_n occurs *with high probability* if $Prob[\epsilon_n] = 1 - o(1)$ as $n \to \infty$.

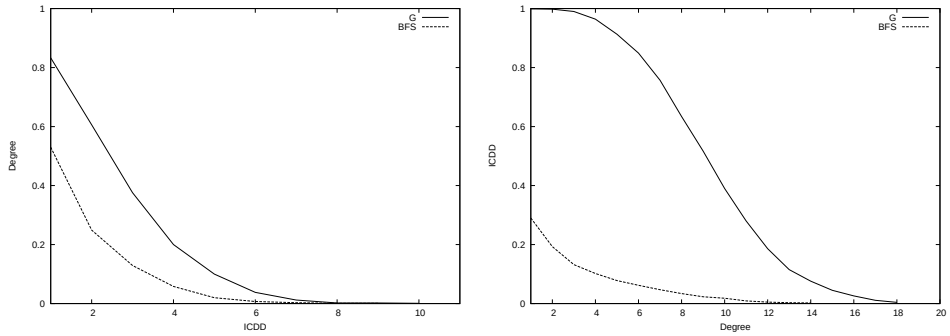

Figure 2.6: The degree distribution of BFS from Poisson 3 graph and Poisson 10 with 1000 nodes.

then, $a_{m+1}^{BFS} \sim \frac{1}{m(r-2)}$[4] *for large* r*.*

Proof. Note that $\Gamma(m) < \Gamma(m+\epsilon) < \Gamma(m)m^{\epsilon}$, for all $m \geq 2$ and all $0 < \epsilon < 1$, therefore, for $m \geq 2$, a_{m+1}^{BFS} is bounded as follows:

$$\frac{m^{-1}}{r^{\frac{1}{r-2}}\,(r-2)} < a_{m+1}^{BFS} < \frac{m^{-1+\frac{1}{r-2}}}{r-2} \tag{2.5}$$

For any fixed r, this gives a power-law degree distribution, and where r is large, $a_{m+1}^{BFS} \sim \frac{1}{m(r-2)}$. □

Poisson graphs

Lemma 2. *The generating function for a Poisson graph is* $g(z) = e^{-\lambda(1-z)}$*, the expected degree distribution of BFS* T *is given by:*

$$a_{m+1}^{BFS} = \frac{\Gamma(m)}{\lambda m!}\,(1 - o(1)) \tag{2.6}$$

Lemma 2 implies that for a random Poisson graph, the degree distribution of BFS T is $a_{m+1}^{BFS} \sim \frac{1}{\lambda m}$.

In Figure 2.6, we show the ICDD curves of Poisson graphs (the left is parameterized 3 and the right is parameterized 10) and their BFS trees. The X-ordinate is the degrees and the Y-ordinate is the value of ICDD. In both two cases, the curves of the BFS trees have an obviously bias on degree distribution.

Power-law graphs

For power-law-α graph, the corresponding generating function of degree distribution is $g(z) = \frac{\sum_j j^{-\alpha}}{\sum_j j^{1-\alpha}} = \frac{\zeta(\alpha)}{\zeta(\alpha-1)}$, where $\zeta(\alpha)$ is the Riemann zeta function

[4]The $\sim$ sign means "approximately equal", in the precise sense that the ratio of both terms tends to 1 as n gets large.

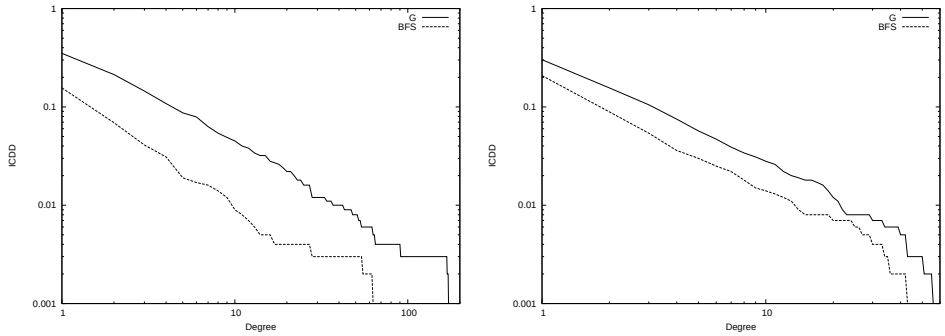

Figure 2.7: The degree distribution of BFS from power-law 2.1 graph and power-law 2.3 with 1000 nodes.

[Edw01, Ivi03] that is defined as $\zeta(\alpha) = \sum_{j=1}^{\infty} j^{-\alpha}$. There is not yet a close and explicit calculation for a general real α. Because there is no explicit form of $g(z)$, some researches [CGW07] have been conducted using some reasonable approximations.

Lemma 3. *Let v be a node with degree j in graph G, Then*

$$E\left[deg_{BFS}(v)|deg_G(v) = j\right] \approx \frac{j\,(j-1)}{j+3} \tag{2.7}$$

With Equation (2.7), we observe that for high degree nodes, the factor $\frac{j-1}{j+3}$ is asymptoticly to 1. That is to say, the estimated exponent converges to the underlying one. However, nodes of low and moderate degree have a large effect on numerical estimates of the exponent, especially in finite-sized graphs or when the average degree of the underlying graph is relatively large.

In Figure 2.7, we show the ICDD curves of power-law graphs (The left is parameterized 2.1 and the right is parameterized 2.3) and their BFS trees. The X-ordinate is the degrees and the Y-ordinate is the value of ICDD. Both X-ordinate and Y-ordinate are in log-log scale. We observe that both the degree distribution of the original graph and that of the BFS tree have a power-law, but they are not coincident. That is to say, the estimation of degree distribution has a bias on the parameter of the power-law.

Chapter 3

Deciding from a BFS, the number of nodes n, and the number of links m

Contents

3.1 Introduction

As deciding on the degree distribution is a challenging task, we will make a few important assumptions in order to make a first step towards this ambitious goal. We first assume that the order of the graph, *i.e.* its number of nodes n, is given. In the case of the Internet, this is a reasonable assumption [VBD+07, HPG+08]. In addition, we will assume that the underlying graph is a random graph with either a Poisson or power-law degree distribution. And we also assume that we have a *complete* BFS of the considered graph: all nodes (but not all links) in the graph are reached by the exploration. Finally, we assume that the number of links m of the graph is known. It is clear that these assumptions are very strong, and are not attainable in practice. We however consider them as reasonable for a first approximation, and give hints of how to reduce the knowledge we used in the next chapters.

We describe our methodology that decides on the type of a graph with the number of nodes, the number of links and a complete BFS tree in Section 3.2. It relies on several strategies to infer a degree distribution from a BFS and we detail the procedure in Section 3.3 and the algorithm in Section 3.4. In Section 3.5 and Section 3.6 we experimentally evaluate the validity of our approach on random model graphs and real-world graphs respectively.

3.2 Methodology

Our methodology is sketched in Figure 3.1. It aims at deciding the type of the degree distribution of an unknown graph G from one of its BFS tree T, its number of nodes n and its number of links m obtained through a measurement. To do so, we consider the two following hypotheses:

- (H1) G has a Poisson degree distribution $P^{(1)}$: $P^{(1)}_j = \frac{\lambda^j e^{-\lambda}}{j!}$, in which $\lambda = \frac{2m}{n}$ is the average degree.
- (H2) G has a power-law degree distribution $P^{(2)}$: $P^{(2)}_j = Cj^{-\alpha}$,where C is a normalizing coefficient and the average degree is $\frac{\sum j^{1-\alpha}}{\sum j^{-\alpha}}$.

For each hypothesis, we build a graph according to a particular strategy which is detailed in Section 3.3, thus obtaining $G1$ and $G2$, respectively. To compare two distributions, we will use different *distances between two degree distribution* D (such as KS or SD). Our expectation is that if hypothesis $H1$ is true (G is Poisson) then the degree distribution P^{G1} of $G1$ will be closer to the theoretical distribution $P^{(1)}$ than P^{G2} to $P^{(2)}$, $D(P^{G1}, P^{(1)}) < D(P^{G2}, P^{(2)})$, and conversely if $H2$ is true (G is power-law) then the degree distribution P^{G2} of $G2$ will be closer to $P^{(2)}$ than P^{G1} to $P^{(1)}$.

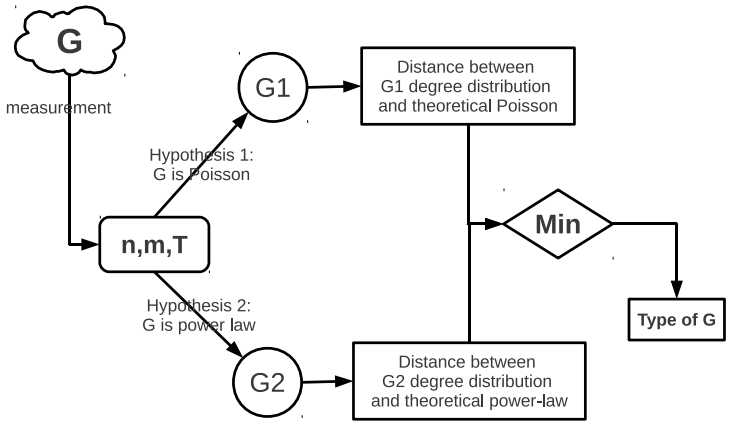

Figure 3.1: Schema of our method. G is an unknown graph on which we perform a measurement which gives its number of nodes n, its number of links m and a BFS T. We then consider two different hypotheses: G has a Poisson degree distribution with average degree λ or it has a power-law degree distribution with exponent α. We build two graphs $G1$ and $G2$ each with a strategy in accordance with the corresponding hypothesis. We then compare the degree distribution of $G1$ to the expected one of G if hypothesis 1 is true, and the one of $G2$ to the expected one of G if hypothesis 2 is true. The hypothesis which leads to the most similar degree distributions is expected to be correct.

We experimentally assess the validity of this approach by applying it to cases where we know the original graph G (we obtain such graphs using models in Section 3.5 and using real-world data in Section 3.6). We then compare the expected theoretical degree distribution to the ones of the graphs obtained from each strategy and check conformance of results with expectations.

3.3 Strategies for building graphs

Starting from a BFS T of a graph G with n nodes and m links and a hypothesis (H1 or H2) on the degree distribution of G (type Poisson or power-law), our objective here is to iteratively add $m - n + 1$ links to T in order to build a graph G' with n nodes, m links, and degree distribution similar to the one of G. We define different link addition strategies according to the supposed type of G, Poisson or power-law. In each case, we also show how to compute the expected degree distribution of the resulting graph, without explicitly building them.

3.3.1 H1: G is Poisson (RR strategy)

Suppose G is a Poisson random graph. It may therefore be seen as the result of an Erdös Rényi (ER) construction [Bol01]: starting with n nodes and no link, m links are uniformly chosen among the $\frac{n(n-1)}{2}$ possible pairs of nodes. The expected node degree distribution obtained this way follows a Poisson law: $P_k = \frac{\lambda^k e^{-\lambda}}{k!}$ where $\lambda = \frac{2m}{n}$ is the average degree.

We may think of building a graph G' similar to G by using a variant of the ER construction: starting with the n nodes and $n - 1$ links of a BFS tree T, the $m - n + 1$ missing links are randomly added as in ER model.

But then T may not be a possible BFS tree of the rebuilt graph G': any link in G which is not in T is necessarily between two nodes in consecutive levels of T, or in the same level of T (otherwise T would not be a shortest path tree and thus not a BFS, see Figure 3.2). We call those positions at which we may add a link *allowed positions*: in order to ensure that T is also a possible BFS tree of G' we add links only between nodes in consecutive levels or in the same level. Since both endpoints of links are randomly chosen, we call this construction RR (Random-Random) strategy.

We now show that the expected node degree distribution of G' obtained with RR strategy can be directly computed from n, m and T without explicitly constructing G'.

Let n_k denote the number of nodes at level k in the BFS tree T. For each node v at level k with degree j (one father node and $j - 1$ son nodes), three kinds of links may be added: (1) to upper level $k - 1$, there are $n_{k-1} - 1$ allowed positions, all nodes in the upper level except the father of v; (2) to the same level of v, there are $n_k - 1$ allowed positions, all nodes in this level except v itself; (3) to lower

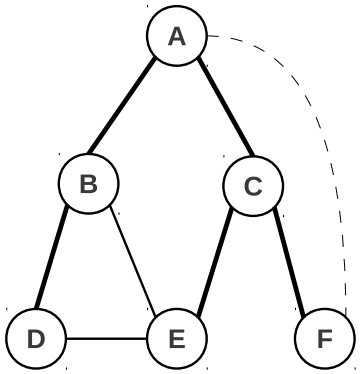

Figure 3.2: Let us consider a part of a BFS of G, composed of a set of links A-B, A-C, B-D, C-E and C-F. Links B-E and D-E may also be present in G, but not link A-F.

level $k+1$, there are $n_{k+1} - j + 1$ allowed positions, all nodes at lower level except the $j-1$ sons of v.

Let S_{jk} be the number of allowed positions linked to a node at level k having j neighbors: $S_{jk} = n_{k-1} + n_k + n_{k+1} - j - 1$. On the other hand, let S be the total number of allowed positions in a BFS tree T: $S = \sum_k \left(n_k n_{k+1} - n_{k+1} + \frac{n_k(n_k-1)}{2} \right)$. Therefore, if we make a random choice, the probability that this position links to a given node v (at level k with degree j) is $\frac{S_{jk}}{S}$. We finally obtain the following theorem.

Theorem 4. *Given a tree T with n nodes, if we construct a graph G' using RR strategy, then the expectation of a node v with degree l in G' is:*

$$E\left(d_{G'}(v) = l\right) = \frac{1}{n} \sum_{k>0} \sum_{j=1}^{l} n_{jk} P\left(j \to l, k\right) \tag{3.1}$$

where n_{jk} is the number of nodes with degree j at level k in T and $P\left(j \to l, k\right) = \binom{m-n+1}{l-j} \left(\frac{S_{jk}}{S}\right)^{l-j} \left(1 - \frac{S_{jk}}{S}\right)^{m-n+1-(l-j)}$ is the probability that a node v with degree j at level k becomes a node of degree l in G' after $m-n+1$ links have been added.

As each newly added link impacts two nodes, the values of $P(j \to l, k)$ are independent from the other $P(j' \to l', k')$. We will consider that this impact can be ignored, when n and m is large enough. From this result, one may estimate the

expectation of the degree distribution of G' from n, m and T, without constructing it explicitly. This is of high interest in practice, since it allows to compute the expectation of degree distribution and compare it with degree distributions obtained by the constructing strategies. Going further would however need precise results on the expectation of degrees in T, which is a difficult problem [AKCM05].

3.3.2 H2: G is power-law (PP strategy)

Suppose now that G is a power-law graph. We therefore aim at designing a process which builds from a BFS T of G, a graph G' with power-law node degree distribution. To do this, as before, we add $m - n + 1$ links between nodes in appropriate levels of T. However, these pairs of nodes are no longer chosen uniformly at random. Instead, we use a selection schema inspired from the preferential attachment of the classical Barabási-Albert (BA) model [BAJ99, BA99, PR04]: we choose (in the appropriate levels) nodes randomly with probability proportional to their degree in T. As we choose both endpoints of added links according to preferential attachment, we call this procedure PP (Preferential-Preferential) strategy.

We now show how to compute, for this strategy, the expected obtained degree distribution.

Theorem 5. *Given a tree T with n nodes, if we construct a graph G' using PP strategy, then the expectation of a node v with degree l in G' is:*

$$E\left(d_{G'}(v) = l\right) = \frac{1}{n} \sum_{k>0} \sum_{j=1}^{l} n_{jk} P(j \to l, k, m') \tag{3.2}$$

where n_{jk} is the number of nodes with degree j at level k in T and $P\left(j \to l, k, m'\right)$ is the probability that a node v with degree j at level k in T is constructed as a node with degree l in G' after m' links have been added to T.

Computation process. The term $P(j \to l, k, m')$ may be obtained recursively:

$$\begin{aligned} P\left(j \to l, k, m'\right) = &\left(1 - \theta\right) P\left(j \to l, k, m' - 1\right) \\ &+ \theta\, P\left(j + 1 \to l, k, m' - 1\right) \end{aligned} \tag{3.3}$$

We split $P(j \to l, k, m')$ into two parts which correspond to two cases (linked to this node or not) when a new link is added. The probability that a newly added link changes the node from degree j to $j + 1$ is denoted by θ.

In the following, we compute θ using four terms:

- $\theta 1$: the probability that the node v is selected as the first endpoint of the newly added link.
- $\theta 2$: the probability that a node at upper level $k - 1$ is selected as the first endpoint, and v is selected as the second endpoint.

- $\theta 3$: the probability that a node (except v) at level k is selected as the first endpoint, and v is selected as the second endpoint.

- $\theta 4$: the probability that a node at lower level $k+1$ is selected as the first endpoint, and v is selected as the second endpoint.

The sum of degrees of all nodes at the level k after t links have been added is $d_{k,t}$. In the following, we give the details of how to calculate for the case of PP strategy.

$$\begin{aligned}
\theta 1 &= \frac{j}{2\,(n-1+t)} \\
\theta 2 &= \frac{d_{k-1,t}}{2\,(n-1+t)} \frac{j}{d_{k-2,t}+d_{k-1,t}+d_{k,t}} \\
\theta 3 &= \frac{d_{k,t}-j}{2\,(n-1+t)} \frac{j}{d_{k+1,t}+d_{k,t}+d_{k-1,t}} \\
\theta 4 &= \frac{d_{k+1,t}}{2\,(n-1+t)} \frac{j}{d_{k+2,t}+d_{k+1,t}+d_{k,t}}
\end{aligned}$$

The next problem is how to compute $d_{k,t}$. We denote by $K_{k\to k'}$ the probability that a directed link is added from level k to k', where k' must be k, $k-1$, $k+1$.

$$\begin{aligned}
d_{k,t} &= K_{k\to k}\,(d_{k,t-1}+2) + (K_{k\pm 1\to k} + K_{k\to k\pm 1})\,(d_{k,t-1}+1) \\
&\quad + (1 - K_{k\to k} - K_{k\to k\pm 1} - K_{k\pm 1\to k})\,d_{k,t-1} \qquad (3.4) \\
K_{k\to k'} &= \frac{d_{k,t-1}}{2\,(n-1+t)} \frac{d_{k',t-1}}{d_{k-1,t-1}+d_{k,t-1}+d_{k+1,t-1}} \qquad (3.5)
\end{aligned}$$

Using Equation (3.5) in the expressions Equation (3.4) shows that $d_{k,t}$ is a function of $d_{k',t-1}$, which can be calculated by dynamic programming techniques.

Our computation process is not exact, since we have neglected to take into account possible collision, *i.e.* positions selected several times. However, since we deal with sparse graphs, in the case of model graphs as well as real graphs, the number of links to be added to the BFS tree is much smaller than the number of possible positions, and so there are very few collisions. From a practical viewpoint, in the building process, we just ignore multiple links.

Similar to RR and PP strategies, one may propose two other possible strategies: RP (first endpoint random, second endpoint preferential) strategy and PR strategy whose theoretical degree distribution can be computed in a similar way. Intuitively, these two strategies lead a rebuilt graph with mixed type, neither similar to Poisson nor similar to power-law.

3.4 Algorithm

Finally our approach consists in applying Algorithm 1. We call it $SB_m(RR, PP)$ which means the algorithm that decides on the type with hypothesis m using single BFS tree and rebuild the graph using RR and PP strategies.

Data: The number of nodes n, the number of links m and a BFS tree T of a graph G.
Result: The type of G.

Compute the set of allowed positions $E_{allowed}$ according to T ;
Let $m' \leftarrow m - n + 1$;
The rebuilt graph $G'_1 \leftarrow T$;
while $m' > 0$ **do**
    Randomly (RR strategy) choose a position uv from $E_{allowed}$;
    Add link uv into G'_1: $G'_1 \leftarrow G'_1 + uv$;
    $m' \leftarrow m' - 1$;
end
$m' \leftarrow m - n + 1$;
The rebuilt graph $G'_2 \leftarrow T$;
while $m' > 0$ **do**
    Preferentially (PP strategy) choose a position uv from $E_{allowed}$;
    Add link uv into G'_2, $G'_2 \leftarrow G'_2 + uv$;
    $m' \leftarrow m' - 1$;
end
Compute theoretical distribution $P^{Poisson}$ corresponding to n and m ;
Compute theoretical distribution $P^{Power-law}$ corresponding to n and m ;
$KS_{Poisson} = KS(P^{Poisson}, P^{G'_1})$;
$KS_{Power-law} = KS(P^{power-law}, P^{G'_2})$;
if $KS_{Poisson} < KS_{Power-law}$ **then**
    Return Poisson ;
else
    Return power-law ;
end

Algorithm 1: $SB_m(RR, PP)$: Algorithm of deciding type of the distribution of a graph for m, n and a BFS tree.

Remarks: Notice that in Algorithm 1 we use KS test to compare two distributions. Other comparison methods, such as SD distance, may be used in place of KS test and lead to different conclusion. We will see in the following that, in all the cases we considered, the results were the same using KS and SD. Importantly, one may also obtain the result without explicitly building the graph, using Theorem 4 and Theorem 5.

3.5 Validation using model graphs

Our expectation is that the strategies described in Section 3.3 succeed in building a graph G' similar (regarding degree distribution) to G when the appropriate strategy is used with an appropriate graph (RR if G is Poisson, PP if G is power-law). In addition, we expect that the degree distribution of G' will differ significantly from that of G if a wrong strategy is applied (RR if G is power-law, PP if G is Poisson). In this section we conduct experiments on *model graphs*, *i.e.* random graphs in the classes of Poisson graphs or power-law graphs with given parameters (average degree λ and exponent α respectively). To ensure that the BFS covers all nodes of the graph, we use a program which generates *random simple connected graphs* according to a given degree sequence (sampled from the given degree distribution) [VL05], [2].

For each model graph G, we first extract a BFS tree T from a randomly chosen node, then build G'_{RR} (using RR strategy) and G'_{PP} (using PP strategy), and we denote by P^{RR} and P^{PP}, the corresponding degree distributions. According to the method described in Figure 3.1, we suppose that we know n and m. Therefore, the corresponding parameters λ (for RR strategy), α (for PP strategy) and the corresponding theoretical degree distributions P^{λ} and P^{α} can be derived from n and m. By comparing the values $KS(P^{RR}, P^{\lambda})$ and $KS(P^{PP}, P^{\alpha})$ (or $SD(P^{RR}, P^{\lambda})$ and $SD(P^{PP}, P^{\alpha})$), we may then decide on the type of the underlying graph. If the concluded type is the same as the type that we have used to generate the underlying graph, and so is the same as the actual one, then our method has succeeded.

We conducted experiments on Poisson model graphs with average degree from 3 to 50 (step is 0.1), and power-law model graphs with exponents from 2.01 to 2.50 (step is 0.01), which are typical values used in Internet modeling. In each case, we consider graphs with 10000 and 100000 nodes, and all results in tables and figures are averaged over ten samples. We display below the results for average degrees 3 and 10 and for exponents 2.1 and 2.3, which are representatives of our observations.

The general conclusion of our experiments is that for both cases of Poisson and power-law, we succeed in deciding the type of the underlying graphs: for Poisson model graphs, the type decided by our method is actually Poisson and for power-law graphs, the type decided is actually power-law.

3.5.1 Poisson graphs

In Figure 3.3 we present the results for Poisson graphs with average degrees 3 and 10.

The degree distribution obtained with RR strategy is closer to the original one, as expected. This is confirmed by KS and SD statistics (Table 3.1): the smallest values are obtained with RR strategy.

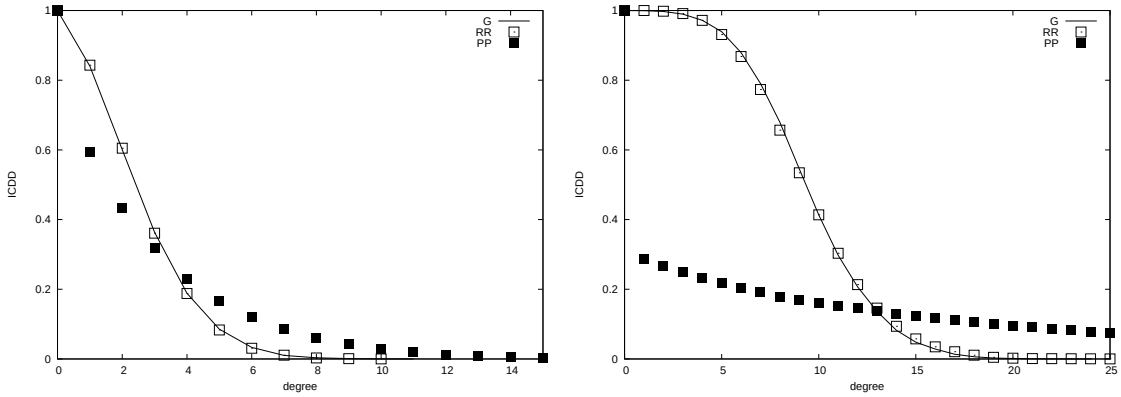

Figure 3.3: Reconstruction of a graph Poisson 3 and Poisson 10. We draw the ICDD (Inverse Cumulative Degree Distribution) for three graphs: the original graph G and the ones obtained with the RR and the PP strategies. Horizontal axis: degree k; vertical axis: fraction of nodes with degree lower than or equal to k.

Table 3.1: KS and SD for Poisson model graphs. The smallest KS or SD values mean the smallest difference on degree distribution.

Poisson 3					Poisson 10				
	$n = 10000$		$n = 100000$			$n = 10000$		$n = 100000$	
	KS	SD	KS	SD		KS	SD	KS	SD
RR	0.043	0.091	0.037	0.075	RR	0.025	0.116	0.014	0.048
PP	0.284	0.610	0.268	0.571	PP	0.119	0.444	0.111	0.446

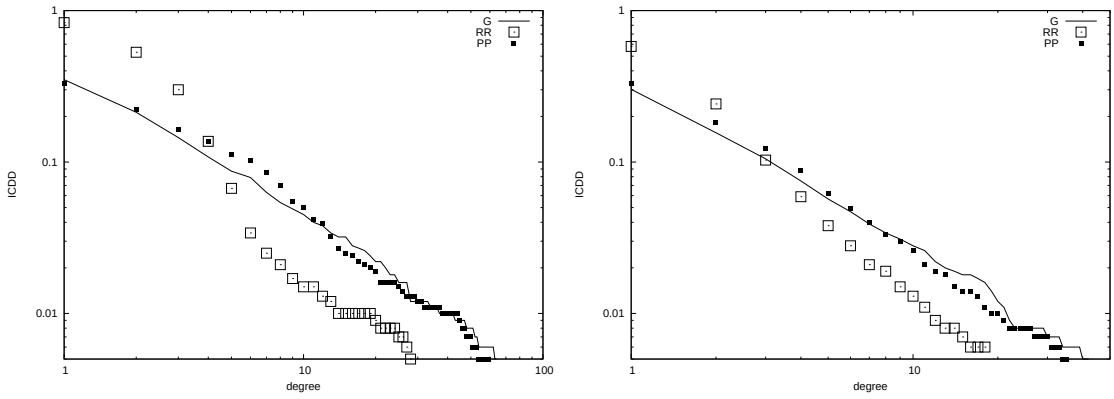

Figure 3.4: Reconstruction of a graph power-law 2.1 and 2.3. We draw the ICDD for three graphs: the original graph G, and the ones obtained with the RR and the PP strategies.

Table 3.2: KS and SD for power-law model graphs

Power-law 2.1					Power-law 2.3				
	$n = 10000$		$n = 100000$			$n = 10000$		$n = 100000$	
	KS	SD	KS	SD		KS	SD	KS	SD
RR	0.201	0.432	0.194	0.405	RR	0.278	0.591	0.274	0.553
PP	0.038	0.138	0.049	0.180	PP	0.030	0.086	0.024	0.095

Notice that a Poisson graph with a higher degree gives better results. This is probably due to the fact that we add more links in this case, and so strategies for doing this make much more difference.

Finally, we conclude that our method succeeds in recognizing random Poisson graphs. This is true for all average degrees (we tested from 3 to 50.), but performs best on graphs with a relatively high average degree.

3.5.2 Power-law graphs

Similar to Poisson model graphs, we conduct our experiments with power-law model graphs.

In Figure 3.4 we present obtained results for power-law graphs with exponent 2.1 and 2.3. To better show the characteristic of the power-law, all plots are in log-log scale. Both the ICDD plot and the statistic test (Table 3.2) support our conclusion for all exponents (we tested from 2.01 to 2.50.) and all sizes.

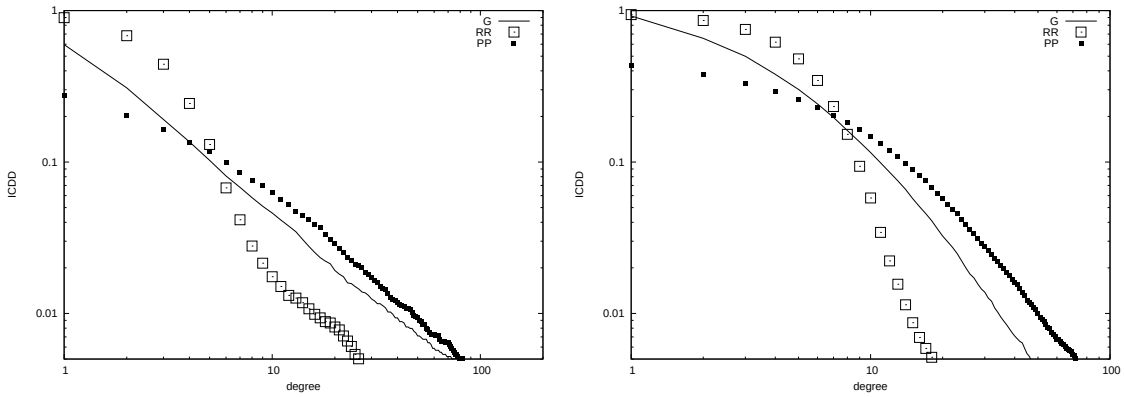

Figure 3.5: ICDD of the Skitter-AS (left) and Radar-japon (right) graph G and the ones obtained with RR and PP strategies.

3.6 Experiments on real-world data

Previous section shows that our method succeeds in making the difference between Poisson and power-law *random graphs*. It is clear however that, in practice, considered graphs have neither perfect Poisson nor power-law degree distribution, and are even not random.

We consider in this section several real-world datasets among the current largest measurements of the Internet topology. Although obtained graphs are still partial views and probably are strongly biased, they constitute current state-of-the-art of available data and we use them as benchmarks.

Like in previous section, for each case of real-world graph G, we consider two hypotheses: (H1) G has a degree distribution close to a Poisson law; (H2) G has a degree distribution close to a power-law. Using strategies RR and PP, we build graphs G'_{RR} and G'_{PP}, respectively. Our expectation is that if (H1) is true then G'_{RR} degree distribution is close to the theoretical Poisson law; if (H2) is true then G'_{PP} degree distribution is close to the theoretical power-law; and the converse is not true.

3.6.1 Skitter-AS graph

We first try our method on an AS-level map collected by Skitter project of CAIDA [KCC+07] (See the related resources in [9]). The obtained graph has 5775 nodes and 12025 links.

Figure 3.5 (left) shows the ICDD obtained with our strategies and shows that the entire Skitter-AS graph follows a degree distribution of type power-law (a perfect power-law in a log-log scale is a straight line). The degree distribution obtained with RR strategy is clearly far from a power-law. The one obtained with PP strategy is much closer. Table 3.3 confirms this, even though the difference between RR strategy and PP strategy is as strong as for model graphs (see Table 3.1

Table 3.3: KS and SD for Skitter-AS graph.

	RR	PP	Decided Type
KS	0.166	0.082	Power-law
SD	0.359	0.235	Power-law

Table 3.4: KS and SD for Radar graphs.

	RR	PP	Decided Type
Radar-japon $n = 26698$, $m = 77545$			
KS	0.074	0.163	Poisson
SD	0.202	0.363	Poisson
Radar-cm $n = 21185$, $m = 15728$			
KS	0.064	0.241	Poisson
SD	0.151	0.521	Poisson
Radar-ortolan $n = 24262$, $m = 48516$			
KS	0.062	0.213	Poisson
SD	0.156	0.476	Poisson
Radar-enix $n = 30433$, $m = 73576$			
KS	0.067	0.219	Poisson
SD	0.187	0.446	Poisson

and Table 3.2).

Finally, our method succeeds in deciding that Skitter-AS graph has a degree distribution close to a power-law.

3.6.2 Radar graphs

A Radar graph is a part of the Internet topology observed by periodic running traceroute-like measurements from one monitor to a set of targets during several weeks [OML08], see also [6], for details and the original data. We use here several instances of Radar graphs, from different monitors: Radar-cm (21185 nodes and 15728 links), Radar-japon (26698 nodes and 77545 links), Radar-enix (30433 nodes and 73576 links) and Radar-ortolan (24262 nodes and 48516 links).

In Figure 3.5 (right), we plot the ICDD of Radar-japon graph and the corresponding ICDDs obtained by RR and PP strategies. The shape of the original ICDD indicates that the degree distribution of the underlying graph is likely to be a mixture of both Poisson and power-law.

Table 3.4 shows numerical results for Radar graphs. All results show that the difference between RR strategy and its corresponding theoretical distribution is smaller than that of PP strategy. Therefore our method decides that the type

of Radar graphs is more likely Poisson even though their distributions have a long-tail. Note that the difference between RR and PP are much smaller than for previous cases, thus indicating that the confidence in the conclusion is poor.

Finally, our method decides that Radar graphs are more likely to have Poisson degree distribution, but with poor confidence (which is in accordance with their actual type).

3.7 Conclusion

In this chapter, we presented a new approach to decide on the type of the degree distribution of a graph when a complete BFS tree T, the number of nodes n and the number of links m are known. According to the presupposed type of the underlying graph, we use RR strategy for Poisson case and PP strategy for power-law case to rebuild graphs expectedly similar to the original one. Then by comparison with the presupposed distribution, we can decide on the type of the graph, either Poisson or power-law. The validation of the methodology that uses RR and PP strategies is first conducted on random model graphs: random Poisson graphs and random power-law graphs. In both cases, our methodology allows to decide well on the type of the underlying graph. Then we apply this methodology to real-world graphs: Skitter-AS graph shows a power-law and Radar graphs a Poisson degree distribution.

As we have mentioned several times, the hypotheses that we use in this chapter are very strong and not attainable in practice. We explore a method to reduce them in the next chapter.

Chapter 4

Deciding without the number of links m

Contents

4.1 Introduction

In Chapter 3, we have introduced a methodology for deciding on the type of the degree distribution of a graph from its number of nodes n, its number of links m and one of its BFS tree T. In this chapter, we develop and improve our methodology to estimate the type without m but only n and T.

Section 4.2 describes the methodology and Section 4.3 the corresponding results on model graphs and real-world graphs.

4.2 Methodology and algorithm

We extend our approach to get information on the degree distribution of a graph when the number of links m is not known but only n and T. In that case we use our rebuilding strategies for a wide range of possible values of m and infer the most probable type of degree distribution as in the previous chapter.

We proceed as follows: for each building strategy, we compute KS distance between the obtained distribution and the theoretical one for a wide range of realistic values of m. We then plot this distance as a function of m and select the value m' which gives the minimum KS value. If the minimum value is given by RR strategy, we conclude that the degree distribution of the original graph is Poisson, whereas if m' is given by PP strategy, we conclude that the degree distribution is power-law. See Figure 4.1.

We can rewrite this procedure as Algorithm 2. We call it $SB(RR, PP)$ which means that we decide without knowing m with a single BFS tree using RR and PP strategy.

The value m_{begin} and m_{end} are assigned with values that correspond to the background of application. A general principle is that m_{begin} should be greater than double number of nodes in order to ensure the connectivity of the tested graph, while m_{end} can be a value of scores times of the number of nodes. The step Δm corresponds to the accuracy of our algorithm. As a variant, we can use parameter λ for Poisson and α for power-law to replace m in the loop and then we use $\Delta\lambda$ and $\Delta\alpha$ as the loop step. Generally, in our work we set $\Delta\lambda = 0.1$ and $\Delta\alpha = 0.01$, holding a good balance between accuracy and speed.

4.3 Experiments

As in the previous chapter, we experiment on model graphs and real-world graphs. Each experiment leads to a decision on the type of the degree distribution of the considered graphs and an estimate of its number of links m. We then compare these results to reality. In the following we only mention the results of KS tests, but the results of SD are similar. In both cases our method succeeds in deciding on the type.

Data: The number of nodes n, a complete BFS tree T.
Result: Type of the underlying graph and estimated number of links m.

Compute the set of allowed positions $E_{allowed}$ according to T ;
$KS_{min} \leftarrow \infty$;
foreach *hypothesis in* {*Poisson, power-law*} **do**
  for $m_{test} \leftarrow m_{begin}$ **to** m_{end} **Step** Δm **do**
    $G'_{hypothesis,m_{test}} \leftarrow T$;
    $m_{add} \leftarrow m_{test} - n + 1$;
    while $m_{add} > 0$ **do**
      if *hypothesis Poisson* **then**
        Randomly (RR) choose a position uv from $E_{allowed}$;
      else
        Preferentially (PP) choose a position uv from $E_{allowed}$;
      end
      Add uv to $G'_{hypothesis,m_{test}}$: $G'_{hypothesis,m_{test}} \leftarrow G'_{hypothesis,m_{test}} + uv$;
      $m_{add} \leftarrow m_{add} - 1$;
    end
    Compute the theoretical distribution $P^{hypothesis,m_{test}}$ corresponding $(hypothesis, m_{test})$;
    $KS_{test} \leftarrow KS(P^{hypothesis,m_{test}}, P^{G'_{hypothesis,m_{test}}})$;
    if $KS_{test} < KS_{min}$ **then**
      $KS_{min} \leftarrow KS_{test}$;
      $type \leftarrow hypothesis$;
      $m \leftarrow m_{test}$;
    end
  end
end
Return $(type, m)$;

Algorithm 2: $SB(RR, PP)$: Algorithm for deciding the type of the degree distribution of a graph from a single BFS tree using RR and PP strategy.

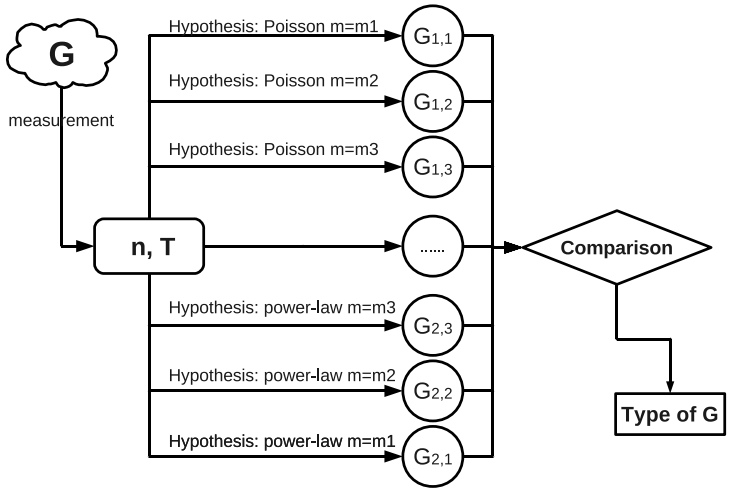

Figure 4.1: Schema for deciding on the type of degree distribution without m with a single BFS tree and n: $G_{1,1}$, $G_{1,2}$, $G_{1,3}$,... are graphs built with RR strategy, respectively with parameters $m1$, $m2$, $m3$. Similarly $G_{2,1}$, $G_{2,2}$, $G_{2,3}$,... are graphs built with PP strategy. For each $G_{i,m}$, we compute KS value between the degree distribution of $G_{i,m}$ and the corresponding theoretical distribution. Then we decide on the type and estimate the number of links by selecting the minimum of these KS values.

Table 4.1: Results for Poisson model graphs

n=1000	KS	m	m'	n=10000	KS	m	m'
Poisson 3	0.068	1500	2085	Poisson 3	0.027	15000	23750
Poisson 5	0.074	2500	2835	Poisson 5	0.025	25000	36250
Poisson 10	0.096	5000	5650	Poisson 10	0.041	50000	68750

4.3.1 Poisson model graphs

We show how the obtained KS test evolves as a function of the value of $\lambda = \frac{2m}{n}$ in Figure 4.2 for a typical case. RR strategy performs much better than PP one, and KS reaches a minimal for a value m' close to the actual value of m.

Table 4.1 shows the results in a different way, for Poisson model graphs of size 1000 and 10000. First, since the minimum KS is always observed in the case of RR strategy, we conclude that the type of graph is Poisson. Second, according to the minimum value of KS we can estimate the value of λ and thus the value of the (unknown) number of links of the original graph.

For example for $n = 1000$, Table 4.1 shows that the minimal value for KS is obtained for $m' = 2085$ for a Poisson 3 graph (whereas the real value is $m = 1500$, corresponding to a Poisson model graph with $\lambda = 3$).

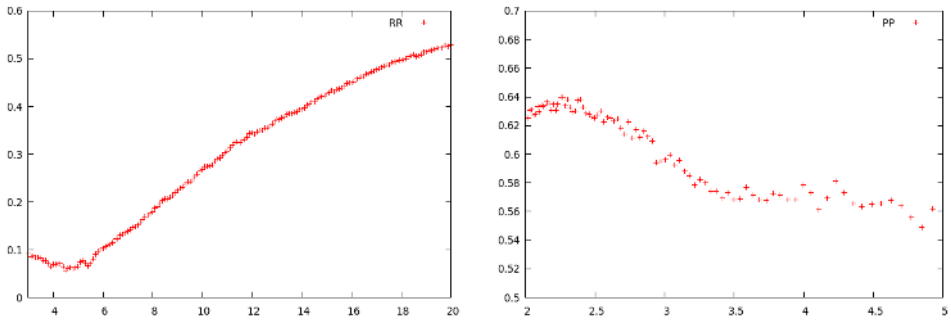

Figure 4.2: Results obtained for a Poisson model graph with average degree 3. We plot KS as a function of the supposed degree. In the left plot we use RR strategy, which shows minimum KS of 0.05 in the region of $\lambda = 5$. In the right figure we use PP strategy , which shows much bigger values of KS. Thus we conclude that the underlying graph is Poisson.

Table 4.2: Results for power-law model graphs

n=1000	KS	m	m'	n=10000	KS	m	m'
power-law 2.10	0.07	4233	3002	power-law 2.10	0.06	42326	28129
power-law 2.15	0.09	3662	2936	power-law 2.15	0.04	36622	25043
power-law 2.20	0.07	3220	2550	power-law 2.20	0.03	32198	25043
power-law 2.25	0.07	2873	2418	power-law 2.25	0.03	28730	21085
power-law 2.30	0.06	2598	2168	power-law 2.30	0.03	25982	21377
power-law 2.35	0.05	2378	2200	power-law 2.35	0.02	23780	20534
power-law 2.40	0.06	2200	2109	power-law 2.40	0.02	21996	20274

Notice that our estimate m' of m is always overestimated. We have no clear explanation for this observation, but will show how to reduce this error in next chapter.

4.3.2 Power-law model graphs

In Table 4.2 and Figure 4.3, we give the results of our experiments for power-law graphs with exponents between 2.10 and 2.40.

Again, the method succeeds in deciding the appropriate type of the graph: with PP, the value of KS test is smaller than with RR. The estimate m' of m is reasonable, but lower than the actual value.

4.3.3 Real-world graphs

We finally apply our method to some real-world graphs, see Table 4.3 and Figure 4.4. As before, our method succeeds in deciding that Skitter-AS graph is very

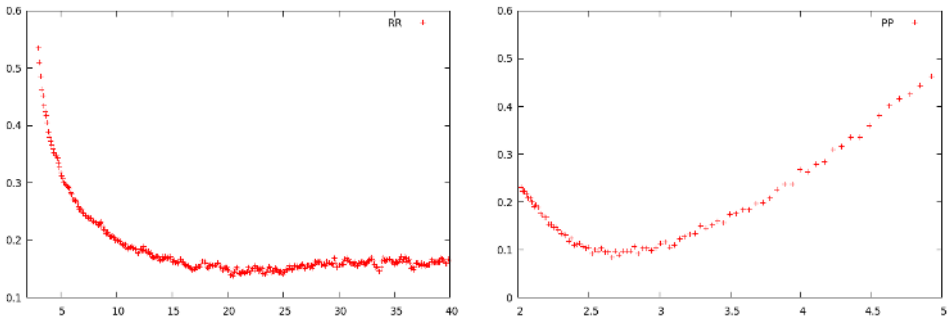

Figure 4.3: Results for a power-law model graph with exponent 2.2. The left plot is the result with the RR strategy and has a minimum greater than 0.1. The right plot is the result with the PP strategy with a minimum less than 0.1.

Table 4.3: Results for real-world graphs

	RR	PP	m	m'	Decided Type
Skitter-AS	0.109	0.091	12025	8297	Power-law
Radar-ortolan	0.150	0.435	48516	50686	Poisson
Radar-japon	0.067	0.162	77545	60899	Poisson
Radar-cm	0.051	0.225	15728	25500	Poisson
Radar-enix	0.058	0.175	73576	68004	Poisson

close to a power-law graph, while Radar graphs are closer to a Poisson graph, but are actually in between. In both cases, we obtain reasonable estimates m' of the number of links m.

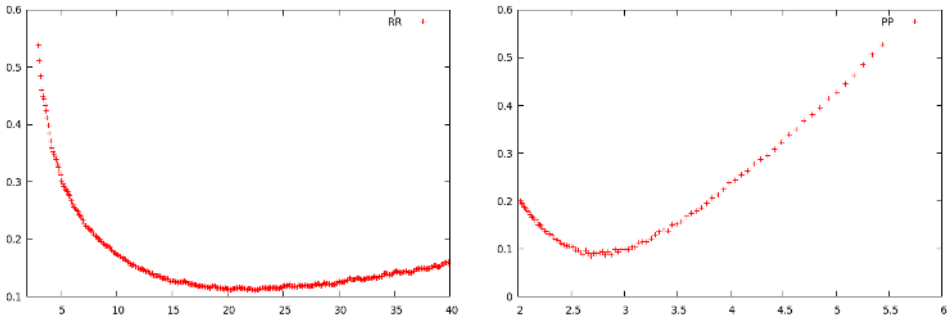

Figure 4.4: KS for Skitter-AS graph.

4.4 Conclusion

We presented in this chapter a method for deciding on the type of the degree distribution of a graph from its number of nodes n and one of its BFS tree T, but

without its number of links m. It consists in trying values of m in wide ranges of possible values, and then by selecting the case in which we obtain the smallest value for KS (or SD) statistics. This provides both a decision on the type of the degree distribution of the considered graph and an estimate of its number of links m. Although this estimate is not very precise, it already provides reasonable insight and, more importantly, in all our experiments the decision on the type is correct.

Chapter 5

Deciding with several BFS trees

Contents

5.1 Introduction

In practice, it is possible to have several BFS-like measurements of the Internet topology by using several monitors. In this chapter we explore how this may be used to improve our results. We now focus on the topic that decides on the type of degree distribution of a graph with multi-BFS trees. First we are interested in the ratio of the link-detection, which impact the number of BFS trees that we use. Then we propose a new schema and algorithm that fit with multi-BFS trees. The experiments are conducted on the same dataset in order to compare with the single version described in the precedent chapter.

5.2 Methodology

Our methodology is sketched in Figure 5.1. As before, G is an unknown graph on which we perform a measurement which gives its number of nodes n and a set of BFS trees $T1$, $T2$... Tk. First, we merge these BFS trees into a graph G_{bfs}. We then consider two different type of hypotheses: (H1) G has a Poisson degree distribution with average degree λ or (H2) it has a power-law degree distribution with exponent α. Both parameters λ and α are equivalent, for the given n, to the number of links m. Then we build two families of graphs $G_{1,m}$ and $G_{2,m}$ in accordance with these two hypotheses and different m. We then compare the degree distribution of $G_{1,m}$ to the expected one of G if (H1) was true, and the one of $G_{2,m}$ to the expected one of G if (H2) was true. The hypothesis which leads to the most similar degree distributions is expected to be correct.

We call $MB(RR, PP)$ denote the algorithm for deciding the type of a graph without knowing m but with multi-BFS trees and using RR and PP strategies. See Algorithm 3.

This algorithm is very similar to $SB(RR, PP)$ (Algorithm 2) except for two different points: (1) $E_{allowed}$ is the intersection of allowed positions of all BFS trees; (2) the rebuilt graph begins with one graph merged from a set of BFS trees.

In this case, it contains the only set of possible positions for the missing links. Figure 5.2 shows an example. BFS 1 is rooted at node 1 and BFS 3 is rooted at node 3. $E1$ and $E2$ are the corresponding sets of allowed positions for these two BFS trees. The right part is the result: the graph is obtained by merging from BFS 1 and BFS 2 and the allowed positions E are the intersection of $E1$ and $E2$. We explore this in more details in the next section.

Two reasons make the use of several BFS trees more accurate than the use of only one BFS tree. First, we build graphs from a merged graph G_{bfs} instead of only a BFS tree T. There is no doubt that G_{bfs} contains more links than T. In addition, each BFS tree comes with its own set of allowed positions for missing links, and the set of allowed positions for G_{bfs} is the intersection of all these sets. It therefore decreases significantly with the number of BFS trees, and so the graph we build is chosen to the original one.

Data: The number of nodes n, several complete BFS trees T_1, T_2...T_k.
Result: Type of the underlying graph and the number of links m.

Compute the set of allowed positions $E_{allowed}$ according to $\{T_i\}$;
$KS_{min} \leftarrow \infty$;
foreach *hypo in* {*Poisson, power-law*} **do**
  for $m_{test} \leftarrow m_{begin}$ **to** m_{end} **Step** Δm **do**
    Let $G'_{hypo,m_{test}} \leftarrow \bigcup T_i$;
    $m_{add} \leftarrow m_{test} - |E(G'_{hypo,m_{test}})|$;
    while $m_{add} > 0$ **do**
      if *hypo* = *Poisson* **then**
        Randomly (RR) choose a position uv from $E_{allowed}$;
      else
        Preferentially (PP) choose a position uv from $E_{allowed}$;
      end
      Add uv into $G'_{hypo,m_{test}}$: $G'_{hypo,m_{test}} \leftarrow G'_{hypo,m_{test}} + uv$;
      $m_{add} \leftarrow m_{add} - 1$;
    end
    Compute the theoretical distribution $D_{hypo,m_{test}}$ corresponding $(hypo, m_{test})$;
    $KS_{test} \leftarrow KS(P^{hypo,m_{test}}, P^{G'_{hypo,m_{test}}})$;
    if $KS_{test} < KS_{min}$ **then**
      $KS_{min} \leftarrow KS_{test}$;
      $type \leftarrow hypo$;
      $m \leftarrow m_{test}$
    end
  end
end
Return $(type, m)$

Algorithm 3: $MB(RR, PP)$: Algorithm for deciding the type of the degree distribution of a graph from with several BFS trees using RR and PP strategy.

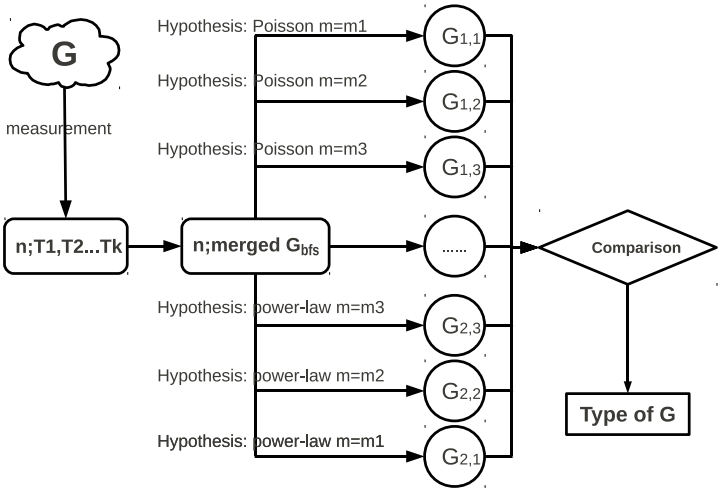

Figure 5.1: Schema of our method, in the case when m is unknown and several BFS trees are given. G is an unknown graph on which we perform a measurement which gives its number of nodes n and a set of BFS trees T_1, T_2... T_k. First, we merge these BFS trees into a graph G_{bfs}. We then consider two different types of hypotheses: (H1) G is Poisson with average degree λ and (H2) G is power-law with exponent α. Both parameters λ and α are equivalent to the number of links m. Then we build two families of graphs $G_{1,m}$ and $G_{2,m}$ in accordance with these two hypotheses and different $m = m1, m2, m3$.... We then compare the degree distribution of $G_{1,m}$ to the expected one of G if (H1) is true, and the one of $G_{2,m}$ to the expected one of G if (H2) is true, for all m. The hypothesis which leads to the most similar degree distributions is expected to be correct.

5.3 Discovery of links with several BFS trees

In this section, we conduct experiments to address the following key problem:

How many BFS trees are needed to decide on the type of the degree distribution of a graph and estimate its number of links?

In parallel, we also explore the impact of the choice of monitors (roots of our BFS trees) for the measurement. To do so, we consider two scenarios:

- Random: we choose roots of BFS trees at random;
- Maxdegree: we choose the nodes with greatest degrees.

We give in Table 5.1, the discovery probability of links with *random* and *maxdegree* strategies for root selection. All graphs have a size of 10000 nodes and are tested with 1, 2, 5, 10, 20, 50 and 100 roots (*i.e.* 1, 2, 5, 10, 20, 50 and 100 BFS trees).

First, for those graphs with high average degree, they need more BFS trees to cover all links of the underlying graph. Such as a Poisson 3 graph need about 20

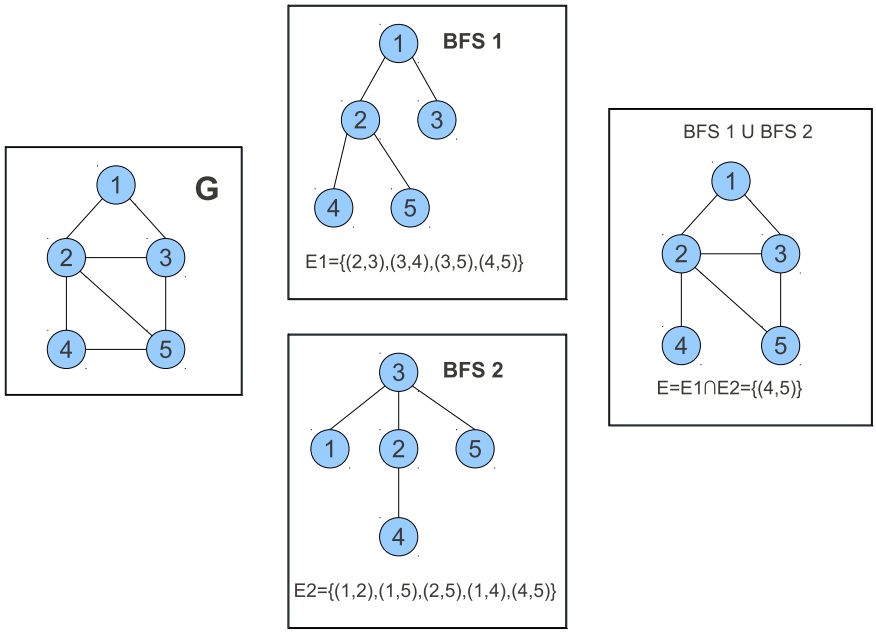

Figure 5.2: Left: the underlying graph G; Center: two BFS trees rooted at node 1 and node 3; $E1$ and $E2$ are corresponding sets of allowed positions for each BFS tree; Right: a graph merged from two BFS trees, while the allowed positions are the intersection of those of two BFS trees.

BFS trees to cover the entire graph, while a Poisson 10 graph need about 50 BFS trees. Second, power-law graphs need more BFS trees to cover than Poisson ones. As power-law 2.1 graph, although 100 BFS trees (one tenth of total nodes) are used, some links are still uncovered. Third, random-root strategy and maxdegree-root strategy have little impact with Poisson model graphs; while for power-law graphs, in case of small number of roots, maxdegree-root strategy has always a better result than random-root strategy.

5.4 Experiments

In this section, we present experiments on various graph types including real-world cases with various number of BFS trees.

5.4.1 Random model graphs

In Table 5.2, we show the estimation of m we obtain using several BFS trees of random model graphs. Each graph has 10000 nodes and the estimated number of links are averaged over 10 samplings. The first column is the type and the

Table 5.1: Discovery probability of links in several scenarios: Poisson 3, 5 and 10 and power-law 2.1, 2.2, 2.3 graphs, each with the random and maxdegree root selection strategies, and with 1, 2, 5, 10, 20, 50 and 100 BFS trees.

Graph	root choice	number of BFS trees						
		1	2	5	10	20	50	100
Poisson 3	Random	0.6397	0.8433	0.9852	0.9994	1	1	1
	Maxdegree	0.6397	0.8472	0.9870	0.9997	1	1	1
Poisson 5	Random	0.4006	0.6283	0.9069	0.9904	0.9998	1	1
	Maxdegree	0.4006	0.6291	0.9075	0.9893	0.9999	1	1
Poisson 10	Random	0.1994	0.3590	0.6696	0.8880	0.9860	1	1
	Maxdegree	0.1994	0.3583	0.6670	0.8853	0.9870	1	1
Power 2.1	Random	0.3757	0.4494	0.6386	0.7676	0.8541	0.9413	0.9772
	Maxdegree	0.3757	0.4776	0.6415	0.7765	0.9013	0.9928	0.9997
Power 2.2	Random	0.5316	0.6216	0.7353	0.8334	0.9238	0.9833	0.9979
	Maxdegree	0.5316	0.6344	0.7925	0.9002	0.9716	0.9989	0.9999
Power 2.3	Random	0.5525	0.6526	0.7887	0.8648	0.9320	0.9849	0.9976
	Maxdegree	0.5525	0.6537	0.8021	0.9044	0.9697	0.9991	0.9996

parameter of the considered graphs. The second column is the mode of selection of the root(s). The third column is the actual number of links in the considered graph. The fourth, fifth, sixth, seventh columns represent the estimated number m' of links with respectively 1[1], 2, 5, 10 roots. The last column shows the strategy that shows the best result. Notice that each estimation (the content of each cell in Table 5.2) is the result of the whole execution of Algorithm 3 and corresponds to the minimum KS value. We choose the steps $\Delta\lambda = 0.1$ for Poisson and $\Delta\alpha = 0.01$ for power-law, which is reasonable concerning time consuming, and moreover, as in the case of a single BFS, the results are not improved by diminishing the step.

First, we observe that our strategy can distinguish well the type of graphs. For all Poisson model graphs, the best estimation is given by RR strategy. On the contrary, for those power-law graphs, PP strategy shows a better estimation.

Also, in most cases, the selection strategy for roots of BFS is not an important factor in our methodology.

However we can see that with 2 BFS trees the results are poor, sometimes even worse than with a single BFS tree, which is suspecting and counter-intuitive.

In general, however the more BFS trees we use, the more precise is the result we observe. In case of Poisson model graphs, single BFS strategy always overestimates the number of links; while in case of power-law model graphs, single BFS underestimates the number of links. When we use about 5 or 10 BFS trees, the

[1]This is exactly the case that was treated in Section 4.3

Table 5.2: Results with several BFS trees on random model graphs

n=10000	root choice	m	number of BFS tree				Best strategy
			1	2	5	10	
Poisson 3	Random	15728	25500	21500	15000	15000	RR
Poisson 3	Maxdegree	15728	23000	20050	18000	15000	RR
Poisson 5	Random	25379	35000	30000	27500	26000	RR
Poisson 5	Maxdegree	25379	38000	30500	27500	26000	RR
Poisson 10	Random	50010	69000	71000	63000	57000	RR
Poisson 10	Maxdegree	50010	66000	69000	60000	54500	RR
Power 2.1	Random	29358	14365	9662	16099	21163	PP
Power 2.1	Maxdegree	29358	14365	12991	18311	21163	PP
Power 2.2	Random	20719	12990	11890	18311	18311	PP
Power 2.2	Maxdegree	20719	11890	11890	18311	18311	PP
Power 2.3	Random	21679	14365	14365	16099	16099	PP
Power 2.3	Maxdegree	21679	14365	14365	16099	18311	PP

estimation of the number of links is much better.

5.4.2 Real-world graphs

We present in Table 5.3 the results for typical real-world graphs.

We only display results when roots of BFS trees are chosen at random. Indeed, results with the other strategy are very similar. We conduct our experiments with 1, 2, 5 and 10 roots and all results are averaged over ten samplings.

Table 5.3: Results with several BFS on real-world graphs.

	n	m	BFS number				Best strategy
			1	2	5	10	
Radar-cm	21185	15728	25500	21500	15000	15000	RR
Radar-enix	30433	73576	68004	89642	75731	74186	RR
Radar-japon	26698	77545	60899	67665	73078	79845	RR
Radar-ortolan	24262	48516	50686	54395	49450	49450	RR
Skitter-AS	5775	12025	8297	8294	10576	13368	PP

For Radar graphs, the minimum values of KS are always given by RR strategy, in other words the estimated type is more likely Poisson. Although the results with 2 BFS is sometimes worse than those with only one BFS, the results with 5 and 10 BFS lead to much better estimates.

The results for Skitter-AS are similar, but this time with a power-law type.

5.5 Conclusion

In this chapter, we studied how the fact that we may obtain measurement from several monitors, modeled by several BFS trees, improves our results. This is particularly relevant for our estimate of the number of links m, which was poor in the previous chapter.

To explore this, we designed a multiple BFS methodology, $MB(RR, PP)$ and conducted experiments on a wide variety of cases. The obtained results are very good and clearly show that even a small number of combined BFS trees is sufficient to accurately estimate m, while still being able to correctly decide on the type of the degree distribution of the considered graph. This is due not only to the fact that several BFS uncovered a larger proportion of all links, but also to the fact that they identify many forbidden positions for invisible links, which plays a key role in our method.

Chapter 6

Analysis of the profile of BFS trees

Contents

6.1 Introduction

In previous sections, we designed several methodologies for deciding on the type of a graph from its BFS tree(s): we distinguish the type (for example Poisson versus power-law) of the degree distribution and estimate the number of links of an unknown graph G from the knowledge of one of its BFS tree T (or several BFS trees). These methodologies can be significantly improved when some supplementary informations concerning the profile of the BFS tree is given. In this chapter we show that the knowledge of the number of nodes at each level of the BFS tree, and the number of missing (invisible) links between two consecutive levels of the BFS tree, allows a reconstruction process which leads to a graph with a more similar topology to the original one.

As mentioned in Chapter 2, we consider BFS trees of random graphs described a given node degree sequence. A node degree sequence is defined as an ordered j-tuple: $(d_1, d_2..., d_j)$, which means that there are d_j nodes with node degree j; with a degree sequence, the number of nodes is $n = \sum_j d_j$ and the number of links is $\frac{1}{2}\sum_j jd_j$. We use the configuration model [Bol01] to construct a random (multi)graph with a given node degree sequence: for each node of degree j, we create j copies, and then define the links of the graph according to a uniformly random matching on these copies. Recently Achlioptas, Kempe, Clauset and Moore [AKCM05] proposed a random BFS process on a given node degree sequence in which configuration model [Bol01] is used as the generator of graphs. The profile of such BFS is a random structure concerning all possible BFS trees from a set of graphs generated by a degree sequence.

The profile of BFS trees involves to several aspects: (1) the node degree distribution; (2) the number of nodes at each level $\{n_k\}$; (3) the number of invisible links at each level $\{e_k\}$. The node degree distribution has been proved both mathematically and experimentally [AKCM05, CGW07] to be biased (it is always a power-law) with respect to that of the underlying graphs or the underlying sequence that is used to generate the graphs. In this chapter, we particularly focus on the second and the third problem and prove two theoretical properties (Theorem 8 concerning the number of nodes at each level and Theorem 9 concerning the number of invisible links at each level), that can further be used in experiments to improve the deciding methodology in the next chapter (Chapter 7).

This chapter is organized as follows. We first present in Section 6.2 the BFS algorithm on configuration model. In Section 6.3 we evaluate some quantities regarding nodes and copies, which will be used in the following sections. In Section 6.4 we describe the schema that we shall follow to carry out the analysis of the profile: from the behavior of the queue to the urn problem. We state our key results in Section 6.5. Section 6.6 and Section 6.7 give the proofs of our theorems.

6.2 BFS on configuration model

Given a graph G, a BFS tree of G is built one node at a time as follows. At each step, every node in G is in *explored, untouched or pending* state. The process is initialized by labeling the root node as pending, and all other nodes as untouched. Then, the process consists in exploring one pending node at a time, until there is no pending node remaining. Exploring a pending node consists in turning all its untouched neighbors to pending state and then turning itself to explored state. Therefore a node is *explored* if both it and its neighbors are in the tree; *untouched* if it is not yet reached by the process; and *pending* if it is on the boundary of the tree. Links between the pending nodes and its untouched neighbors are called *visible* links. This leads to a tree defined by the set of all visible links during the process. Other links are said to be *invisible*. If pending nodes are considered in a first-in-first-out (FIFO) order, then the obtained result is a BFS tree.

In [AKCM05], the authors show that this is equivalent to the following. First suppose that we have as many *copies* of each node as its degree, and let us call two copies of the same node *siblings*. Therefore, there are $m = \sum_j d_j = \sum_j a_j n$ copies, where $\{a_j\}$ denotes the degree distribution and n the number of nodes (m is twice the number of links). Now let us label each copy with an integer chosen uniformly at random in $[1, m]$ in a way such that no two copies have the same label. We initialize a FIFO queue Q with the copy with maximal label, which we put in *enqueued* state, all other copies being *untouched* state initially. A BFS is then constructed by iterating the following process [AKCM05]. At each step, we consider u the *unexposed* (that means either enqueued or untouched.) copy with largest label. Then we match it with the first element v of the queue (which we thus remove from Q) and turn u to *exposed* state. If v is untouched then (u, v) belongs to the BFS tree and we add v's siblings to Q. This process naturally ends after m steps, as each copy is then exposed. See Algorithm 4 and the example below.

Example 4. *Figure 6.1 illustrates the process step by step. In this example, a degree sequence* $(0, 1, 0, 1, 2)$ *is given (two nodes with degree* 4*, one node with degree* 3 *and one with degree* 1*). This leads to* $2 \times 4 + 1 \times 3 + 1 \times 1 = 12$ *copies, each randomly tagged with an index between* 1 *and* 12*. In this example, copy tagged by* 4 *is chosen as the root, then we have:*

- *Initialization: Copy* 4 *is chosen as the root and put in the queue* Q*.*
- *Step* $t = 12$*: At time* 12*, match copy* 4 *(at the head of* Q*) and copy* 12 *(with index equal to* t*); as copy* 12 *is untouched, reveal the visible link* $4 - 12$ *and put the siblings of* 12 *(*i.e. 9, 7, 3*) into* Q*.*
- *Step* $t = 11$*: At time* 11*, match copy* 9 *(at the head of* Q*) and copy* 11 *(with index equal to* t*); as copy* 11 *is untouched, reveal the visible link* $9 - 11$ *and put the siblings of* 11 *(*i.e. 1, 2, 6*) into* Q*.*

Data: A degree sequence $\{d_j\}$
Result: A sampling BFS tree T, corresponding to a graph G with degree sequence $\{d_j\}$.

For each node of degree j, create j copies; $m \leftarrow \sum j d_j$;
Index the copies with a random permutation of $[1..m]$;
Set all copies *untouched* ;
Randomly choose a copy r, add it at the head of an empty queue Q;
Append r's siblings to Q and set all copies in Q *enqueued*;
for $i \leftarrow m$ *to 1* **do**
    Let v be the copy with index i;
    if v *is not exposed* **then**
        Let u be the copy popped from the head of Q;
        if v *is untouched* **then**
            Add link (u, v) to T ;
            (Add link (u, v) to G);
            Append v's siblings to Q ; set them *enqueued* ;
        else
            %v is *enqueued*% ;
            (Add link (u, v) to G);
            Remove v from Q;
        end
        Set u and v *exposed*;
    end
end

Algorithm 4: Constructing a BFS of a model graph.

- *Step* $t = 10$*: At time* 10*, match copy* 3 *(at the head of* Q*) and copy* 10 *(with index equal to* t*); as copy* 10 *is untouched, reveal the visible link* $3 - 10$ *and put the siblings of* 10 *(*i.e. 8, 5*) into* Q*.*

- *Step* $t = 8$*: At time* 8*, match copy* 7 *(at the head of* Q*) and copy* 8 *(with index equal to* t*); as copy* 8 *is touched, reveal the invisible link* $7 - 8$*.*

- *Step* $t = 6$*: At time* 6*, match copy* 1 *(at the head of* Q*) and copy* 6 *(with index equal to* t*); as copy* 6 *is touched, reveal the invisible link* $1 - 6$*.*

- *Step* $t = 5$*: At time* 5*, match copy* 2 *(at the head of* Q*) and copy* 5 *(with index equal to* t*); as copy* 5 *is touched, reveal the invisible link* $2 - 5$*.*

Here, we do not show $t = 9, 7, 4, 3, 2, 1$*, because at these times the corresponding copies (ex. at time* $t = 9$*, the copy with index* 9*) had been exposed.*

Two kinds of links either visible (both in the graph and in the BFS tree) or invisible (only in the graph), are revealed. The invisible links are those who are get rid off during the process of BFS. They come from the event that the copy at the head of Q *have a partner who has been touched, also in the* Q*. Obviously, the duplicated links (such as link between* 6 *and* 5*) and the self-loop links (such as link between* 12 *and* 7*) are certainly invisible ones in a BFS tree.*

6.3 Evolution of nodes, copies and POP operation

As described in the last section, the sampling BFS trees may be different from each other according to the permutation generated at random. So the profile that we mention here is concerned to all permutations of m elements, namely $m!$ possibilities for m copies, then $m!$ BFS trees. Using Algorithm 4, we can trace the number of copies (or nodes), either untouched, enqueued or exposed. First, we introduce a useful concept.

Definition 6. Maximum index*: the maximum index of a node is the maximum index of all its copies' indices.*

At any time i, the untouched nodes are precisely those whose maximum index is less than i, and the explored or pending nodes (whose copies are exposed or enqueued) are those whose maximum index is greater than i. This observation allows us to carry out an explicit analysis as a function of time. Then we have the following properties:

Property 1. *[AKCM05] At time* i*, the indices of the unexposed copies, both inside and outside the queue, are random in* $[1, i]$*.*

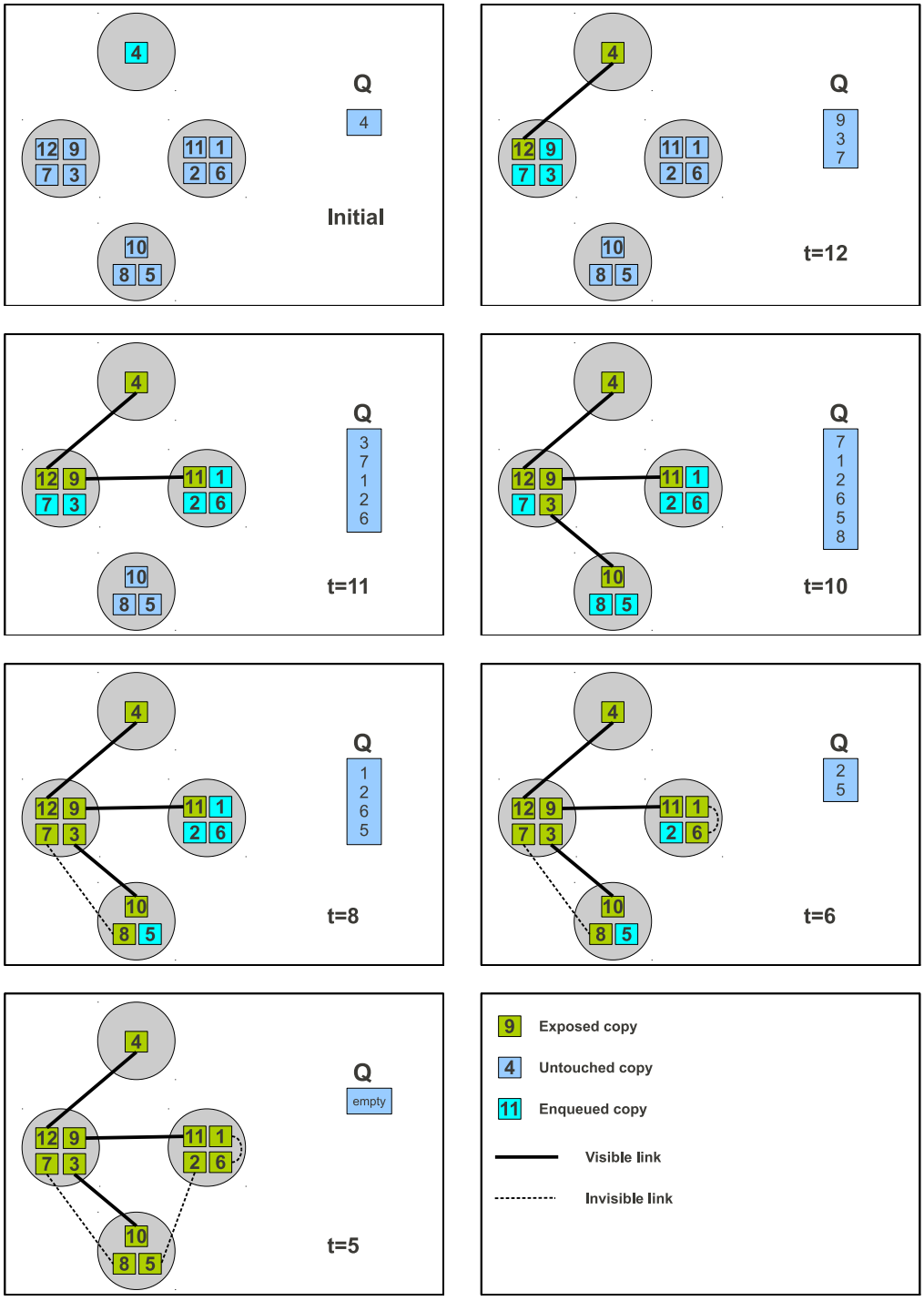

Figure 6.1: An example of BFS from degree sequence $(0, 1, 0, 1, 2)$, *i.e.* two nodes of degree 4, one of degree 3 and one of degree 1.

Proof. At a time, we match the copy at the head of queue Q with the copy with the same label as the corresponding time, so all copies that have an index i' greater than i must be exposed by two ways:

- At time i'' with $i'' > i' > i$ (recall that the time is decreasing from m to 1), copy i' is at the head of queue Q. Copy i' and copy i'' are matched.
- At time i' with $i' > i$, another copy at the head of queue Q matches copy i'.

□

6.3.1 Properties of nodes

The distribution of the number of untouched nodes can be specialized by the following property:

Property 2. *[AKCM05] Let $N_{unto,j}(i)$ denote the random variable of the number of untouched nodes of degree j at time i, then the expectation and the variance of $N_{unto,j}(i)$ with n nodes and a given degree distribution $\{a_j\}$ are:*

$$E[N_{unto,j}(i)] = a_j \left(\frac{i}{m}\right)^j n \tag{6.1}$$

$$V[N_{unto,j}(i)] = a_j \left(\frac{i}{m}\right)^j \left(1 - \left(\frac{i}{m}\right)^j\right) n \tag{6.2}$$

Proof. The probability that a node of degree j has maximum index less than i is exactly $\left(\frac{i}{m}\right)^j$ and there are $a_j n$ nodes with degree j, so the number of untouched nodes follows binomial distribution $B(a_j n, \left(\frac{i}{m}\right)^j)$. □

Property 3. *[AKCM05] Let $N_{unto}(i)$ denote the random variable of the total number of untouched nodes at time i. Then the expectation and the variance of $N_{unto}(i)$ with n nodes and a given degree distribution $\{a_j\}$ are:*

$$E[N_{unto}(i)] = \sum_j a_j \left(\frac{i}{m}\right)^j n \tag{6.3}$$

$$V[N_{unto}(i)] = \sum_j a_j \left(\frac{i}{m}\right)^j \left(1 - \left(\frac{i}{m}\right)^j\right) n \tag{6.4}$$

Proof. The binomial distributions of Property 2 are independent of each other, so we can add the expectation and variance. □

As the extension of the Property 3, the next property gives an asymptotic distribution of the number of untouched nodes at time i.

Property 4. *Let $N_{unto}(i)$ denote the random variable of the total number of untouched nodes at time i. Then it is distributed normally with mean and variance as follows*[1]*,*

$$N_{unto}(i) \sim \mathcal{N}\left(\sum_j a_j \left(\frac{i}{m}\right)^j n, \sum_j a_j \left(\frac{i}{m}\right)^j \left(1 - \left(\frac{i}{m}\right)^j\right) n\right) \tag{6.5}$$

Proof. If a suitable continuity correction is used when n is large enough, then an excellent approximation to $B(n,p)$ is given by the normal distribution $\mathcal{N}(np, np(1-p))$[2]. The approximation generally improves as n increases. As we know that the sum of two independent normally distributed random variables is normal, with its mean being the sum of the two means, and its variance being the sum of the two variances. □

6.3.2 Properties of copies

With the properties of nodes, the corresponding properties of copies are directly developed.

Property 5. *[AKCM05] Let $C_{unto}(i)$ denote the random variable of the number of untouched copies at time i, then the expectation of $C_{unto}(i)$ with n nodes and a given degree distribution $\{a_j\}$ is:*

$$E[C_{unto}(i)] = \sum_j j a_j \left(\frac{i}{m}\right)^j n \tag{6.6}$$

Proof. Notice that $C_{unto}(i) = \sum_j j N_{unto,j}(i)$. □

Property 6. *[AKCM05] Let $C_{unex}(i)$ denote the random variable of the number of unexposed copies at time i, then the expectation and the variance of $C_{unex}(i)$ are:*

$$E[C_{unex}(i)] = \delta \left(\frac{i}{m}\right)^2 n = \left(\frac{i}{m}\right)^2 m = \frac{i^2}{m} \tag{6.7}$$

$$V[C_{unex}(i)] = \delta \left(\frac{i}{m}\right)^2 \left(1 - \left(\frac{i}{m}\right)^2\right) n = \left(\frac{i}{m}\right)^2 \left(1 - \left(\frac{i}{m}\right)^2\right) m \tag{6.8}$$

where n is the number of nodes and $\delta = \frac{m}{n}$ (recall $m = 2|E|$) is the average degree.

Proof. To calculate $E[C_{unex}(i)]$, recall that the copy at the head of the queue has a uniformly random index conditioned on being less than i. Therefore, the process forms a matching on the list of indices as follows: take the indices in decreasing

[1]The notation $X \sim \mathcal{N}(\mu, \sigma^2)$ means the random variable X is distributed normally with mean μ and variance σ.

[2]binomial distribution, see `http://en.wikipedia.org/wiki/Binomial_distribution`

order from m to 1, and at time i match the index i with a randomly chosen index less than i. This creates a uniformly random matching on the δn indices. Now, note that a given index is still remaining at time i if both it and its partner are less than i, and since the indices are uniformly random in $[1, m]$ the probability of this is $\left(\frac{i}{m}\right)^2$. So the probability follows the binomial distribution $B(m, \left(\frac{i}{m}\right)^2)$. □

With the random variables $C_{unex}(i)$ and $C_{unto}(i)$, the random variable of the number of copies enqueued $C_{enqu}(i)$ can be calculated directly:

Property 7. *Let $C_{enqu}(i)$ denote the random variable of the number of enqueued copies at time i, then the expectation of $C_{enqu}(i)$ with n nodes and a given degree distribution $\{a_j\}$ is:*

$$E[C_{unto}(i)] = \delta \left(\frac{i}{m}\right)^2 n - \sum_j j a_j \left(\frac{i}{m}\right)^j n \tag{6.9}$$

Proof. From the relation $C_{enqu}(i) = C_{unex}(i) - C_{unto}(i)$. □

All these properties can be specified with a given distribution.

Example 5. *In Table 6.1, we give the expectation of $C_{unex}(i)$ and $C_{unto}(i)$ for some graphs with typical types.*

Table 6.1: $E[C_{unex}(i)]$ and $E[C_{unto}(i)]$ for regular, Poisson and power-law graphs.

Distribution	a_j	$E[C_{unex}(i)]$	$E[C_{unto}(i)]$
Regular r	$a_r = 1$	$r\left(\frac{i}{m}\right)^2 n$	$r\left(\frac{i}{m}\right)^r n$
Poisson λ	$a_j = \frac{e^\lambda \lambda^j}{j!}$	$\lambda\left(\frac{i}{m}\right)^2 n$	$ie^{\lambda\left(\frac{i}{m}-1\right)}$
Power-law α	$a_j = \frac{1}{\sum j^{1-\alpha}} j^{-\alpha}$	$\frac{\sum i^{1-\alpha}}{\sum i^{-\alpha}}\left(\frac{i}{m}\right)^2 n$	$\frac{\sum j^{1-\alpha}\left(\frac{i}{m}\right)^j}{\sum j^{-\alpha}} n$

6.3.3 Properties of POP operation

We now turn to the expectation of the number of POP operations. In Algorithm 4, at time i, the copy u at the head of queue Q is popped only when the copy v with index i is in unexposed state. There is no POP operation if the copy with index i has been exposed before time i (that is to say, this copy has arrived at the head of Q before time i, such as for $t = 9, 7, 4, 3, 2, 1$ in Figure 6.1).

Property 8. *Let $P_{POP}(i)$ denote the number of POP operation (either 1 or 0) at time i, then the expectation of $P_{POP}(i)$:*

$$E\left(P_{POP}(i)\right) = \frac{2i-1}{2m} \tag{6.10}$$

Proof. After a POP operation , two copies are exposed, so the probability of a POP satisfies:

$$E(C_{unex}(i-1)) = (1 - E(P_{POP}(i)))\, E(C_{unex}(i)) + E(P_{POP}(i))\, (E(C_{unex}(i)) - 2) \tag{6.11}$$

And the expectation of $P_{POP}(i)$ can be directly calculated from Equation (6.7) and Equation (6.11). □

Finally, the expected number of POP operations during a sequence of steps is easily computed.

Property 9. *Let $X_{POP}(i_k, i_{k+1})$ denote the number of POP operations between time i_k and i_{k+1} ($i_k > i_{k+1}$), then the expectation of $X_{POP}(i_k, i_{k+1})$ is:*

$$E(X_{POP}(i_k, i_{k+1})) = \sum_{i=i_k}^{i_{k+1}} E\left(P_{POP}(i)\right) = \frac{i_k^2 - (i_{k+1} - 1)^2}{2m} \tag{6.12}$$

6.4 Schema of our analysis: scanning the queue

This section is contributed to how to rephrase the process of BFS in form of an urn model.

6.4.1 Critic timing i_k

Our analysis of the BFS profile relies on the study of the FIFO queue during the BFS process, by exhibiting the relationship between the construction of the successive levels and the evolution of the queue.

We are interested in the moment i_k when the exploration of level k of the BFS begins: during the time between i_k and i_{k+1} the nodes at level k are exposed, (and also the invisible links between levels k and $k+1$ are "revealed"). Time i_1 represents the moment when the root of the BFS is exposed and time i_{k+1} represents the moment when all copies that in Q at time i_k are finally in the state exposed. For example, in Figure 6.1, $i_1 = 12$, $i_2 = 8$ and $i_3 = 5$.

At time i_k, there are $C_{enqu}(i_k)$ copies in the queue Q. When these copies are exposed, the construction of level k is done and the BFS process begins to explore the next level. Since a POP operation always corresponds to the head of the queue (and its partner) being exposed, the number of copies at level k is equal to the number of POP operation between time i_k and i_{k+1}.

In Equation (6.12), the number of POP operation between i_k and i_{k+1} is expressed as a simple function of i_k and i_{k+1}. Thus if we can evaluate the number of POP, we can then give an iterative expression of i_{k+1} in function of i_k.

The evaluation of the number of POP between i_k and i_{k+1} can be rephrased in terms of urn model. Let us denote by X the set of enqueued copies at time i_k and denote by Y the set of untouched copies at time i_k.

Property 6 shows that at time i the probability that a copy has been exposed is $\left(\frac{i}{m}\right)^2$, which is independent to the degree of node that is attached. Therefore all copies have a uniform probability to present at the head of Q. If we colorize all enqueued copies in Q as white balls and all untouched copies as black balls, the problem of the BFS process may be reduced to a corresponding urn model problem.

6.4.2 Basic concept of urn models

To familiarize this kind of problem, we introduce first some basic concepts of urn models. In the context of this thesis, only the case of two colors is considered. At the beginning, there are a_0 white and b_0 black balls in the urn. At each step, a ball is chosen (not take it out of the urn) at random from the urn, then we examine its color and add or remove balls according to the color. Generally speaking, if the tested ball is white, then a white balls and b black balls are put into (or remove from) the urn, while if the tested ball is black, then c white balls and d black balls are put into (or remove from) the urn. The value $a, b, c, d \in Z$ are fixed integer (could be negative) and the urn is specified by the transition matrix $M = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$. We are also allowed on occasion to describe M linearly as $(a, b; c, d)$. With the help of the expression of matrix, urn models with r types of colors can be described in an analogous way by using an $r \times r$ transition matrix.

The urn model is said to be *balanced* if $a+b = c+d$, in which case the common sum s of the matrix rows is the balance. After n step, there will be $n \times s$ balls added. If the sum s is negative, the urn is a *diminishing* model.

For urns with negative diagonal entries, as $a \leq -2$, we have to ensure that the process of removing balls from the urn would not be blocked. To do that, the initial configuration of the urn must satisfy $a|a_0$ and similarly for black balls. We call this restriction the condition *tenable*. For the urn involving substraction for both a and d in $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$, such as $a = -1$ and $d = -2$, the condition of tenability is broken. To deal with this kind of diminishing urn, we use the halt condition $a_k <= min(a, d)$ instead of $a_k = 0$.

In the reference book of Johnson and Kotz (1977) [JK77], many kinds of urn model have been described. In recent years, researchers developed the problem of urn models and some general methodologies have been established. [FGP05] describes a purely analytic approach to urn models of the generalized or extended Pólya-Eggenberger type, in the case of two types of balls and constant "balance"; In [FDP06], a fundamental isomorphism between discrete-time balanced urn processes and certain ordinary differential systems, which are nonlinear, autonomous, and of a simple monomial form. As a consequence, all balanced urn processes with balls of two colors are proved to be analytically solvable in finite terms. In [HKP07], several exactly solvable urn models with a diminishing character are

studied.

6.4.3 *POP* problem

In the BFS construction, at each step when a *POP* operation occurs, the copy u at the head of the queue Q matches another copy v which is either in X or in Y, and then we throw out both u and v. If v belongs to X, two copies are thrown out of X, whereas if v belongs to Y, one copy is thrown out from X and the other one from Y.

First we describe *POP* **rule**: this is a variant of the diminishing urns with transition matrix $M = \begin{pmatrix} -2 & 0 \\ -1 & -1 \end{pmatrix}$. *POP* rule consists of two steps:

- choose randomly a white ball and throw it off.
- if there is still white ball, choose randomly a ball (white or black) among all balls in the urn and throw it off.

Notice that the halt condition is that there is less than or equal to one white ball instead of no white ball in order to avoid abnormality after having thrown the first white ball.

The *POP* problem is described as follows:

Definition 7. *POP* **problem***: The problem of the number of POP operation, when we apply iteratively POP rule, either two copies of X are removed, or one copy of X and one copy of Y are removed. The initial configuration is $p = |X|$ enqueued copies (as white balls) and $q = |Y|$ untouched copies (as black balls).*

In the form of diminishing urn model, our problem is described as follows: In our work, we are interested in the number of *POP* while X becomes empty. The *POP* problem can be modeled as an equivalent diminishing urn model. It is similar to the lower triangle case $M = \begin{pmatrix} -a & 0 \\ -c & -d \end{pmatrix}$ in [HKP07] (In our case, the denominator is $p - 1$ instead of p). The problem is a variant of the diminishing urns with the transition matrix: $M = (-2, 0; -1, -1)$. But the general solution is too complex to extract the coefficient.

6.5 Statement of our results

In this section, we list the main results of our work. It relies on the configuration model which we have described in Section 6.2. In order to analyze more rigorously, we use the generating functions[3].

[3] A generating function is a formal power series in one indeterminate, whose coefficients encode information about a sequence of numbers a_n that is indexed by the natural numbers. Generating

Data: An initial configuration (p, q), p white balls and q black balls.
Result: The number of step k.

```
k ← 0;
while p > 1 do
    p ← p − 1: choose a white ball and remove it;
    choose a ball at random and check the color;
    if white ball then
        p ← p − 1: remove the white ball ;
    else
        q ← q − 1: remove the black ball ;
    end
    k ← k + 1 ;
end
Return k ;
```

Algorithm 5: The diminishing urn model equivalent to *POP* problem.

Theorem 6. *Let $h_{p,q,k}$ denote the probability that for a urn with p white balls and q black balls, after k steps there is no white ball in the urn using POP rule described. The corresponding generating function is denoted $H(x,y,z) = \sum_{p,q,k} h_{p,q,k} x^p y^q z^k$. Then the $h_{p,q,k} = \left[x^p y^q z^k\right] H(x,y,z)$ is the extracted coefficient of H, where $H(x,y,z) = H1(x,y,z) + H2(x,y,z) + H3(x,y,z)$ and $H1$, $H2$, $H3$ are*

$$H1\,(x,y,z) = \int_0^1 \frac{x^4 w z^2\,(xw+1)}{yF^2\,(1-x^2w^2z)^{\frac{1}{2}}\,(1-x^2z)^{\frac{3}{2}}} dw$$

$$H2\,(x,y,z) = \int_0^1 \frac{x^4 w z\,(1+2xwz)\,(1-x^2w^2z)^{\frac{1}{2}}}{y\,(1-x^2z)^{\frac{3}{2}}\,F\,(F-xw)} dw$$

$$H3\,(x,y,z) = \int_0^1 \frac{x^5 w^2 z^2\,(1-x^2w^2z)^{\frac{1}{2}}\,(2F-xw)}{y\,(1-x^2z)^{\frac{3}{2}}\,F^2\,(F-xw)^2} dw$$

where $F = \left(1 + \left(\frac{x}{y} - 1\right)\left(\frac{1-x^2w^2z}{1-x^2z}\right)^{\frac{1}{2}}\right)$.

In Subsection 6.6.3, we simulate the behavior of the queue by a decreasing urn model. *POP* problem is modeled as a decreasing urn model with transition matrix $(-2, 0; -1, -1)$. The next Theorem 6 gives the explicit solution of Partial Differential Equation (PDE) of the corresponding system.

functions are often expressed in closed form (rather than as a series), by some expression involving operations defined for formal power series. These expressions in terms of the indeterminate x may involve arithmetic operations, differentiation with respect to x and composition with (i.e., substitution into) other generating functions; since these operations are also defined for functions, the result looks like a function of x. See also details in [FS09, S.W93].

In Subsection 6.6.4, we focus on the problem how to extract the exact coefficient of the solution H. The explicit exact coefficients of H is so complicated that we cannot give an intuitive explanation. The next Theorem gives an expression of $\sum_k kh_{p,q,k}$, namely the expected number of POP operation.

Theorem 7. *Let $h_{p,q,k}$ denote the probability that for a urn with p white balls and q black balls, after k steps there is no white ball in the urn using POP rule. Then the expected number of POP is*

$$E_{p,q}(k) = \sum_k kh_{p,q,k} = p - \frac{p^2}{2\,(p+q)} + O\left(\frac{p}{\sqrt{q}\sqrt{p+q}}\right) \tag{6.13}$$

The analysis of the POP problem helps us to develop the relation between the number of nodes of the consecutive levels, as described in the following theorem and the proof and the examples will be given in Subsection 6.6.5.

Using Theorem 7, two main results of the profile of a BFS tree are inferred. See detail in Section 6.7. The first one concerns the expected number of nodes at level k in T.

Theorem 8. *Let n_k denote the number of nodes of level k in the BFS tree, respectively, n_{k-1}, n_{k-2},..., n_1 the number of nodes of level $k-1$, $k-2$,... , 1 and a degree distribution $\{a_j\}$ in form of generating function $g(z) = \sum_{j=1} a_j z^j$, then the expected number of nodes $E(n_{k+1})$ of level $k+1$ in the BFS tree is:*

$$E\,(n_{k+1}) = n - \sum_{i=0}^{k} n_i - ng\left(\frac{g'\left(g^{-1}\left(1 - \sum_{i=0}^{k} \frac{n_i}{n}\right)\right)}{g'\,(1)}\right) \tag{6.14}$$

Applying Theorem 8, we can compute the expected number of nodes at level $k+1$ from the numbers of nodes of the first k levels. In practice, we replace n_k in Equation (6.14) with $E\,(n_k)$. Then n_1, n_2...n_k can be iteratively computed.

The second one concerns the expected number of invisible links at level k (which are useful for the strategy of reconstruction of a graph from a BFS).

Theorem 9. *Let e_k denote the number of invisible links between level k and level $k+1$ in the BFS tree, and a degree distribution $\{a_j\}$ in form of generating function $g(z) = \sum_{j=1} a_j z^j$, then the expectation of $E(e_k)$ is:*

$$E\,(e_k) = \frac{i_k^2 - n^2 g'\left(\frac{i_k}{m}\right)^2}{2m} - m\left(g\left(\frac{i_k}{m}\right) - g\left(\frac{g'\left(\frac{i_k}{m}\right)}{g'\,(1)}\right)\right) \tag{6.15}$$

The term i_k is decided by expression $i_{k+1} \sim ng'\left(\frac{i_k}{m}\right)$ (See detail in Section 6.7).

The rest of this chapter is contributed to the proof of the theorems listed in this section.

6.6 Analysis of POP problem

The queue Q make a crucial role in the process of Algorithm 4. In this section we analyze the behavior of the Q, particularly on the problem: number of POP operation. In Subsection 6.6.1, we describe the relevance between our model and the urn model. Subsection 6.6.2 and Subsection 6.6.3 explain how to get the corresponding PDE and the solution from our POP problem. Subsection 6.6.4 is contributed to the extraction of coefficients. Subsection 6.6.5 gives an explicit expression of the expected number of POP problem.

6.6.1 Description of POP problem

After having thrown a white ball (after step 1 of POP rule) out, there are still $p+q-1$ balls in the urn. So the probability that a white ball is selected as the second ball is $\frac{p-1}{p+q-1}$, while the probability of a black ball is $\frac{q}{p+q-1}$.

We denote $h_{p,q,k}$ the probability that with exactly k steps and an initial configuration (p,q), there is no longer white ball. So we have the recursive relation of $h_{p,q,k}$:

$$h_{p,q,k} = \frac{q}{p+q-1} h_{p-1,q-1,k-1} + \frac{p-1}{p+q-1} h_{p-2,q,k-1} \qquad p>1,\ q>0,\ k>0 \quad (6.16)$$

The Equation (6.16) can be expressed as

$$(p+q-1)\, h_{p,q,k} = q h_{p-1,q-1,k-1} + (p-1)\, h_{p-2,q,k-1} \quad (6.17)$$

The initial condition of $h_{p,q,k}$ is: $h_{0,q,0}=1$ (when there is no white ball, 0 POP operation will take place) and $h_{1,q,1}=1$ (when there is only one white ball in the urn, exactly one POP operation will take place).

Then we rewrite $h_{p,q,k}$ in form of generating function:

$$H(x,y,z) = \sum_{p\geq 2} \sum_{q\geq 1} \sum_{k\geq 1} h_{p,q,k} x^p y^q z^k \quad (6.18)$$

With help of the generating function, the recurrence Equation (6.17) can be translated into a first order linear partial differential equation (PDE).

The proof of Theorem 6 consists of two parts: (1) get the expression of a partial differential equation (in Subsection 6.6.2); (2) resolve the corresponding PDE (in Subsection 6.6.3).

6.6.2 Expression of PDE

Lemma 4. *Using POP rule, the corresponding PDE is:*

$$\left(x - x^3 z\right) H_x + \left(y - xy^2\right) H_y - \left(1 + x^2 z + xyz\right) H = F(x,y,z)$$

$$F(x,y,z) = \frac{x^2 y z^2 (x+1)}{1-x^2 z} + x^2 y^2 z^2 \left(\frac{1}{(1-y)^2} + \frac{1}{1-y}\right) + \left(x^2 z + 2x^3 z^2\right) \frac{y}{1-y} \quad (6.19)$$

In fact, the reduced PDE $(x - x^3z)\,H_x + (y - xy^2)\,H_y - (1 + x^2z + xyz)\,H = 0$ of Equation (6.19) is decided by the transition matrix $\begin{pmatrix} -2 & 0 \\ -1 & -1 \end{pmatrix}$ and the $F(x, y, z)$ is only impacted by the initial conditions $h_{0,q,0} = 1$ and $h_{1,q,1} = 1$.

Proof. **Left term of Equation (6.17)**

First, we conduct the addition on the term $(p + q - 1)\,h_{p,q,k}$ according to Equation (6.18),

$$\begin{aligned} A &= \sum_{q\geq 1}\sum_{p\geq 2}\sum_{k\geq 1}(p+q-1)\,h_{p,q,k}x^p y^q z^k \\ &= xH_x + yH_y - H \end{aligned} \tag{6.20}$$

The first term of the right part of Equation (6.17)

We conduct the addition on the term $qh_{p-1,q-1,k-1}$.

$$\begin{aligned} B &= \sum_{q\geq 1}\sum_{p\geq 2}\sum_{k\geq 1} qh_{p-1,q-1,k-1}x^p y^q z^k \\ &= xyz\sum_{q\geq 1}\sum_{p\geq 2}\sum_{k\geq 0}(q+1)\,h_{p,q,k}x^p y^q z^k + xyz\sum_{p\geq 1}\sum_{k\geq 0} h_{p,0,k}x^p z^k \\ &+xyz\sum_{q\geq 1}\sum_{k\geq 0}(q+1)\,h_{1,q,k}xy^q z^k \\ &= xyz\sum_{q\geq 1}\sum_{p\geq 2}\sum_{k\geq 0}(q+1)\,h_{p,q,k}x^p y^q z^k + xyz\sum_{p\geq 1}\sum_{k\geq 0} h_{p,0,k}x^p z^k \\ &+xyz\sum_{q\geq 1}\sum_{k\geq 0}(q+1)\,h_{1,q,k}xy^q z^k + xyz\sum_{q\geq 1}\sum_{p\geq 2}(q+1)\,h_{p,q,0}x^p y^q \\ &= xyz\,(yH_y + H) + xyz\sum_{p\geq 1}\sum_{k\geq 0} h_{p,0,k}x^p z^k + x^2yz^2\sum_{q\geq 1}(q+1)\,y^q \end{aligned} \tag{6.21}$$

where, the term $\sum_{q\geq 1}(q+1)\,y^q$ equals:

$$\sum_{q\geq 1}(q+1)\,y^q = \frac{y}{(1-y)^2} + \frac{y}{1-y} \tag{6.22}$$

The probability $h_{p,0,k}$ corresponds to the case that there is only white ball in the urn. Therefore, when $p > 2$, we choose always 2 white balls with POP rule. We have the recursive relation:

$$h_{p,0,k} = h_{p-2,0,k-1} \qquad p \geq 2 \text{ and } k \geq 1 \tag{6.23}$$

with the initial conditions: $h_{0,0,0} = 1$ and $h_{1,0,1} = 1$ as we have mentioned above. For any k is not 0, there are only two ways to make the $h_{p,0,k}$ non-zero, that is the case $p = 2k$ (With Equation (6.23), we finally get $h_{2k,0,k} = h_{0,0,0}$) or $p = 2k - 1$ ($h_{2k-1,0,k} = h_{1,0,1}$).

$$\begin{aligned} \sum_{p\geq 1}\sum_{k\geq 0} h_{p,0,k}x^p z^k &= \sum_{k\geq 1} h_{2k,0,k}x^{2k}z^k + \sum_{k\geq 1} h_{2k-1,0,k}x^{2k-1}z^k \\ &= \sum_{k\geq 1} x^{2k}z^k + \sum_{k\geq 1} x^{2k-1}z^k \\ &= \frac{x^2z+xz}{1-x^2z} \end{aligned} \tag{6.24}$$

The second term of the right part of Equation (6.17)

We conduct the addition on the term $(p-1)\,h_{p-2,q,k-1}$.

$$\begin{aligned} C &= \sum_{q\geq 1}\sum_{p\geq 2}\sum_{k\geq 1}(p-1)\,h_{p-2,q,k-1}x^p y^q z^k \\ &= x^2 z\sum_{q\geq 1}\sum_{p\geq 0}\sum_{k\geq 1}(p+1)\,h_{p,q,k}x^p y^q z^k + x^2 z\sum_{q\geq 1}\sum_{p\geq 2}(p+1)\,h_{p,q,0}x^p y^q \\ &+ x^2 z\sum_{q\geq 1}h_{0,q,0}y^q + 2x^2 z\sum_{q\geq 1}h_{1,q,1}xy^q z \\ &= x^2 z\,(xH_x + H) + (x^2 z + 2x^3 z^2)\,\tfrac{y}{1-y} \end{aligned} \tag{6.25}$$

PDE

Together with Equations (6.20), (6.21), (6.25), we finally get a partial differential equation:

$$\begin{aligned} &(x-x^3 z)\,H_x + (y-xy^2)\,H_y - (1+x^2 z+xyz)\,H \\ = \;& \tfrac{x^2 y z^2(x+1)}{1-x^2 z} + x^2 y^2 z^2\left(\tfrac{1}{(1-y)^2}+\tfrac{1}{1-y}\right) + (x^2 z + 2x^3 z^2)\,\tfrac{y}{1-y} \end{aligned} \tag{6.26}$$

□

6.6.3 Solution of PDE

In the following subsections, we introduce a methodology to resolve the PDE (6.19). Some useful techniques to resolve PDEs are introduced in [Cod89, Str92, Pol01]. The process of the resolution of PDEs is described as below:

Data: A first-order partial linear differential equation (PDE):
$f_1(x,y)\frac{\partial H(x,y)}{\partial x} + f_2(x,y)\frac{\partial H(x,y)}{\partial y} + f_3(x,y)H(x,y) = f_4(x,y)$
Result: The solution of the PDE

1 Obtain the corresponding characteristic differential system:
$\frac{dx(t)}{dt} = f_1(x(t), y(t))$ and $\frac{dy(t)}{dt} = f_2(x(t), y(t))$;
2 Resolve differential equation $y' = \frac{dy}{dx} = \frac{f_2(x,y)}{f_1(x,y)}$ and get solution $C = g(x,y)$. A first integral is $\xi(x,y) = g(x,y)$;
3 Apply a transformation from (x,y) to (η,ξ), where $\eta = x$ and $\xi = \xi(x,y)$ and obtain a PDE in form of $f_5(\eta,\xi)\,H_\eta(\eta,\xi) + f_6(\eta,\xi)\,H(\eta,\xi) = f_7(\eta,\xi)$;
4 Resolve reduced PDE $f_5(\eta,\xi)\,H_\eta(\eta,\xi) + f_6(\eta,\xi)\,H(\eta,\xi) = 0$ and the solution is $H^{[h]}(\eta,\xi)$;
5 Calculate the particular solution using equation:
$H^{[p]}(\eta,\xi) = H^{[h]}(\eta,\xi)\int_0^\eta \frac{f_7(q,\xi)}{f_5(q,\xi)}\frac{1}{H^{[h]}(q,\xi)}dq$;
6 Transform $H^{[p]}(\eta,\xi)$ back to $H^{[p]}(x,y)$ which is the solution of the original PDE. ;

Algorithm 6: Resolve the PDE.

In the following, we detail and specialize the Algorithm 6.

As the first step, we consider the corresponding reduced partial differential equation, which is given by

$$\left(x - x^3 z\right) H_x + \left(y - xy^2\right) H_y - \left(1 + x^2 z + xyz\right) H = 0 \tag{6.27}$$

To resolve PDE (6.27), we first obtain the system of characteristic differential equations:

$$\begin{array}{rcl} \frac{dx(t)}{dt} & = & x\left(t\right) - x\left(t\right)^3 z \\ \frac{dy(t)}{dt} & = & y\left(t\right) - x\left(t\right) y\left(t\right)^2 z \end{array}$$

The arising differential for $y = y(x)$ is given by

$$\frac{dx}{x - x^3 z} = \frac{dy}{y - xy^2 z} \tag{6.28}$$

The Equation (6.28) is a Bernoulli type[4], the solution is $C = \frac{\frac{x}{y} - 1}{(1 - x^2 z)^{\frac{1}{2}}}$, from which we obtain the first integral

$$\xi(x, y) = \frac{\frac{x}{y} - 1}{(1 - x^2 z)^{\frac{1}{2}}} \tag{6.29}$$

We then use a transformation from (x, y)-coordinates to (η, ξ)-coordinates via

$$\begin{array}{rcl} \eta & = & x \\ \xi & = & \frac{\frac{x}{y} - 1}{(1 - x^2 z)^{\frac{1}{2}}} \end{array}$$

or equivalently

$$\begin{array}{rcl} x & = & \eta \\ y & = & \frac{\eta}{\xi (1 - \eta^2 z)^{\frac{1}{2}} + 1} \end{array}$$

Then we obtain an inhomogeneous equation for $H(\eta, \xi) = H(x(\eta, \xi), y(\eta, \xi))$. As $H_x = H_\eta \frac{\partial \eta}{\partial x} + H_\xi \frac{\partial \xi}{\partial x}$ and $H_y = H_\eta \frac{\partial \eta}{\partial y} + H_\xi \frac{\partial \xi}{\partial y}$, the corresponding terms are given by

$$\begin{array}{rcl} \frac{\partial \eta}{\partial x} & = & 1 \\ \frac{\partial \xi}{\partial x} & = & \frac{\frac{1}{y}\left(1 - x^2 z\right)^{\frac{1}{2}} + xz\left(\frac{x}{y} - 1\right)\left(1 - x^2 z\right)^{-\frac{1}{2}}}{1 - x^2 z} \\ \frac{\partial \eta}{\partial y} & = & 0 \\ \frac{\partial \xi}{\partial y} & = & -\frac{1}{y^2} \frac{x}{(1 - x^2 z)^{\frac{1}{2}}} \end{array}$$

In fact, we can obtain a more easily resolvable PDE (there arise only the partial differential of variable η), if a transformation inspired from the first integral is applied.

$$\eta \left(1 - \eta^2 z\right) \frac{\partial H\left(\eta, \xi\right)}{\partial \eta} - \left(1 + \eta^2 z + \frac{\eta^2 z}{1 + \xi \left(1 - \eta^2 z\right)^{\frac{1}{2}}}\right) H\left(\eta, \xi\right) = G\left(\eta, \xi\right) \tag{6.30}$$

[4] An ordinary differential equation of the form $y' + P(x)y = Q(x)y^n$ is called a Bernoulli equation. Dividing the equation by y^n and a change of variable $w = \frac{1}{y^{n-1}}$, we yield a linear first-order differential equation: $\frac{w'}{1-n} + P(x)w = Q(x)$.

Finally, we get a PDE with only one partial differential. So it is can be considered as an ordinary differential equation (ODE).

The right term $G(\eta,\xi,z)$ can be calculated by the right part of Equation (6.19)

$$\begin{aligned} G(\eta,\xi,z) &= \frac{\eta^3 z^2(\eta+1)}{\left(1+\xi(1-\eta^2 z)^{\frac{1}{2}}\right)(1-\eta^2 z)} + \frac{\eta^4 z^2}{\left(1+\xi(1-\eta^2 z)^{\frac{1}{2}}-\eta\right)^2} \\ &+ \left(\frac{\eta z}{1+\xi(1-\eta^2 z)^{\frac{1}{2}}} + 2\eta z + 1\right)\frac{\eta^3 z}{1+\xi(1-\eta^2 z)^{\frac{1}{2}}-\eta} \end{aligned} \tag{6.31}$$

From Equation (6.30), we resolve first the corresponding reduced equation:

$$\eta\left(1-\eta^2 z\right)\frac{\partial H^{[h]}(\eta,\xi)}{\partial \eta} - \left(1+\eta^2 z + \frac{\eta^2 z}{1+\xi\left(1-\eta^2 z\right)^{\frac{1}{2}}}\right) H^{[h]}(\eta,\xi) = 0 \tag{6.32}$$

The solution of the homogeneous differential equation is given by

$$H^{[h]}(\eta,\xi) = C(\xi)\frac{\eta\left(1+\xi\left(1-\eta^2 z\right)^{\frac{1}{2}}\right)}{\left(1-\eta^2 z\right)^{\frac{3}{2}}} \tag{6.33}$$

The solution has the form $H = H^{[p]} + CH^{[h]}$. But the problems of the diminishing urn model lead always $C = 0$. So the solution is just $H^{[p]}$. The particular solution of Equation (6.30) can be expressed as $H^{[p]} = \mu(\eta) H^{[h]}$.
In general, given a PDE:

$$a(\eta)\frac{\partial H(\eta,\xi)}{\partial \eta} + b(\eta) H(\eta,\xi) = r(\eta)$$

Replace H by $\mu(\eta)H$:

$$\mu'(\eta) H^{[h]}(\eta,\xi) = \frac{r(\eta)}{a(\eta)}$$

$$\mu(\eta) = \int_0^\eta \frac{r(t)}{a(t)}\frac{1}{H^{[h]}(t,\xi)}dt$$

Replace t by ηw:

$$\mu(\eta) = \int_0^1 \eta\frac{r(\eta w)}{a(\eta w)}\frac{1}{H^{[h]}(\eta w,\xi)}dw$$

In Equation (6.30) $a(\eta w) = \eta w\left(1-\eta^2 w^2 z\right)$, so we have:

$$H^{[p]}(\eta,\xi,z) = \frac{\eta\left(1+\xi\left(1-\eta^2 z\right)^{\frac{1}{2}}\right)}{\left(1-\eta^2 z\right)^{\frac{3}{2}}}\int_0^1 \frac{r(\eta w)\left(1-\eta^2 w^2 z\right)^{\frac{1}{2}}}{\eta w^2\left(1+\xi\left(1-\eta^2 w^2 z\right)^{\frac{1}{2}}\right)}dw$$

where $r(\eta w) = G(\eta w,\xi,z)$.

To simplify the analysis, we rewrite $H(x,y,z) = H1 + H2 + H3$

$$H1\,(x,y,z) = \int_0^1 \frac{x^4 w z^2\,(xw+1)}{yF^2\,(1-x^2w^2z)^{\frac{1}{2}}\,(1-x^2z)^{\frac{3}{2}}}dw$$

$$H2\,(x,y,z) = \int_0^1 \frac{x^4 w z\,(1+2xwz)\,(1-x^2w^2z)^{\frac{1}{2}}}{y\,(1-x^2z)^{\frac{3}{2}}\,F\,(F-xw)}dw$$

$$H3\,(x,y,z) = \int_0^1 \frac{x^5 w^2 z^2\,(1-x^2w^2z)^{\frac{1}{2}}\,(2F-xw)}{y\,(1-x^2z)^{\frac{3}{2}}\,F^2\,(F-xw)^2}dw$$

where $F = 1 + \left(\frac{x}{y} - 1\right)\left(\frac{1-x^2w^2z}{1-x^2z}\right)^{\frac{1}{2}}$.

Finally, we resolve the Equation (6.19). The final solution $H^{[p]}$ is one with an integral. It is not necessary to reduce the integral. In the following step, there is no difficulty to extract the coefficients from a solution with an integral.

6.6.4 Extraction of the coefficient of $H(x,y,z)$

In this section we compute the exact coefficient of $H(x,y,z)$. The relative methods are referred from [HKP07, SF95, FS09].

Extraction of the coefficient of $H1(x,y,z)$

$$H1\,(x,y,z) = \int_0^1 \frac{x^4 w z^2\,(xw+1)}{yF^2\,(1-x^2w^2z)^{\frac{1}{2}}\,(1-x^2z)^{\frac{3}{2}}}dw \tag{6.34}$$

$$\begin{aligned}
&[x^p y^q z^k]H1 \qquad \text{replace } x^2z \text{ with } t\\
&= [x^{p-2k}y^{q+1}t^{k-2}]\int_0^1 \frac{w\,(xw+1)}{\left(1+\left(\frac{x}{y}-1\right)\left(\frac{1-tw^2}{1-t}\right)^{\frac{1}{2}}\right)^2\,(1-tw^2)^{\frac{1}{2}}\,(1-t)^{\frac{3}{2}}}dw\\
&= [x^{p-2k+q}t^{k-2}]n\sum_{i=0}^{q-1}\binom{q-1}{i}(-1)^{q-i-1}\int_0^1 w^2\,(1-t)^{\frac{q-i-2}{2}}\left(1-tw^2\right)^{\frac{i-q-2}{2}}dw\\
&+[x^{p-2k+q+1}t^{k-2}]n\sum_{i=0}^{q-1}\binom{q-1}{i}(-1)^{q-i-1}\int_0^1 w\,(1-t)^{\frac{q-i-2}{2}}\left(1-tw^2\right)^{\frac{i-q-2}{2}}dw\\
&= [x^{p-2k+q}]q\sum_{i=0}^{q-1}\binom{q-1}{i}(-1)^{q-i-1}\int_0^1 w^2\sum_{j=0}^{k-2}[t^j]\,(1-t)^{\frac{q-i-2}{2}}\,[t^{k-j-2}]\left(1-tw^2\right)^{\frac{i-q-2}{2}}dw\\
&+[x^{p-2k+q+1}]q\sum_{i=0}^{q-1}\binom{q-1}{i}(-1)^{q-i-1}\int_0^1 w\sum_{j=0}^{k-2}[t^j]\,(1-t)^{\frac{q-i-2}{2}}\,[t^{k-j-2}]\left(1-tw^2\right)^{\frac{i-q-2}{2}}dw
\end{aligned}$$

$$= [x^{p-2k+q}] q \sum_{i=0}^{q-1} \binom{q-1}{i} (-1)^{q-i-1} \sum_{j=0}^{k-2} \binom{j+\frac{i-q}{2}}{j} \binom{k-j+\frac{q-i}{2}-2}{k-j-2} \frac{1}{2(k-j)-1}$$

$$+[x^{p-2k+q+1}] q \sum_{i=0}^{q-1} \binom{q-1}{i} (-1)^{q-i-1} \sum_{j=0}^{k-2} \binom{j+\frac{i-q}{2}}{j} \binom{k-j+\frac{q-i}{2}-2}{k-j-2} \frac{1}{2(k-j)-2}$$

The Equation (6.35) implies that only if $p+q=2k$ or $p+q+1=2k$, the coefficient $[x^p y^q z^k]H1$ is not zero.

Extraction of the coefficient of $H2(x,y,z)$

$$H2(x,y,z) = \int_0^1 \frac{x^4 wz(1+2xwz)(1-x^2w^2z)^{\frac{1}{2}}}{y(1-x^2z)^{\frac{3}{2}} F(F-xw)} dw \tag{6.35}$$

where $F = 1 + \left(\frac{x}{y} - 1\right)\left(\frac{1-x^2w^2z}{1-x^2z}\right)^{\frac{1}{2}}$.

$[x^p y^q z^k]H2$ replace x^2z with t

$$= [x^{p-2k-1} y^{q+1} t^{k-1}] \int_0^1 \frac{w(x+2tw)(1-tw^2)^{\frac{1}{2}}}{(1-t)^{\frac{3}{2}}\left(1+\left(\frac{x}{y}-1\right)\left(\frac{1-tw^2}{1-t}\right)^{\frac{1}{2}}\right)\left(1+\left(\frac{x}{y}-1\right)\left(\frac{1-tw^2}{1-t}\right)^{\frac{1}{2}} - xw\right)} dw$$

$$= [x^{p+q-2k+1} t^{k-1}] \sum_{i=0}^{q} \binom{q}{i} \int_0^1 (x+2tw)(xw-1)^i \left(1-tw^2\right)^{-\frac{i}{2}} (1-t)^{\frac{i-2}{2}} dw$$

$$-[x^{p+q-2k+1} t^{k-1}] \sum_{i=0}^{q} \binom{n}{i} \int_0^1 (x+2tw)(-1)^i \left(1-tw^2\right)^{-\frac{i}{2}} (1-t)^{\frac{i-2}{2}} dw$$

$$= [x^{p+q-2k}] \sum_{i=0}^{q} \binom{q}{i} \int_0^1 (xw-1)^i \sum_{j=0}^{k-1} [t^j] \left(1-tw^2\right)^{-\frac{i}{2}} [t^{k-1-j}] (1-t)^{\frac{i-2}{2}} dw$$

$$+[x^{p+q-2k+1}] \sum_{i=0}^{q} \binom{q}{i} \int_0^1 2w(xw-1)^i \sum_{j=0}^{k-1} [t^j] \left(1-tw^2\right)^{-\frac{i}{2}} [t^{k-2-j}] (1-t)^{\frac{i-2}{2}} dw$$

$$-[x^{p+q-2k}] \sum_{i=0}^{q} \binom{n}{i} \int_0^1 (-1)^i \sum_{j=0}^{k-1} [t^j] \left(1-tw^2\right)^{-\frac{i}{2}} [t^{k-1-j}] (1-t)^{\frac{i-2}{2}} dw$$

$$-[x^{p+q-2k+1}] \sum_{i=0}^{q} \binom{n}{i} \int_0^1 2w(-1)^i \sum_{j=0}^{k-1} [t^j] \left(1-tw^2\right)^{-\frac{i}{2}} [t^{k-2-j}] (1-t)^{\frac{i-2}{2}} dw$$

If $p+q \neq 2k$ and $p+q \neq 2k-1$, then

$[x^p y^q z^k]H2(x,y,z)$

$$= [x^{p+q-2k}] \sum_{i=0}^{q} \binom{q}{i} \sum_{j=0}^{k-1} \binom{j+\frac{i}{2}-1}{j} \binom{k-j-1-\frac{i}{2}}{k-j-1} \int_0^1 (xw-1)^i \, q^{2j} dw$$

$$+[x^{p+q-2k+1}] \sum_{i=0}^{q} \binom{q}{i} \sum_{j=0}^{k-2} \binom{j+\frac{i}{2}-1}{j} \binom{k-j-2-\frac{i}{2}}{k-j-2} \int_0^1 (xw-1)^i \, 2q^{2j+1} dw$$

If $p \leq 2k-1$, then

$$\begin{aligned}
&[x^p y^q z^k] H2(x,y,z) \\
&= \sum_{i=0}^{q} \binom{q}{i} \sum_{j=0}^{k-1} \binom{j+\frac{i}{2}-1}{j} \binom{k-j-1-\frac{i}{2}}{k-j-1} \binom{i}{p+q-2k} \\
&\times (-1)^{i-p-q+2k} \frac{1}{p+q-2k+2j+1} \\
&+2 \sum_{i=0}^{q} \binom{q}{i} \sum_{j=0}^{k-2} \binom{j+\frac{i}{2}-1}{j} \binom{k-j-2-\frac{i}{2}}{k-j-2} \binom{i}{p+q-2k+1} \\
&\times (-1)^{i-p-q+2k-1} \frac{1}{p+q-2k+2j+3}
\end{aligned}$$

Extraction of the coefficient of $H3(x,y,z)$

$$H3(x,y,z) = \int_0^1 \frac{x^5 w^2 z^2 \left(1-x^2w^2z\right)^{\frac{1}{2}} (2F-xw)}{y \left(1-x^2z\right)^{\frac{3}{2}} F^2 (F-xw)^2} dw \tag{6.36}$$

where $F = \left(1 + \left(\frac{x}{y}-1\right)\left(\frac{1-x^2w^2z}{1-x^2z}\right)^{\frac{1}{2}}\right)$.

$$\begin{aligned}
&[x^p y^q z^k] H3 \qquad \text{replace } x^2 z \text{ with } t \\
&= [x^{p-2k-1} y^{q+1} t^{k-2}] \int_0^1 \frac{w^2 \left(1-tw^2\right)^{\frac{1}{2}} \left(2\left(1+\left(\frac{x}{y}-1\right)\left(\frac{1-tw^2}{1-t}\right)^{\frac{1}{2}}\right)-xw\right)}{(1-t)^{\frac{3}{2}} \left(1+\left(\frac{x}{y}-1\right)\left(\frac{1-tw^2}{1-t}\right)^{\frac{1}{2}}\right)^2 \left(1+\left(\frac{x}{y}-1\right)\left(\frac{1-tw^2}{1-t}\right)^{\frac{1}{2}}-xw\right)^2} dw \\
&= -2[x^{p+q-2k+2} t^{k-2}] \int_0^1 \frac{(1-t)^{\frac{q}{2}-1}}{\left(1-tw^2\right)^{\frac{q}{2}}} \left(\left(\frac{1-tw^2}{1-t}\right)^{\frac{1}{2}} - 1\right)^q dw \\
&-2[x^{p+q-2k+2} t^{k-2}] \int_0^1 \frac{(1-t)^{\frac{q}{2}-1}}{\left(1-tw^2\right)^{\frac{q}{2}}} \left(\left(\frac{1-tw^2}{1-t}\right)^{\frac{1}{2}} + xw - 1\right)^q dw \\
&+2[x^{p+q-2k+2} t^{k-2}] \int_0^1 \frac{(1-t)^{\frac{q}{2}-1}}{\left(1-tw^2\right)^{\frac{q}{2}}} \left(\left(\frac{1-tw^2}{1-t}\right)^{\frac{1}{2}} - 1\right)^q dw
\end{aligned}$$

$$-n[x^{p+q-2k+1}t^{k-2}]\int_0^1 \frac{w\,(1-t)^{\frac{q}{2}-1}}{(1-tw^2)^{\frac{q}{2}}}\left(\left(\frac{1-tw^2}{1-t}\right)^{\frac{1}{2}}-1\right)^{q-1}dw$$

$$+2[x^{p+q-2k+2}t^{k-2}]\int_0^1 \frac{(1-t)^{\frac{q}{2}-1}}{(1-tw^2)^{\frac{q}{2}}}\left(\left(\frac{1-tw^2}{1-t}\right)^{\frac{1}{2}}+xw-1\right)^{q}dw$$

$$+n[x^{p+q-2k+1}t^{k-2}]\int_0^1 \frac{w\,(1-t)^{\frac{q}{2}-1}}{(1-tw^2)^{\frac{q}{2}}}\left(\left(\frac{1-tw^2}{1-t}\right)^{\frac{1}{2}}+xw-1\right)^{q-1}dw$$

If $p+q-2k+1\neq 0$

$$[x^py^qz^k]H3$$

$$=q[x^{p+q-2k+1}t^{k-2}]\int_0^1 \frac{w\,(1-t)^{\frac{q}{2}-1}}{(1-tw^2)^{\frac{q}{2}}}\left(\left(\frac{1-tw^2}{1-t}\right)^{\frac{1}{2}}+xw-1\right)^{q-1}dw$$

$$=[t^{k-2}]q\int_0^1 \frac{(1-t)^{\frac{q}{2}-1}}{(1-tw^2)^{\frac{q}{2}}}\binom{q-1}{p+q-2k+1}w^{p+q-2k+2}\left(\left(\frac{1-tw^2}{1-t}\right)^{\frac{1}{2}}-1\right)^{2k-p-2}dw$$

If $p\leq 2k-2$

$$[x^py^qz^k]H3$$

$$=[t^{k-2}]q\int_0^1 \sum_{i=0}^{2k-p-2}\binom{2k-p-2}{i}\left(1-tw^2\right)^{\frac{i-q}{2}}(1-t)^{\frac{q-i}{2}-1}$$

$$\times(-1)^{2k-p-i-2}\binom{p-1}{p+q-2k+1}w^{p+q-2k+2}dw$$

$$=q\int_0^1 \sum_{i=0}^{2k-p-2}\binom{2k-p-2}{i}(-1)^{2k-p-i-2}\binom{p-1}{p+q-2k+1}w^{p+q-2k+2}$$

$$\times\sum_{j=0}^{k-2}[t^j]\left(1-tw^2\right)^{\frac{i-q}{2}}[t^{k-2-j}](1-t)^{\frac{q-i}{2}-1}\,dw$$

$$=q\sum_{i=0}^{2k-p-2}\binom{2k-p-2}{i}(-1)^{2k-p-i-2}\binom{p-1}{p+q-2k+1}$$

$$\times\sum_{j=0}^{k-2}\binom{k-j-2-\frac{q-i}{2}}{k-j-2}\binom{j+\frac{q-i}{2}-1}{j}\frac{1}{p+q-2k+2j+3}$$

We obtain a complex expression that is a solution of computation but gives little intuitive impression. For the further analysis we rely on average value of the number of iterations.

6.6.5 Expectation of the number of POP

In the last subsection, we developed the coefficients of $H(x, y, z)$ in an explicit form. These expressions have not much intuitive meaning, therefore we resolve the expectation by another approach.

Theorem 7. Let $h_{p,q,k}$ denote the probability that for a urn with p white balls and q black balls, after k steps there is no white ball in the urn using POP rule. Then the expected number of POP is

$$E_{p,q}(k) = \sum_k k h_{p,q,k} = p - \frac{p^2}{2\,(p+q)} + O\left(\frac{p}{\sqrt{q}\sqrt{p+q}}\right) \tag{6.37}$$

Proof. We denote X_k (respectively Y_k) the random variable of the number of white balls (black balls) in the urn after k times POP rule have been conducted, while the initial conditions are $X_0 = p$ and $Y_0 = q$. Then we have,

$$\begin{aligned} X_{k+1} &= \frac{X_k - 1}{X_k + Y_k - 1}\,(X_k - 2) + \frac{Y_k}{X_k + Y_k - 1}\,(X_k - 1) \\ Y_{k+1} &= \frac{X_k - 1}{X_k + Y_k - 1} Y_k + \frac{Y_k}{X_k + Y_k - 1}\,(Y_k - 1) \\ X_k + Y_k &= X_0 + Y_0 - 2k \end{aligned}$$

By reducing the variable Y_k, we simplified the system of equations as follows :

$$X_{k+1} = \frac{(X_0 + Y_0 - 2k - 2)\,(X_k - 1)}{X_0 + Y_0 - 2k - 1} \tag{6.38}$$

The Equation (6.38) is a standard recursion with form: $X_{k+1} = a_{k+1}X_k + b_{k+1}$, where $a_{k+1} = \frac{X_0+Y_0-2k-2}{X_0+Y_0-2k-1}$ and $b_{k+1} = -a_{k+1}$. We divide both sides by the factor $a_{k+1}a_k...a_0$, then we have:

$$\frac{X_{k+1}}{a_{k+1}a_k...a_1a_0} = X_0 + \sum_{i=0}^{k} \frac{b_{i+1}}{a_{i+1}...a_1a_0} \tag{6.39}$$

The $\{X_k\}$ is a decreasing sequence, so at some moment k, we have $\lfloor X_k \rfloor = 0$. To resolve the corresponding k, we consider the left part of the Equation (6.39) equals to 0. The right part then can be simplified as follows.

We denote $T = X_0 + Y_0$, then $p = X_0 = \sum\limits_{i=0}^{k-1} \frac{1}{a_i...a_1a_0}$

$$\begin{aligned} p &= \sum_{i=0}^{k-1} \frac{1}{a_i...a_1a_0} = \sum_{i=1}^{k} \frac{1}{a_i...a_1} \qquad \text{as } a_k = \tfrac{T-2k}{T-2k+1} \\ &= \sum_{i=1}^{k} \frac{(T-1)!!}{(T-2i-1)!!} \frac{(T-2i-2)!!}{(T-2)!!} \end{aligned}$$

$$= \sum_{i=1}^{k} \frac{T!}{\left(\frac{T}{2}\right)!\left(\frac{T-2}{2}\right)!} 2^{-2i} \frac{\left(\frac{T-2i-2}{2}\right)!\left(\frac{T-2i}{2}\right)!}{(T-2i)!}$$

$$= \sum_{i=1}^{k} \frac{T!}{\left(\frac{T}{2}\right)!\left(\frac{T-2}{2}\right)!} \sqrt{\frac{\pi}{2}} e 2^{-T+1} \sqrt{T-2i-2} \frac{(T-2i-2)^{\frac{T-2i-2}{2}}}{(T-2i)^{\frac{T-2i}{2}}} \left(1 + O\left(\frac{1}{T-2i}\right)\right)$$

Applying Stirling formula with error term $O\left(\frac{1}{T-2i}\right)$

$$= \sum_{i=1}^{k} \frac{T!}{\left(\frac{T}{2}\right)!\left(\frac{T-2}{2}\right)!} \sqrt{\frac{\pi}{2}} e 2^{-T+1} \left(1 - \frac{2}{T-2i}\right)^{\frac{T-2i}{2}} \frac{1}{\sqrt{T-2i-2}} \left(1 + O\left(\frac{1}{T-2i}\right)\right)$$

$$= \frac{T!}{\left(\frac{T}{2}\right)!\left(\frac{T-2}{2}\right)!} \sqrt{\frac{\pi}{2}} 2^{-T+1} \sum_{i=1}^{k} \frac{1}{\sqrt{T-2i-2}} \left(1 + O\left(\frac{1}{T-2i}\right)\right)$$

where $\left(1 - \frac{2}{T-2i}\right)^{\frac{T-2i}{2}} \to e^{-1}$

$$= \frac{T!}{\left(\frac{T}{2}\right)!\left(\frac{T-2}{2}\right)!} \sqrt{\frac{\pi}{2}} 2^{-T+1} \left(\sqrt{T-2} - \sqrt{T-2k-2} + O\left(\frac{1}{\sqrt{Y_0}}\right)\right)$$

where $\frac{T}{2} \gg 0$

$$= \sqrt{T-2} \left(\sqrt{T-2} - \sqrt{T-2k-2} + O\left(\frac{1}{\sqrt{T}}\right)\right) \left(1 + O\left(\frac{1}{T}\right)\right)$$

$$= \sqrt{T-2} \left(\sqrt{T-2} - \sqrt{T-2k-2}\right) \left(1 + O\left(\frac{1}{\sqrt{Y_0}\sqrt{T}}\right)\right) \left(1 + O\left(\frac{1}{T}\right)\right)$$

$$= \sqrt{T-2} \left(\sqrt{T-2} - \sqrt{T-2k-2}\right) \left(1 + O\left(\frac{1}{\sqrt{Y_0}\sqrt{T}}\right)\right)$$

Then we have $E_{p,q}(k) = X_0 - \frac{X_0^2}{2T} + O\left(\frac{X_0}{\sqrt{Y_0}\sqrt{X_0+Y_0}}\right) = p - \frac{p^2}{2(p+q)} + O\left(\frac{p}{\sqrt{q}\sqrt{p+q}}\right)$. □

In the proof, we apply several big O expression. In case when $T \gg 0$, all conditions are satisfactory.

Theorem 7 has an intuitive explanation: if $X_0 \gg Y_0$, almost each step we throw two white balls, then $E_{X_0,Y_0}(POP) \approx \frac{X_0}{2}$; whereas when $Y_0 \gg X_0$, at each step we throw one white ball and one black ball, then $E_{X_0,Y_0}(POP) \approx X_0$.

6.7 Analyzing the profile of BFS tree

With the expected number of POP operation between time i_k and i_{k+1}, we may then express i_{k+1} as a function of i_k, and deduce the expected values of the interesting quantities of the profile.

In the following we shall make use of the generating function of degrees given by $g(z) = \sum_j a_j z^j$ where $\{a_j\}$ is the degree distribution. We obviously have $g(1) = 1$ and $g'(1) = \frac{m}{n}$.

Lemma 5. *Given a degree sequence $\{d_j\}$ in the form of a generating function $g(z) = \sum_j d_j z^j$, the time i_{k+1} when the BFS process begins the exploration of the nodes at level $k+1$ satisfies:*

$$i_{k+1} \sim g'\left(\frac{i_k}{m}\right) \tag{6.40}$$

Proof. Using Equation (6.9) and Equation (6.7), the expected number of enqueued copies is:

$$X = \delta n \left(\frac{i}{m}\right)^2 - \frac{i}{m} g'\left(\frac{i}{m}\right) n \tag{6.41}$$

The expected number of untouched copies is:

$$Y = \frac{i}{m} g'\left(\frac{i}{m}\right) n \tag{6.42}$$

From Equation (6.11), we get

$$\sum_{i=i_k}^{i_{k+1}} POP\,(i) = \frac{C_{ue}\,(i_k) - C_{ue}\,(i_{k+1})}{2} \tag{6.43}$$

The left side of Equation (6.43) is just the number of POP for an initial configuration (X, Y). Using Theorem 7, we have

$$\begin{array}{rcl} \frac{C_{ue}(i_k) - C_{ue}(i_{k+1})}{2} & = & \sum_{i=i_k}^{i_{k+1}} POP\,(i) \\ & = & X - \frac{X^2}{2(X+Y)} \\ & = & \frac{i_k^2}{2m} - \frac{n}{2\delta} g'\left(\frac{i_k}{m}\right)^2 \end{array}$$

Then we draw the conclusion $i_{k+1} \sim n g'\left(\frac{i_k}{m}\right)$. □

Applying Lemma 5, some typical cases can be easily computed,

Corollary 1. *For a Regular r graph with n nodes:*

$$i_{k+1} \sim rn \left(\frac{i_k}{m}\right)^{r-1}$$

Corollary 2. *For a Poisson λ graph with n nodes:*

$$i_{k+1} \sim n\lambda e^{\lambda\left(\frac{i_k}{m} - 1\right)}$$

Corollary 3. *For a power-law α graph with n nodes:*

$$i_{k+1} \sim n \frac{\sum_j j^{1-\alpha} \left(\frac{i_k}{m}\right)^{j-1}}{\sum_j j^{-\alpha}}$$

Theorem 8. Let n_k denote the number of nodes of level k in the BFS tree, respectively, n_{k-1}, n_{k-2},..., n_1 the number of nodes of level $k-1$, $k-2$,... , 1 and a degree distribution $\{a_j\}$ in form of generating function $g(z) = \sum_{j=1} a_j z^j$, then the expected number of nodes $E(n_{k+1})$ of level $k+1$ in the BFS tree is:

$$E\left(n_{k+1}\right) = n - \sum_{i=0}^{k} n_i - ng\left(\frac{g'\left(g^{-1}\left(1 - \sum_{i=0}^{k} \frac{n_i}{n}\right)\right)}{g'\left(1\right)}\right) \tag{6.44}$$

Proof. Using Lemma 5, we have:

$$E\left(n_{k+1}\right) = ng\left(\frac{i_k}{m}\right) - ng\left(\frac{i_{k+1}}{m}\right)$$
$$E\left(n_{k+1}\right) = ng\left(\frac{i_k}{m}\right) - ng\left(\frac{g'\left(\frac{i_k}{m}\right)}{g'\left(1\right)}\right)$$

□

From Theorem 8, for three typical types of the degree distribution, we have:

Corollary 4. *For a regular r graph with n nodes:*

$$E\left(n_{k+1}\right) = n\left(\frac{i_k}{m}\right)^r - n\left(\frac{i_k}{m}\right)^{r(r-1)} \tag{6.45}$$

Corollary 5. *For a Poisson λ graph with n nodes:*

$$E\left(n_{k+1}\right) = ne^{\lambda\left(\frac{i_k}{m}-1\right)} - ne^{\lambda\left(e^{\lambda\left(\frac{i_k}{m}-1\right)}-1\right)} \tag{6.46}$$

Corollary 6. *For a power-law α graph with n nodes:*

$$E\left(n_{k+1}\right) = n\frac{\sum j^{-\alpha}\left(\frac{i_k}{m}\right)^j}{\zeta\left(\alpha\right)} - n\frac{\sum j^{-\alpha}\left(\frac{\sum j^{1-\alpha}\left(\frac{i_k}{m}\right)^{j-1}}{\zeta(\alpha-1)}\right)^j}{\zeta\left(\alpha\right)} \tag{6.47}$$

Theorem 9. Let e_k denote the number of invisible links between level k and level $k+1$ in the BFS tree, and a degree distribution $\{a_j\}$ in form of generating function $g(z) = \sum_{j=1} a_j z^j$, then the expectation of $E(e_k)$ is:

$$E\left(e_k\right) = \frac{i_k^2 - n^2 g'\left(\frac{i_k}{m}\right)^2}{2m} - m\left(g\left(\frac{i_k}{m}\right) - g\left(\frac{g'\left(\frac{i_k}{m}\right)}{g'\left(1\right)}\right)\right) + O\left(\frac{1}{m}\right) \tag{6.48}$$

Proof. At time i, the conditioned probability that an invisible link appears is:

$$P_{IL}(i) = E\left(POP(i)\right) \times \left(1 - \frac{E\left(C_{unto}\left(i\right)\right)}{E\left(C_{unex}\left(i\right)\right)}\right) \tag{6.49}$$

The first factor of the right part of Equation (6.49) is the probability that at time i, a POP occurs. The second factor of the right part is the probability that the partner of the copy at the head of queue Q is also located in the Q (that means an invisible link). Using Lemma 5, we have the critic points of time that indicates the beginning of a level. Then from time i_k to i_{k+1}, the expected number of invisible links is:

$$\begin{aligned}
E\left(e_k\right) &= \sum_{i=i_k}^{i_{k+1}} \frac{2i-1}{2m}\left(1 - \frac{ig'\left(\frac{i}{m}\right)}{\delta n\left(\frac{i}{m}\right)^2}\right) \\
&= m\int_{\frac{i_k}{m}}^{\frac{i_{k+1}}{m}} t\left(1 - \frac{g'\left(t\right)}{\delta t}\right)dt + O\left(\frac{1}{m}\right) \\
&= \frac{i_k^2 - i_{k+1}^2}{2m} - n\left(g\left(\frac{i_k}{m}\right) - g\left(\frac{i_{k+1}}{m}\right)\right) + O\left(\frac{1}{m}\right)
\end{aligned}$$

□

6.8 Conclusion

We studied the profile of BFS trees originated from a set of graphs given by a degree sequence. Our analysis is based on configuration model for random graphs, manipulating copies of the nodes. In the configuration model, the process of constructing a BFS tree, uses a queue to record the behavior of the copies. Discretizing time into an interval from m to 1, we establish a method to trace the queue, particularly the number of copies and the number of nodes at a certain time.

The core problem of the analysis is POP problem, that may be transformed into a classic balanced diminishing urn model. This diminishing urn model is characterized by a partial differential equation whose solution has an integral form, and we can extract the exact coefficients $h_{p,q,k}$. However, the expression of the $h_{p,q,k}$ is so complex that the computation of the expectation and the variance is not so practical. Then we directly calculate the expectation $\sum_k kh_{p,q,k}$ which has a relatively simple expression and using the expectation we get the results concerning the number of nodes at each level (Theorem 8) and the number of invisible links at each level (Theorem 9).

Other properties of the profile should be also attainable with our method, such as the height of the BFS tree. But in practice this property plays a relatively light

role (because it grows too slowly with respect to the size) in distinguishing the type and predict the number of links.

Since we directly calculate the expectation $\sum_k k h_{p,q,k}$, we cannot trivially further develop higher moments: the computation of variance for example, appeals to probability generating function [HKP07, FS09]. The next step of our work will be first to try and simplify the expression $h_{p,q,k}$ and hope to find an explicit distribution to describe it.

Chapter 7

Deciding with the profile of BFS trees

Contents

7.1 Introduction

In this chapter, we decide on the type of an unknown graph using not only a BFS tree but also the information of profile that we study in the previous chapter. In Subsection 7.2, we introduce two new strategies RRIL and PPIL, which are the counterparts of RR and PP strategy. In RRIL and PPIL strategies, we use the number of invisible links at each level to refine the process of rebuilding. In Subsection 7.3, we make use of the number of nodes at each level, from which we may decide on the type and the number of links using a bounded BFS tree.

7.2 RRIL and PPIL strategies

As we know the profile of BFS tree, some improvements can be applied in the process of rebuilding. In Chapter 3, two rebuilding strategies, RR strategy (Subsection 3.3.1) and PP (Subsection 3.3.2) strategy, have been described. In this section, we introduce a methodology that rebuilds graphs using the analytic results of the profile of BFS tree. Two new strategies RRIL and PPIL take place of RR and PP strategy. The process of RR and PP strategy is one that retrieves the invisible links, in other words to retrieve the links that present in the underlying graph but not in the tested BFS tree. However, in the process of RR and PP, we have missed an important factor that the links near around the root, namely the monitor from which we conduct traceroute, have a bigger probability to be visible [DAHB+06]. Therefore, there are less invisible links located around the root.

An intrinsic bias is introduced by the fact that RR and PP strategies uniformly choose from all allowed positions. But in strategies RRIL and PPIL, we improve the adding process by considering the fraction of missing links at each level. For example, if we have a sequence of missing links: $\{e_k\} = (1, 4, 5, 1)$, then one link will be added on level 1, four links on level 2, five links on level 3 and one link on level 4.

In mathematics, the number of invisible links at level k is a random variable that depends on the number of nodes n, degree sequence $\{d_j\}$ and the corresponding level k. Its expectation $E(e_k)$ is given by the expression of Theorem 9.

Then, we propose a process of rebuilding graphs. We call it $SB(RRIL, PPIL)$ which means the algorithm of deciding on the type without m with a single BFS tree using RRIL and PPIL strategies. See Algorithm 7.

Remarks:

- The theoretical value e_k is not guaranteed to be an integer. Therefore, in practice, we just take the integer part and we neglect those levels that have a value $e_k < 1$.

- The e_k links are randomly chosen among all allowed positions between levels (k, k) and levels $(k, k+1)$. The number of possible positions is $n'_k = \frac{n_k(n_k-1)}{2} +$

Data: The number of nodes n, a complete BFS tree T.
Result: Type of the underlying graph and the number of links m.

Compute the set of allowed positions $\{E_{allowed,k}\}$ according to T for each level k;
$KS_{min} \leftarrow \infty$;
foreach *hypothesis in* {*Poisson, power-law*} **do**
  for $m_{test} \leftarrow m_{begin}$ **to** m_{end} **Step** Δm **do**
    The rebuilt graph $G'_{hypothesis,m_{test}} \leftarrow T$;
    for $k \leftarrow 1$ **to** $level_{bfs}$ **do**
      $m_{add} \leftarrow e_k$;
      while $m_{add} > 1$ **do**
        if *hypothesis Poisson* **then**
          Randomly choose the first endpoint u and the second endpoint v from $E_{allowed,k}$ (RRIL) ;
        else
          Preferentially choose the first endpoint u and the second endpoint v from $E_{allowed,k}$ (PPIL) ;
        end
        Add link uv into $G'_{hypothesis,m_{test}}$, $G'_{hypothesis,m_{test}} \leftarrow G'_{hypothesis,m_{test}} + uv$;
        $m_{add} \leftarrow m_{add} - 1$;
      end
    end
    Compute the theoretical distribution $P^{type,m}$ corresponding to $(type, m)$;
    $KS_{test} \leftarrow KS(P^{type,m}, P^{G'_{type,m_{test}}})$;
    if $KS_{test} < KS_{min}$ **then**
      $KS_{min} \leftarrow KS_{test}$;
      $type \leftarrow hypothesis$;
      $m \leftarrow m_{test}$
    end
  end
end

Algorithm 7: $SB(RRIL, PPIL)$: Algorithm of deciding on the type of graph without m with a single BFS tree using RRIL and PPIL strategies.

$n_k n_{k+1} - n_{k+1}$. The first term indicates the number of links between (k, k); the second term indicates the number of links between $(k, k+1)$; the third term indicates the number of links that exist in BFS tree. So the randomness rebuilding is to choose e_k links from n'_k possible positions. However, in practice, n'_k may be smaller than e_k, the real program must handle this problem. We propose that at each level we take the minimum of e_k and n'_k.

- Two kinds of methods to retrieve a link, randomly or preferentially, we call the two strategies RRIL and PPIL strategy, respectively.

- RRIL and PPIL strategies are more sensitive to the form of BFS tree than RR and PP strategies. In theory, the profile of BFS tree should be exponentially increasing at the first levels, while in practice we observe sometimes a real measure has one node at level 1, one node at level 2, one node at level 3 and then it disperse to a large number of nodes at level 4. Such a BFS tree like this will make our strategy a little stupid, but there is no much impact with RR and PP strategies.

7.2.1 Random model graphs

First, we apply RRIL and PPIL strategies on random model graphs with 1000 nodes. The second column of Table 7.1 is the estimated type and parameters where the process gave the minimum KS value. The third and the fourth columns are m, the number of links in the underlying graph, and the estimated number of links m'.

Table 7.1: RRIL and PPIL strategies on random model graphs.

	Decide Type	m	m'
Poisson 3	Poisson 3	1612	1500
Poisson 5	Poisson 4.7	2486	2350
Poisson 10	Poisson 9.9	4863	4950
Power 2.1	Power 2.2	2461	1795
Power 2.2	Power 2.25	1254	1436
Power 2.3	Power 2.3	1083	1299

This application is similar to the methodology that we decide on the type of a graph without m described in Section 4.2. Comparing with Table 4.1 and Table 4.2, RRIL and PPIL strategies run still well for deciding on the type. With the help of the profile of BFS tree, the estimate of the number of links looks like better than the previous estimate. In practice, RRIL and PPIL can be considered the improvement of RR and PP strategies.

7.2.2 Real-world graphs

Then, we apply RRIL and PPIL on some real-world graphs.

Table 7.2: Using RRIL and PPIL strategies on real-world graphs.

	Decided Type	m	m'
Skitter-AS	Power-law	12025	8008
Radar-japon	Poisson	77545	46012
Radar-cm	Poisson	15728	36431
Radar-ortolan	Poisson	48516	40796
Radar-enix	Poisson	73556	52549

First for deciding on the type, in all cases Algorithm $SB(RRIL, PPIL)$ gives the same decided type to Algorithm $SB(RR, PP)$. But for estimating the number of links, it is even worse than $SB(RR, PP)$. This may be due to the abnormal structure of Radar graphs. Specially speaking, if a tested BFS tree has not an approximately exponentielly increasing form, RRIL and PPIL are not comptatible in this case and they are lack of the stability. The stability of stategy means that two samplings may not differ significantly from each other.

7.3 Deciding from a bounded BFS tree

As we have mentioned in Theorem 8, with a given degree distribution or a degree sequence and the node number n, we can compute a theoretical vector V of node numbers $\{n_k\}$, where n_k is denoted as the number of nodes at level k in the BFS tree T. For a fix n, a suite of vectors V, such as $V_{Poisson3}$, $V_{Poisson4}$ and $V_{Powerlaw2.1}$, $V_{Powerlaw2.2}$, are determined, where each V is a BFS tree vector:

Definition 8. BFS tree Vector *V: $V = (n_1, n_2, ...n_k...)$, where n_k corresponds to the number of nodes at level k in a BFS tree T.*

Remark: The BFS tree vector from a real BFS tree contains only integer elements; while the vector from a theoretical profile contains real number elements.

Then we propose a process of deciding the type of a graph from its BFS tree. It is described in Figure 7.1. G is an unknown graph on which we perform a measurement which gives its number of nodes n and a BFS tree T, which is possible to be a bounded BFS tree (limited by the number of hops, that is to say, a partial BFS tree with first i levels). In practice, we usually choose from 5 to 10 upper levels of a BFS tree. Then for each type, such as regular, Poisson and power-law, we build a family of graphs, where $V_{i,j}$ means the theoretical BFS profile computed with type i and parameter j. In the comparison step, we compare the difference of two BFS vectors, V_{bfs} and one $V_{i,j}$ and decide on the type and the corresponding parameter according to where there is least difference.

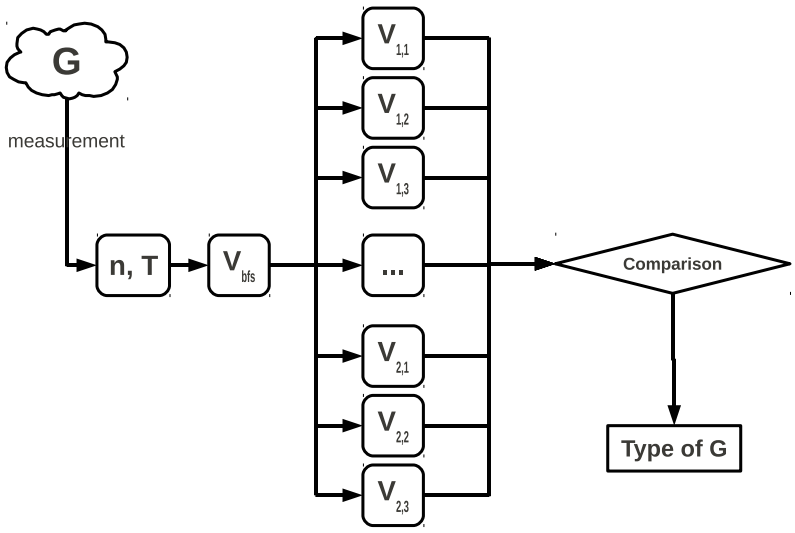

Figure 7.1: G is an unknown graph on which we perform a measurement which gives its number of nodes n and a BFS tree T, which is possible to be a bounded BFS tree (limited by the number of hops, that is to say, a partial BFS tree with first k levels). In practice, we usually choose from 5 to 10 upper levels of a BFS tree. Then for each type, such as regular, Poisson and power-law, we build a family of graphs, where $V_{i,j}$ means the theoretical BFS profile computed with type i and parameter j. In the step of comparison, we compare the difference of two BFS vectors, V_{bfs} and one $V_{i,j}$ and decide on the type and the corresponding parameter according to where there is least difference.

From the statistic view, there are a lot of methods to quantify the difference between two vectors, such as Manhattan distance (1-norm distance: $\sum_{i=1}^{n} |x_i - y_i|$), Euclidean distance (2-norm distance: $\sum_{i=1}^{n} (|x_i - y_i|^2)^{\frac{1}{2}}$), Chebyshev distance (infinity norm distance: $\lim_{p\to\infty} \sum_{i=1}^{n} (|x_i - y_i|^p)^{\frac{1}{p}}$) etc (see [ED06]). Then we have Algorithm 8 and we call it NV which means algorithm decides on the type using Node Vector.

Data: The number of nodes n, a bounded by k levels BFS tree T_k.
Result: Type of the underlying graph and the number of links m.

Compute a node vector of T_k: $V_{bfs} \leftarrow T_k$;
$MD_{min} \leftarrow 1$;
foreach *type in* {*Poisson, power-law*} **do**
  for $m_{test} \leftarrow m_{begin}$ **to** m_{end} **Step** Δm **do**
    The rebuilt graph $G'_{type,m_{test}} \leftarrow T$;
    Compute the corresponding node vector: $V_{type,m}$
    $MD_{test} \leftarrow MD(V_{bfs}, V_{type,m})$;
    if $MD_{test} < MD_{min}$ **then**
      $MD_{min} \leftarrow MD_{test}$;
      Record the position of m_{test} ;
    end
  end
end

Algorithm 8: NV: Algorithm of deciding on the type of graph without m with a single (Bounded) BFS tree using node vector.

Then we apply the methodology with two kinds of graphs: random model graphs and real-world graphs.

7.3.1 Random model graphs

With the process described in Figure 7.1, we conducted the experiments with random model graphs, each graph has 100000 nodes. In Table 7.3, we show the results of four typical cases: Poisson 3, Poisson 10, Power-law 2.1 and Power-law 2.3. The first column is the type of the tested graphs, while the second is the result estimated by a complete BFS tree and the third is the result estimated by upper levels of a BFS tree. The number of the upper levels is not fix and it depends on the form of BFS tree, which is first increasing and then decreasing. The appropriate number, considering of the reason in practice, is smaller than the height of increasing stage. If it is too small, there is no enough information to distinguish the type. On the contrary, it's difficult (almost impossible) to obtain a total view of the Internet to a too far distance. Therefore, we choose 6 upper levels for Poisson 3 (see detail in Table 7.4), while 5 for Poisson 10, Power-law

2.1 and Power-law 2.2. In all four cases, the type estimated with a complete BFS

Table 7.3: Result of deciding the type with BFS tree's profile.

Tested graphs	Estimate with complete levels	Estimate with upper k levels
Poisson 3	Poisson 3	Poisson 3
Poisson 10	Poisson 10	Poisson 10
Power-law 2.1	Power-law 2.1	Power-law 2.1
Power-law 2.3	Power-law 2.3	Power-law 2.3

tree or with a bounded BFS tree is correct regarding to the underlying type. In most cases, the number of links (there is a bijection from a parameter to m, if n is fix.)is also well estimated. Only in the case of Poisson 10, there is a few times that the best estimate is located at Poisson 11. This error comes probably from the difference of the degree of the root of BFS tree.

Table 7.4: Manhattan Distance (MD) for a Poisson 3 graph.

Poisson theoretical profile			Power-law theoretical profile		
	All levels	First 6 levels		All levels	First 6 levels
λ	MD	MD	λ	MD	MD
3	**19584**	**293**	2.05	188630	89676.3
4	112060	708	2.1	185801	87127.8
5	161899	3163	2.15	182312	84089.9
6	172783	8177	2.2	178671	80499.2
7	188045	16886	2.25	175098	76286.1
8	189213	29893	2.3	171005	71378.9
9	193320	46472	2.35	166298	65716.6
10	195874	64176	2.4	160850	59270.6
11	196144	79619	2.45	154513	52078.5
12	196168	90320	2.5	147278	44283.2

7.3.2 Real-world graphs

In this example, we apply our methods on some real-world graphs: Skitter graphs and Radar graphs. Here we observe quite different results, in Table 7.5 to that of random model graphs. Radar-enix is more likely to be a Poisson 4 graph, in both cases, with a complete BFS tree or a bounded BFS tree. In the case of Radar-cm, the complete profile shows that it is probable to be a Poisson 3 graph; while the bounded profile give a result between Poisson 3 and Poisson 4. In the case of Radar-japon and Skitter-AS, the results are even a little strange. Different type

Table 7.5: Result of deciding the type of real-world graphs with BFS tree's profile.

Tested graphs	Complete BFS	Bounded BFS
Radar-cm	Poisson 3	Poisson 3 or Poisson 4
Radar-enix	Poisson 4	Poisson 4
Radar-japon	Poisson 6 or Power-law 2.5	Poisson 4 or Power-law 2.5
Skitter-AS	Poisson 18 or Power-law 2.1	Power-law 2.1
Skitter-Router	Power-law 2.03	Power-law 2.03

is determined, when the BFS process running from different root. So we conclude that the type of the graph Radar-japon and Skitter-AS is not significant with this methodology.

7.4 Conclusion

Applying Theorem 8 and Theorem 9 in Chapter 6, we develop two new strategies to decide on the type and to estimate the number of links.

RRIL and PPIL take the place of RR and PP strategies and consider the information of the number of invisible links at each level. Then the random adding links procedure is conducted more precisely and we observe indeed a better estimation for random model graphs.

NV algorithm is a first trial that we use only a bounded BFS tree, which based on the number of nodes at each level of BFS tree.

However these two methodologies have a strong drawback. They are too sensitive to the selection of the root. From one root to another, the estimation of the number of links, even the type may be changed.

Chapter 8

Conclusion

Contents

8.1 Summary

In this thesis, we have explored the possibility to deduce type of the degree distribution of a graph with traceroute-like measurements modeled by BFS trees. Indeed, it is known that observing directly this property on actual measurements gives a biased results. We therefore developed an orthogonal approach consisting in modeling the measurement process and inferring from its properties the ones of the underlying graph, in particular the type of its degree distribution. More precisely, we focus on distinguishing between homogeneous degree distributions (modeled by Poisson laws) and heterogeneous ones (modeled by power-laws), which is currently at the core of an important controversy in the area.

In Chapter 3 we assume that we know the number of nodes n, the number of links m and a BFS tree T of an unknown graph G. We then introduce algorithm $SB_m(RR, PP)$ relying on the PP and RR strategies to build a graph from T and decide on the type of the degree distribution of G. We validate this approach with models graphs, thus in cases where G is known. This shows that our method succeeds in deciding on the type of the degree distribution.

In Chapter 4, we improve this method: we get rid of the requirement of knowing m by scanning large ranges of possible values of this parameter and selecting the one which gives the best results. The obtained algorithm, $SB(RR, PP)$, still leads to valid conclusions and provides an estimate of m in addition to the type of the underlying degree distribution.

We explore in Chapter 5 the improvement obtained by using *several* BFS trees rather than just one, a situation often met in practice. The obtained algorithm, $MB(RR, PP)$, improves significantly the estimate of m previously obtained.

These first chapters only use limited information from the BFS tree; we explore in Chapters 6 and 7 the possibility to use more detailed information, namely the *profile* of the BFS tree, *i.e.* the number of nodes at each level. We first study the formal properties of such profiles in Chapter 6 using generating functions and queuing models. We apply obtained results in Chapter 7, where we develop the RRIL and PPIL strategies to reconstruct a graph from a BFS and where we explore the possibility to decide from a *bounded* BFS (*i.e.* only the first levels of the BFS, which is much closer to real-world measurements). This leads to the $SB(RRIL, PPIL)$ algorithm for deciding on the degree distribution of a graph, which improves further previous results, and to the NV algorithm which succeeds in deciding on the type of the degree distribution of a graph from very limited, and rather realistic, information.

All these results are summarized in Table 8.1, leading to the following comments on the key features of our algorithms.

Deciding Type. All five algorithms work well for deciding the type, either Poisson or power-law, of a graph.

Table 8.1: Comparison of algorithms: Y for Yes; N for No; 1 for the best and 5 for the worst.

	$SB_m(RR, PP)$	$SB(RR, PP)$	$MB(RR, PP)$	$SB(RRIL, PPIL)$	NV
Estimate m?	N	Y	Y	Y	Y
Exactness		4	2	3	1
Bounded BFS?	N	N	N	N	Y
Speed	2	4	5	3	1
Stable?	Y	Y	Y	N	N

Estimate m. Except $SB_m(RR, PP)$, all other algorithms can estimate the number of links m. $SB(RR, PP)$ always overestimates m for Poisson model graphs and underestimate m for power-law model graphs. $SB(RRIL, PPIL)$ underestimate for Poisson and overestimate for power-law, but the difference is much smaller than $SB(RR, PP)$. $MB(RR, PP)$ gives an excellent estimate when enough BFS trees are used. NV is the best algorithm to estimate m: for all model graphs, it perfectly fits with the original one(same type and same m).

Bounded BFS. As a complete BFS tree is a too strong hypothesis in practice, NV is an exciting algorithm which can decide on the type and estimate the number of links according to the first levels of a BFS tree.

Speed. In practice, we have to deal with huge graphs with typically one million nodes or even more. Hence the speed of the algorithm is important. The first four algorithms rebuild a graph during the process of deciding and they must track the degree of nodes. Usually graphs that have more than 100000 nodes are too heavy for these algorithms, but NV can readily handle such large graphs.

Stability. Although NV is very appealing regarding other feature, it is sensitive to the choice of root, which is a problem in practice.

8.2 Perspectives

Our work is only a first step; it calls for several improvements in the future. We group them in three categories: using less hypotheses, refining rebuilding strategies and establishing methodology on more general graphs.

8.2.1 Using less hypotheses

The main limitation of our contribution is that we suppose the knowledge of a complete BFS of the unknown graph in all *stable* algorithms. This hypothesis is

not realistic, though, since practical measurements rather provide only paths to a subset of the real graph nodes. Our main perspective therefore is to reduce requirements on data, and design strategies needing bounded BFS (BFS until a certain level) only, and partial BFS (that contain paths to a subset of all nodes of the graph). Assuming that such measurements are available is much more realistic [OML08].

NV is a first proposal using a bounded BFS tree, but it lacks the stability on some real world graphs. One possible improvement is to introduce some additional information to refine the theoretical computation. For example, in *NV* we do not consider the degree of root node in the computation. If we did, the vector would be $(1, r, n_2, n_3...)$ where r is the root degree instead of $(1, n_1, n_2, n_3...)$ where n_1 is always the average degree. The vector with information of root counteract the impact of the randomness of the selection of root, which is very meaningful to resolve the problem of stability. Then, we have to develop a new theorem to compute a vector by considering root degree. Furthermore, if we know the exact topology around the root, such as first three or four levels, this method can be developed to more levels. However, the corresponding computation is not trivial.

If we know only a partial BFS tree, the corresponding estimate becomes more complicated. In this case, the number of nodes is no longer known. A new methodology, which estimates the total number of nodes or even the number of nodes at each level, is necessary.

8.2.2 Refining rebuilding strategies

In future work, we also want to improve our strategies by investigating various refinements.

Additional properties. We first may take into account more subtle properties than the BFS, such as the clustering coefficient and the average distance.

Other distributions. We hope to extend the rebuilding strategies to other types of distributions. In fact, few empirical phenomena obey power-law for all degree. More often the power-law applies only for values greater than some minimum degree. A first step would be to mix RR and PP strategies for mixed Poisson/power-law graphs. The obtained RP strategy would give a mixed Poisson power-law distribution, but still lacks a parameter to indicate the ratio of the impact of Poisson or power-law. It is not difficult to develop the corresponding strategies to rebuild a *regular* graph, but for other distributions, such as power-law with exponential cut-off, the procedure seems nontrivial. Furthermore, how to rebuild a graph, which may have an arbitrary degree sequence, from a BFS tree is an open problem.

8.2.3 Establishing methodology on more general graphs

In this thesis, we conducted experiments on two kinds of graphs: random model graphs (from the configuration model) and real-world graphs. Besides configuration model graphs, some other graphs are also used to model the Internet, like BA graphs [BAJ99]. All our analysis and experiments do not suggest a trivial way to manage these models. The profile of BFS tree of BA graphs is surely different from the one of configuration model graphs. From a statistical viewpoint, we may introduce some additional constraints to smooth the exception to randomness.

Bibliography

[AB02] Réka Albert and Albert L. Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47–97, 2002.

[AJB99] R. Albert, H. Jeong, and A. L. Barabasi. The diameter of the world wide web. *Nature*, 401:130–131, 1999.

[AJB00] Reka Albert, Hawoong Jeong, and Albert-Laszlo Barabasi. Error and attack tolerance of complex networks. *Nature*, 406(6794):378–382, 2000.

[AKCM05] Dimitris Achlioptas, David Kempe, Aaron Clauset, and Cristopher Moore. On the bias of traceroute sampling: or, power-law degree distributions in regular graphs. *In ACM Symposium on Theory of Computing*, pages 694–703, 2005.

[Alb05] Réka Albert. Scale-free networks in cell biology. *Journal of Cell Science*, 118(21):4947–4957, 2005.

[ALWD05] David Alderson, Lun Li, Walter Willinger, and John C. Doyle. Understanding internet topology: principles, models, and validation. *IEEE/ACM Transactions on Networking*, 13(6):1205–1218, 2005.

[BA99] A. L. Barabasi and R. Albert. Emergence of Scaling in Random Networks. *Science*, 286(5439):509–512, 1999.

[BAJ99] A. L. Barabási, Reka Albert, and Hawoong Jeong. Mean-field theory for scale-free random networks. *Physica A: Statistical Mechanics and its Applications*, 272(1-2):173–187, 1999.

[Bas89] M. Basseville. Distance measures for signal processing and pattern recognition. *Signal Process*, 18:349–369, 1989.

[BBCS05] Noam Berger, Christian Borgs, Jennifer T. Chayes, and Amin Saberi. On the spread of viruses on the internet. *In Proceedings of the 16th*

annual ACM-SIAM symposium on Discrete algorithms, pages 301–310, 2005.

[BGLL08] Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008+, 2008.

[Bir08] Maria Biryukov. Co-author Network Analysis in DBLP: Classifying Personal Names. *Modelling, Computation and Optimization in Information Systems and Management Sciences*, 14:399–408, 2008.

[Bol84] Béla Bollobás. The evolution of random graphs. *Transactions of the American Mathematical Society*, 284:257–274, 1984.

[Bol01] Béla Bollobás. *Random Graphs*. Cambridge University Press, 2 edition, 2001.

[BR04] Bélaa Bollobás and Oliver Riordan. The Diameter of a Scale-Free Random Graph. *Combinatorica*, 24(1):5–34, 2004.

[Bra01] Ulrik Brandes. A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology*, 25:163–177, 2001.

[Bro00] A. Broder. Graph structure in the web. *Computer Networks*, 33(1-6):309–320, 2000.

[BV04a] P. Boldi and S. Vigna. The WebGraph framework i: Compression techniques. *In Proceedings of the 13th International World Wide Web Conference*, pages 595–601, 2004.

[BV04b] Paolo Boldi and Sebastiano Vigna. The WebGraph framework II: Codes for the World-Wide web. *Data Compression Conference*, 0:528+, 2004.

[CEAH00] Reuven Cohen, Keren Erez, Daniel B. Avraham, and Shlomo Havlin. Resilience of the internet to random breakdowns. *Physical Review Letters*, 85(21):4626–4628, 2000.

[CGW07] Reuven Cohen, Mira Gonen, and Avishai Wool. Bounding the bias of tree-like sampling in ip topologies. *European Conference on Complex Systems*, 2007.

[CHK+09] Kimberly Claffy, Young Hyun, Ken Keys, Marina Fomenkov, and Dmitri Krioukov. Internet Mapping: From Art to Science. *Cybersecurity Applications and Technology Conference for Homeland Security*, pages 205–211, 2009.

[CL04] Fan Chung and Linyuan Lu. The Average Distance in a Random Graph with Given Expected Degrees. *Internet Mathematics*, 1(1):91–113, 2004.

[CLRS09] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. The MIT Press, 3 edition, 2009.

[CM04] Aaron Clauset and Cristopher Moore. Why mapping the internet is hard. *Disordered Systems and Neural Networks*, 2004.

[Cod89] Earl A. Coddington. *An Introduction to Ordinary Differential Equations*. Dover Publications, 1989.

[CSN09] Aaron Clauset, Cosma R. Shalizi, and M. E. J. Newman. Power-law distributions in empirical data. *Society for Industrial and Applied Mathematics Review*, 51(4):661+, 2009.

[CSWN00] D. Callaway, S. Strogatz, D. Watts, and M. E. J. Newman. Network robustness and fragility: Percolation on random graphs. *Physical Review Letters*, 85(25):5468–5471, 2000.

[DAHB+06] Luca Dall'asta, Ignacio Alvarez-Hamelin, Alain Barrat, Alexei Vazquez, and Alessandro Vespignani. Exploring networks with traceroute-like probes: Theory and simulations. *Theoretical Computer Science*, 355(1):6–24, 2006.

[DF07] B. Donnet and T. Friedman. Internet topology discovery: A survey. *Communications Surveys & Tutorials, IEEE*, 9(4):56–69, 2007.

[DGM08] S. N. Dorogovtsev, A. V. Goltsev, and J. F. F. Mendes. Critical phenomena in complex networks. *Reviews of Modern Physics*, 80(4):1275–1335, 2008.

[ED06] M. Deza Elena Deza. *Dictionary of distances*. Elsevier Science, 2006.

[EDJ+08] W. T. Eadie, D. Drijard, F. E. James, M. Roos, and B. Sadoulet. *Statistical methods in experimental physics*. American Elsevier Publishing Co, 2008.

[Edw01] Harold M. Edwards. *Riemann's Zeta Function*. Dover Publications, 2001.

[EHP00] Merran Evans, Nicholas Hastings, and Brian Peacock. *Statistical Distributions*. Wiley-Interscience, 2000.

[ER60] P. Erdos and A. Renyi. On the evolution of random graphs. *Publication of the Mathematical Institute of the Hungarian Academy of Sciences*, 5:17–61, 1960.

[FDP06] Philippe Flajolet, Philippe Dumas, and Vincent Puyhaubert. Some exactly solvable models of urn process theory. *Discrete Mathematics and Theoretical Computer Science*, pages 59–118, 2006.

[FFF99] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the Internet topology. *SIGCOMM: Conference on Applications, technologies, architectures, and protocols for computer communication*, 29(4):251–262, 1999.

[FGP05] Philippe Flajolet, Joaquim Gabarró, and Helmut Pekari. Analytic urns. *Annals of Probability*, 33(3), 2005.

[FKP02] Alex Fabrikant, Elias Koutsoupias, and Christos Papadimitriou. Heuristically Optimized Trade-Offs: A New Paradigm for Power Laws in the Internet. *Automata, Languages and Programming*, 2380:781, 2002.

[Fre77] Linton C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40(1):35–41, 1977.

[FS09] Philippe Flajolet and Robert Sedgewick. *Analytic Combinatorics*. Cambridge University Press, 2009.

[GL05] J. L. Guillaume and M. Latapy. Relevance of massively distributed explorations of the internet topology: simulation results. *In INFOCOM: 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, 2:1084–1094 vol. 2, 2005.

[GLM06] J. Guillaume, M. Latapy, and D. Magoni. Relevance of massively distributed explorations of the internet topology: Qualitative results. *Computer Networks*, 50(16):3197–3224, 2006.

[GMZ03] C. Gkantsidis, M. Mihail, and E. Zegura. Spectral analysis of internet topologies. 1:364–374, 2003.

[HD03] R. Hill and R. Dunbar. Social network size in humans. *Human Nature*, 14(1):53–72, 2003.

[HKLFM04] S. B. Handurukande, A. M. Kermarrec, F. Le Fessant, and L. Massoulié. Exploiting semantic clustering in the eDonkey P2P network. *In Proceedings of the 11th workshop on ACM SIGOPS European workshop*, 2004.

[HKP07] Hsien-Kuei Hwang, Markus Kuba, and Alois Panholzer. Diminishing urn models: analysis of exactly solvable models. *In Proceedings of the 19th International Conference on Formal Power Series and Algebraic Combinatorics*, 2007.

[HPG+08] John Heidemann, Yuri Pradkin, Ramesh Govindan, Christos Papadopoulos, Genevieve Bartlett, and Joseph Bannister. Census and Survey of the Visible Internet. *In Proceedings of the 8th ACM SIGCOMM conference on Internet measurement*, pages 169–182, 2008.

[Hua06] Zan Huan. Link prediction based on graph topology: The predictive value of the generalized clustering coefficient. *In Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (LinkKDD2006)*, 2006.

[Ivi03] Aleksandar Ivic. *The Riemann Zeta-Function: Theory and Applications*. Dover Publications, 2003.

[JK77] Norman L. Johnson and Samuel Kotz. *Urn models and their application: An approach to modern discrete probability theory*. John Wiley & Sons Inc, 1977.

[JTA+00] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and A. L. Barabasi. The large-scale organization of metabolic networks. *Nature*, 407(6804):651–654, 2000.

[KCC+07] Dmitri Krioukov, Fan Chung, Kc Claffy, Marina Fomenkov, Alessandro Vespignani, and Walter Willinger. The workshop on internet topology (wit) report. *SIGCOMM Computer Communication Review*, 37:69–73, 2007.

[KKR+99] Jon M. Kleinberg, Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew S. Tomkins. The web as a graph: Measurements, models, and methods. *Computing and Combinatorics: 5th Annual International Conference, COCOON'99*, pages 1+, 1999.

[KW08] Balachander Krishnamurthy and Walter Willinger. What are our standards for validation of measurement-based networking research? *ACM SIGMETRICS Performance Evaluation Review*, 36:64–69, 2008.

[Lat07] Matthieu Latapy. Grands graphes de terrain mesure et métrologie, analyse, modélisation, algorithmique. *Habilitation à Diriger des Recherches, Paris 6*, 2007.

[LAWD04] Lun Li, David Alderson, Walter Willinger, and John Doyle. A first-principles approach to understanding the internet's router-level topology. *SIGCOMM: Conference on Applications, technologies, architectures, and protocols for computer communications*, 34(4):3–14, 2004.

[LBCX03] A. Lakhina, J. W. Byers, M. Crovella, and P. Xie. Sampling biases in IP topology measurements. *In INFOCOM: 22th Conference on Computer and Communications Societies*, 1:332–341, 2003.

[LBGL05] Stevens Le Blond, Jean-loup Guillaume, and Matthieu Latapy. Clustering in p2p exchanges and consequences on performances. *In International Workshop on Peer-to-Peer Systems*, 3640, 2005.

[LBLG04] S. Le Blond, M. Latapy, and J. L. Guillaume. Statistical analysis of a P2P query graph based on degrees and their time evolution. 2004.

[LENA+01] Fredrik Liljeros, Christofer R. Edling, Luis A. Nunes Amaral, H. Eugene Stanley, and Yvonne Aberg. The web of human sexual contacts. *Nature*, 2001.

[LM08] M. Latapy and C. Magnien. Complex Network Measurements: Estimating the Relevance of Observed Properties. *INFOCOM: 27th Conference on Computer Communications*, pages 1660–1668, 2008.

[MLG09] Clemence Magnien, Matthieu Latapy, and Jean-Loup Guillaume. Impact of Random Failures and Attacks on Poisson and Power-Law Random Networks. *ACM Computing Surveys*, 2009.

[MP01a] Damien Magoni and Jean J. Pansiot. Analysis of the autonomous system network topology. *SIGCOMM Computer Communication Review*, 31:26–37, 2001.

[MP01b] Damien Magoni and Jean-Jacques Pansiot. Influence of Network Topology on Protocol Simulation. *International Conference on Networking*, 2093:762–770, 2001.

[MR95] Michael Molloy and Bruce Reed. A critical point for random graphs with a given degree sequence. *Random Structures and Algorithms*, 6(2-3):161–180, 1995.

[MR98] Michael Molloy and Bruce Reed. The size of the giant component of a random graph with a given degree sequence. *Combinatorics Probability and Computing*, 7:295–305, 1998.

[MRWK03] Zhuoqing M. Mao, Jennifer Rexford, Jia Wang, and Randy H. Katz. Towards an accurate AS-level traceroute tool. *In Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 365–378, 2003.

[NSW01] M. E. Newman, S. H. Strogatz, and D. J. Watts. Random graphs with arbitrary degree distributions and their applications. *Physical Review E Statistical Nonlinear Soft Matter Physics*, 2001.

[NWS02] M. E. J. Newman, D. J. Watts, and S. H. Strogatz. Random graph models of social networks. *In Proceedings of the National Academy of Sciences of the United States of America*, 99(Suppl 1):2566–2572, 2002.

[OML08] Frédéric Ouédraogo, Clémence Magnien, and Matthieu Latapy. A radar for the internet. *1st International Workshop on Analysis of Dynamic Networks*, 2008.

[PNFB05] Alex Potanin, James Noble, Marcus Frean, and Robert Biddle. Scale-free geometry in OO programs. *Communication ACM*, 48(5):99–103, 2005.

[Pol01] Andrei D. Polyanin. *Handbook of First-Order Partial Differential Equations*. CRC Press, 2001.

[PR04] Thomas Petermann and Paolo De Los Rios. Exploration of scale-free networks. *The European Physical Journal B*, 38:201, 2004.

[RsA04] R.Pastor-satorras and A.Vespignani. *Evolution and Structure of the Internet: A Statistical Physics Approach*. Cambridge University Press, 2004.

[SBH02] Berend Snel, Peer Bork, and Martijn A. Huynen. The identification of functional modules from the genomic association of genes. *Proceedings of the National Academy of Sciences*, 99(9):5890–5895, 2002.

[SF95] Robert Sedgewick and Philippe Flajolet. *An Introduction to the Analysis of Algorithms*. Addison Wesley, 1995.

[SS05] Yuval Shavitt and Eran Shir. DIMES: let the internet measure itself. *SIGCOMM Computer Communication Review*, 35(5):71–74, 2005.

[Str92] Walter A. Strauss. *Partial Differential Equations: An Introduction*. Wiley, 1992.

[S.W93] Herbert S.Wilf. *Generatingfunctionology*. Academic Press, 2 edition, 1993.

[VBD+07] Fabien Viger, Alain Barrat, Luca Dall'Asta, Cun H. Zhang, and Eric D. Kolaczyk. What is the real size of a sampled network? The case of the Internet. *Physical Review E*, 75(5):056111+, 2007.

[VKMVS04] Spyros Voulgaris, Anne-Marie Kermarrec, Laurent Massoulié, and Maarten Van Steen. Exploiting semantic proximity in Peer-to-Peer content searching. *In Proceedings of the 10th International Workshop on Future Trends in Distributed Computing Systems, Suzhu*, 2004.

[VL05] Fabien Viger and Matthieu Latapy. Efficient and simple generation of random simple connected graphs with prescribed degree sequence. *In Proceedings of the 11th International Computing and Combinatorics Conference*, 3595:440–449, 2005.

[WAD09] Walter Willinger, David Alderson, and John C. Doyle. Mathematics and the Internet: A Source of Enormous Confusion and Great Potential. *Notices of the American Mathematical Society*, 56, 2009.

[WF94] S. Wasserman and K. Faust. *Social network analysis*. Cambridge University Press, 1994.

[WS98] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, 1998.

Related Resources

[1] Event based random number, (see http://www.mathworks.fr).

[2] Graph generation software,
(see http://www-rp.lip6.fr/~latapy/FV/generation.html).

[3] Internet mapping project at Lucent Bell Labs.
(see http://cs.bell-labs.com/who/ches/map).

[4] Paris traceroute project, (see http://www.paris-traceroute.net).

[5] Pathping, (see http://ss64.com/nt/pathping.html).

[6] Radar project, (see http://data.complexnetworks.fr/Radar/).

[7] RFC. (see http://tools.ietf.org/html/rfc1393).

[8] SCAN project at the Information Science Institute.
(see http://www.isi.edu/div7/scan).

[9] The Cooperative Association for Internet Data Analysis (CAIDA), located at the San Diego Supercomputer Center.
(see http://www.caida.org/data/overview/).

[10] The National Laboratory for Applied Network Research (NLANR), sponsored by the National Science Foundation. (see http://moat.nlanr.net).

[11] Topology project, Electric Engineering and Computer Science Department, University of Michigan. (see http://topology.eecsd.umich.edu).

[12] Tracepath, (see http://linux.die.net/man/8/tracepath).

[13] Traceroute6, (see http://linux.die.net/man/8/traceroute6).

[14] Tracert, (see http://tracert.com/).

[15] Webgraph project, (see http://webgraph.dsi.unimi.it).

List of Figures

List of Tables

List of Algorithms