Finite-Size Effects in the Dependency Networks of Free and Open-Source Software

Rajiv Nair
Tata Institute of Social Sciences
V. N. Purav Marg, Deonar, Mumbai 400088, India
rajiv@tiss.edu

G. Nagarjuna
Homi Bhabha Centre for Science Education
Tata Institute of Fundamental Research
V. N. Purav Marg, Mankhurd, Mumbai 400088, India
nagarjun@gnowledge.org

Arnab K. Ray
Department of Physics, Jaypee University of Engineering and Technology
Raghogarh, Guna 473226, Madhya Pradesh, India
arnab.kumar@juet.ac.in

We propose a continuum model for the degree distribution of directed networks in free and open-source software. The degree distributions of links in both the in-directed and out-directed dependency networks follow Zipf's law for the intermediate nodes, but the heavily linked nodes and the poorly linked nodes deviate from this trend and exhibit finite-size effects. The finite-size parameters make a quantitative distinction between the in-directed and out-directed networks. For the out-degree distribution, the initial condition for a dynamic evolution corresponds to the limiting count of the most heavily linked nodes that the out-directed network can finally have. The number of nodes contributing out-directed links grows with every generation of software release, but this growth ultimately saturates toward a terminal value, due to the finiteness of semantic possibilities in the network.

1. Introduction

Scale-free distributions in complex networks [1–9] span across diverse domains like the World Wide Web [1, 10] and the internet [1], social, ecological, biological, and linguistic networks [6], trade and business networks [11], and syntactic and semantic networks [12–14]. Scale-free features have also been discovered in electronic circuits [15] and in the architecture of computer software [16]. The structure of object-

oriented software is a heterogeneous network, characterized by a power-law distribution [17], and it is on the basis of scale-free networks that software fragility is explained [18]. Power-law features exist in the inter-package dependency networks in free and open-source software (FOSS) [19], and studies have shown that modifications in this type of software network also follow a power-law decay in time [20, 21].

Continuing on the software theme, while installing a software package from the Debian GNU/Linux distribution, many other packages, known as the "dependencies," are needed as prerequisites. This leads to a dependency-based network among all the packages, with each of these packages being a node in a network of dependency relationships. Each dependency relationship connecting any two packages (nodes) is a link (an edge), and every link establishes a relation between a prior package and a posterior package, whereby the functions defined in the prior package are invoked in the posterior package. So what emerges is a semantic network with a directed flow of meaning, determined by the direction of the links.

Semantic networks are a subject of major interest, especially where small-world structures [22] and scale-free aspects [6] of networks are concerned [14]. With particular regard to component-based software, a semantic relationship among components underlies the network of what are known as strong dependencies [23]. The components of a FOSS network are interconnected by various relationships (including a negative one, "conflicts"), and only one of these is based on the field "depends." This again is further categorized into the two cases of strong dependencies and direct dependencies, with a correlation between the two cases [23]. As regards direct dependencies, the scale-free character of the Debian GNU/Linux distribution has been studied [19, 24].

In our study, the semantic network of nodes in the Debian distribution is founded on one single principle running through all the nodes: Y depends on X; its inverse: X is required for Y. The semantic network so formed is a straightforward dependency-based directed network only. Considering any particular node in such a directed network, its links (the relations with other nodes) are of two types, incoming links and outgoing links, as a result of which there will arise two distinct types of directed networks [6]. For the network of incoming links in the Etch release of Debian, one study [24] has empirically tested Zipf's law in the GNU/Linux distribution. This is a phenomenon discovered originally in the occurrence frequency of words in natural languages [25] that has over the years emerged widely in many other areas.

Our work affirms the existence of Zipf's law as a universal feature underlying the FOSS network. Here, in fact, both the networks of in-

coming and outgoing links follow Zipf's law. However, simple power-law properties do not suffice to provide a complete global model for directed networks. For any system with a finite size, the power-law trend is not manifested indefinitely [26, 27], and for a FOSS network, this matter awaits a thorough investigation [24]. Deviations from the power-law trend appear for both the profusely linked and the sparsely linked nodes. The former case corresponds to the distribution of a disproportionately high number of links connected to a very few important nodes (the so-called "hubs" or rich nodes/top nodes). The particular properties of all these outlying nodes, as well as any distinguishing characteristic of the two directed networks, can only be known by studying the finite-size effects (equivalently the saturation properties) in the respective networks [28] and by understanding how these effects are related to the semantic structure in the network. These are the principal objects of our investigation.

2. A Nonlinear Continuum Model

The main advantage of the Debian GNU/Linux distribution is that it is the largest component-based system that can be accessed freely for study [23]. The mathematical modeling of this FOSS network has been carried out here primarily with the help of data collected from the two stable Debian releases, Etch (Debian GNU/Linux 4.0) and Lenny (Debian GNU/Linux 5.0), available at http://www.debian.org/releases. The networks of both the incoming links and the outgoing links span about 18 000 packages (nodes) in the Etch release, while in the Lenny release, the corresponding number of packages is about 23 000. For this work, the chosen computer architecture supported by both the releases is AMD64. The dynamic features of the model have further been grounded on the first three generations of Debian releases, that is, Buzz (Debian GNU/Linux 1.1), Hamm (Debian GNU/Linux 2.0), and Woody (Debian GNU/Linux 3.0), all of which are supported by the architecture i386. The model founded with the help of the Etch and Lenny releases shows a retrospective compatibility with the earlier releases, and moving forward in time, it is also in consonance with the features shown by the latest stable Debian release, Squeeze (Debian GNU/Linux 6.0), which is again based on the AMD64 architecture. The graphical results presented in this paper are based mostly on the three latest releases, Etch, Lenny, and Squeeze. All of these releases have a substantial number of nodes and links, and even though these numbers are to be counted only discretely, the largeness of their total count allows a continuum description to be adopted, using a differential equation.

For developing the model, we need to count the actual number of software packages ϕ that are connected by a particular number of links x in either kind of network. This gives an unnormalized frequency distribution of $\phi \equiv \phi(x)$ versus x. Normalizing this distribution in terms of the relative frequency distribution of the occurrence of packages would have yielded the usual probability density function. To provide a continuum model for any power-law feature in this frequency distribution, we posit a nonlinear logistic-type equation,

$$(x+\lambda)\,\frac{\mathrm{d}\,\phi}{\mathrm{d}\,x} = \alpha\phi(1-\eta\phi^{\mu}), \tag{1}$$

in which α is a power-law exponent, μ is a nonlinear saturation exponent, η is a "tuning" parameter for nonlinearity, and λ is another parameter that is instrumental in setting a limiting scale for the poorly connected nodes. The motivation behind this mathematical prescription can be easily followed by noting that when $\eta = \lambda = 0$, there will be a globally valid power-law distribution. However, when the distribution is finite, the power-law trend fails to hold true beyond intermediate scales of x. Such deviations from a full power-law behavior are especially prominent for high values of x (related to the very heavily connected nodes), and therefore, it can be argued that finiteness in the distribution is closely related to its saturation. This type of saturation behavior is frequently modeled by a nonlinear logistic equation [29, 30], and so, to understand the saturation properties of the highly connected nodes in the Debian network, it will be necessary to understand the part played by nonlinearity.

Integration of equation (1), which is a nonlinear differential equation, is done by making suitable substitutions on ϕ^{μ} and $x+\lambda$, followed by the application of partial fractions. After that we get the integral solution of equation (1) as (for $\mu \neq 0$)

$$\phi(x) = \left[\eta + \left(\frac{x+\lambda}{c}\right)^{-\mu\alpha}\right]^{-1/\mu}, \tag{2}$$

where c is an integration constant. Evidently, when $\eta = \lambda = 0$ (with the former condition implying the absence of nonlinearity), there will be a global power-law distribution, going as $\phi(x) = (x/c)^{\alpha}$, regardless of any nonzero value of μ. The situation becomes quite different, however, when both η and λ have nonzero values. In this situation, the network will exhibit a saturation behavior on extreme scales of x (both low and high). For high values of x, this can be easily appreciated from equation (1) itself, from which the limiting value of ϕ is obtained as $\phi = \eta^{-1/\mu}$.

3. Model Fitting of the Free and Open-Source Software Network

The parameters α, μ, η, λ, and c in the solution given by equation (2) can now be fixed by the distribution of links and nodes in the Debian repository. In Figure 1 the degree distribution of incoming links in the Etch release is plotted. The dotted straight line in this log-log plot indicates a purely power-law behavior. While this gives a satisfactory description of the distribution on intermediate scales of x, there is a clear departure from the power law both as $x \longrightarrow 0$ and $x \longrightarrow \infty$. The solution given by equation (2) fits the power law, as well as the departure from it, at both the small-connectivity and the high-connectivity ends. Among all the parameters, the values of α and μ remain unchanged while modeling the degree distribution of outgoing links, as shown in the plot in Figure 2. The obvious implication of $\alpha = -2$ in both the cases is that Zipf's law universally underlies the frequency distribution of the intermediate nodes and links in both kinds of networks. The only quantitative measures to distinguish between the two networks are the values of η, λ, and c.

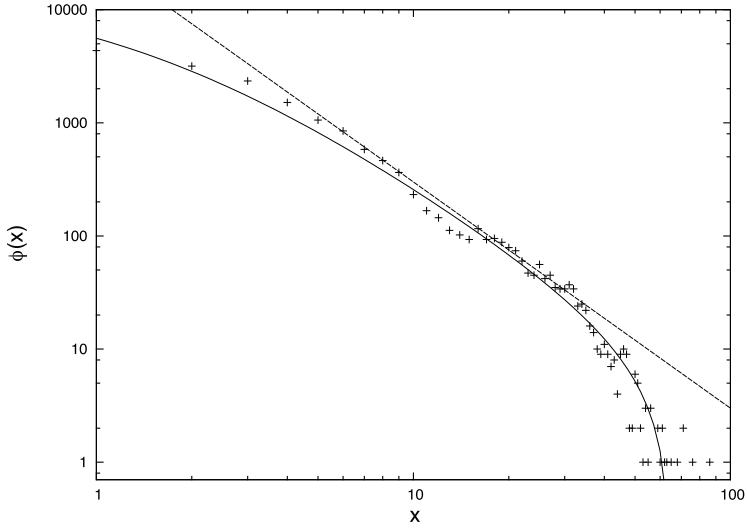

Figure 1. For the network of incoming links in the Etch release, the degree distribution shows a good fit in the intermediate region with a power-law exponent $\alpha = -2$ (as indicated by the dotted straight line), which validates Zipf's law. However, for large values of x, there is a saturation behavior toward a limiting scale that is modeled well with the parameter $\eta = -8$. When x is small, the fit is good for $\lambda = 1.5$. The global fit becomes possible only when $\mu = -1$, which turns out to be a universally valid number. For this specific plot, the data is fitted by $c \simeq 190$.

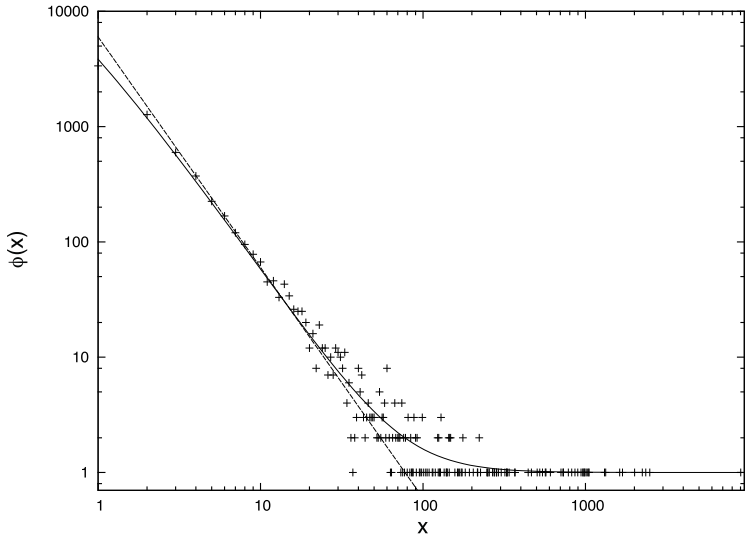

Figure 2. For the network of outgoing links in the Etch release, the degree distribution of intermediate nodes is again modeled well by a power-law exponent $\alpha = -2$ which is Zipf's law (as the dotted straight line shows). However, the saturation behavior of the top nodes is different from that of the network of incoming links. There is a clear convergence of ϕ toward a limit given by $\eta = 1$ (with μ remaining unchanged at -1). For the poorly linked nodes, the convergence is attained for $\lambda = 0.25$. Thus, when α and μ remain the same, the value and the sign of η, as well as the value of λ, distinguish the type of a dependency network. The data is fitted for $c \simeq 80$. A solitary top node is to be seen for $x = 9025$.

Similarly, data from the Lenny release has been plotted in Figures 3 and 4. The former plot gives the in-degree distribution of the nodes, while the latter gives the out-degree distribution. The values of η, λ, and c in the in-degree distribution of the Lenny release change with respect to the previous release, Etch. With changing values of these particular parameters, the saturation properties in the in-degree distribution, therefore, undergo a significant quantitative change at the highly connected end. In contrast, for the out-degree distribution, the changes across a new generation of Debian release are fitted by varying the values of λ and c. The fact that η remains the same as before while λ changes implies that the saturation properties remain unchanged at the richly linked end, but change at the poorly connected end. Changes in the value of c for a particular degree distribution cause a translation of the model curve in the x–ϕ plane. And, as Figures 1 through 4 indicate, Zipf's law prevails in all the cases with $\alpha = -2$.

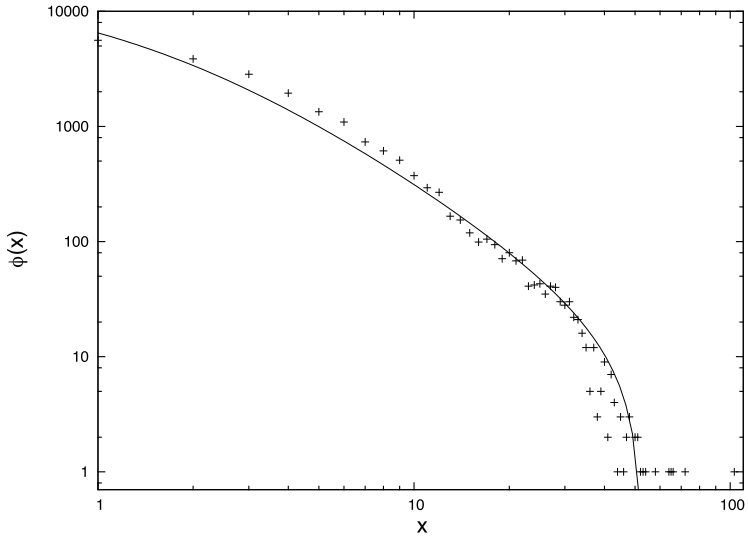

Figure 3. For the network of incoming links in the Lenny release, the intermediate nodes (fitted with a power-law exponent $\alpha = -2$) uphold Zipf's law once again. For large values of x, however, the saturation behavior toward a limiting scale of ϕ is modeled by the value $\eta = -15$. When x is small, the fit is good for $\lambda = 1.6$. Once again $\mu = -1$, but for this particular plot, $c \simeq 210$. The richly linked nodes here are less connected than they are in the case of the Etch release.

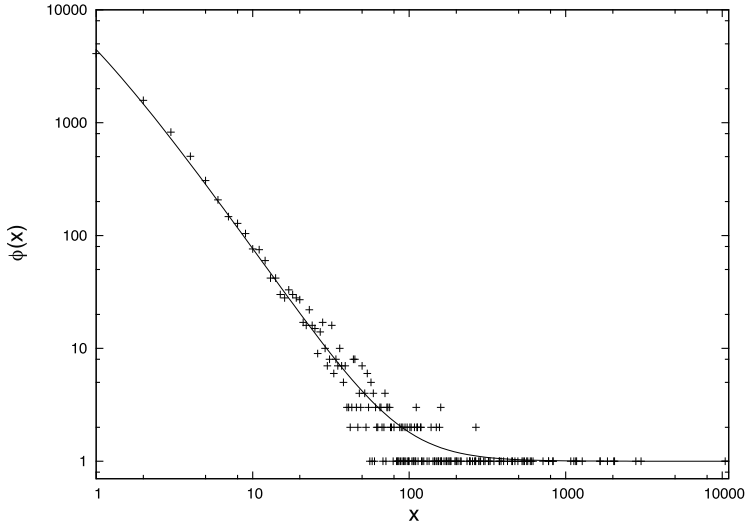

Figure 4. For the network of outgoing links in the Lenny release, the distribution of intermediate nodes obeys Zipf's law, as the power-law exponent

$\alpha = -2$ shows. The saturation behavior of the top nodes remains the same as it is for the Etch data. The convergence of ϕ toward a limit set by $\eta = 1$ is evident, with $\mu = -1$ as before. For the poorly linked nodes, the convergence is given by $\lambda = 0.35$. The other value that distinguishes the out-degree distribution in the Lenny release from that in the Etch release is $c \simeq 90$. A solitary top node is to be seen for $x = 10\,446$.

To appreciate the mathematical implications of obtaining $\mu = -1$ from the data, a power-series expansion of equation (2) has to be carried out, leading to the infinite series,

$$\phi(x) = \left(\frac{x+\lambda}{c}\right)^{\alpha} - \frac{\eta}{\mu}\left(\frac{x+\lambda}{c}\right)^{\alpha(\mu+1)} + \frac{\mu+1}{2}\left(\frac{\eta}{\mu}\right)^2\left(\frac{x+\lambda}{c}\right)^{\alpha(2\mu+1)} + \cdots, \tag{3}$$

from which it is not difficult to see that a self-contained and natural truncation for this series can only be achieved when $\mu = -1$. This is necessary if the scale-free character of the distribution is to be preserved; otherwise, with $\mu \neq 1$, different terms in equation (3) will become dominant on different scales of x. It is remarkable that the Debian data conforms to this fact, and consequently, with $\mu = -1$, equation (1) is reduced to a linear, first-order, nonhomogeneous equation,

$$\frac{\mathrm{d}\phi}{\mathrm{d}x} - \left(\frac{\alpha}{x+\lambda}\right)\phi = -\left(\frac{\eta\alpha}{x+\lambda}\right), \tag{4}$$

in which η plays the role of a nonhomogeneity parameter.

With $\mu = -1$ (implying a power-law in the distribution) and with $\alpha = -2$ (implying that the power-law is specifically Zipf's law), the saturation properties of the network (for any value of η and λ) can be abstracted from equation (2) as

$$\phi(x) = \eta + \left(\frac{c}{x+\lambda}\right)^2. \tag{5}$$

One implication of the foregoing result is that nonhomogeneity in the system sets a firm lower bound to the number of rich nodes in the saturation regime, regardless of any arbitrarily high value of x; that is, $\phi \longrightarrow \eta$ as $x \longrightarrow \infty$. In other words, nonhomogeneity defines a finite lower limit to the discrete count of the rich nodes. This clear deviation from the power-law model enables a few top nodes in the network of outgoing links to get disproportionately rich, as shown in Figures 2 and 4. All the links from these top nodes are outwardly directed toward the dependent nodes, making the presence of these

richly linked nodes an absolute necessity and burdening them with the responsibility of maintaining functional coherence in the FOSS distribution.

A scale for the onset of the saturation effects in the network of out-degree distributions can be found when the two terms in the right-hand side of equation (2) are in rough equipartition with each other. This will set a scale for the saturation of the number of links in the frequency distribution as

$$x_{\text{sat}} \simeq \frac{c}{\sqrt{|\eta|}} - \lambda. \tag{6}$$

Considering the out-degree distribution in particular, it is always the case that $\eta = 1$. From the fitting function, noting that $\lambda \ll c$, we can also conclude that $x_{\text{sat}} \sim c$, a simple result that is useful in identifying the nodes that act as "hubs" and deviate from a scale-free distribution. All nodes with a number of links of the order of x_{sat} or greater will belong to this saturation regime. For the network of outgoing links, the Debian data indicates that approximately the top 1% of the nodes fall within this scale, with the package libc6 being the most profusely connected node in all the releases.

The situation is quite the opposite for the network of incoming links, as Figures 1 and 3 show. Here the nodes draw in links to themselves, with all links being inwardly directed toward the nodes. This network of incoming links is complementary in character to the network of outgoing links. As a result, the richly linked nodes of the latter network are poorly connected in the former. In contrast to Figures 2 and 4, which indicate that the rich nodes serve the network to an extent that is disproportionately greater than what a simple power-law behavior would have required of them, we see from Figures 1 and 3 that the most richly linked nodes in the in-degree distribution display a behavior that falls short of what might be expected of a fully power-law trend (the top nodes here ought to have accreted more links if a power law only were to have been followed). So decreasing values of η over two generations of Debian releases show that for a given number of links x, the count of nodes ϕ is reduced. It is then clear that the ability of the top nodes to acquire links in the in-degree distribution becomes progressively weakened (and so it is that the deviation from the power-law behavior becomes sharper). Saturation in the network can also be quantitatively determined by the parameter η, which, when $\mu = -1$, appears as a nonhomogeneity condition in equation (1). The value and especially the sign of η afford us a precise means to distinguish the directed network of incoming links from that of outgoing links. The difference in the respective degree distributions in Figures 1 and 2 (or Figures 3 and 4) underscores this fact.

For small values of x, the poorly linked nodes also deviate from the power-law solution. This is especially true for the in-degree distribution in Figures 1 and 3. For small in-degree and out-degree distributions in the World Wide Web [31], an improved fit is obtained by a simple modification in the global power-law model [1, 27]. This type of modification can also be engineered in equation (5) to obtain a similar fit for the weakly linked nodes. In the limit of small degree distributions for both the in-directed and out-directed networks, where η ceases to have much significance, and where $x \sim 1$ (which, in the discrete count of links, is the lowest value that x can assume practically), an upper bound to the number of the sparsely linked nodes is found to be

$$\phi_{\mathrm{ub}} \simeq \left(\frac{c}{1+\lambda}\right)^2, \tag{7}$$

with the full range of ϕ, therefore, going as $\eta \leq \phi \lesssim \phi_{\mathrm{ub}}$.

4. Modeling Evolution and Saturation

Our model, based on two generations (Etch and Lenny) of a standard FOSS network (Debian), has shown that the saturation properties of the in-degree and the out-degree distributions are differently affected as time passes (marked by new releases of Debian). The degree distribution of the network of outgoing links shows no change when it comes to the model fitting of the top nodes (η maintains the same value). This is expected of these nodes. They form the foundation of the whole network, and their prime status continues to hold. In a semantic sense, meaning flows from these nodes to the derivative nodes. At the opposite end, the very poorly linked nodes in the outgoing network are fitted by changing values of λ (as shown in Figures 2 and 4). Again this is expected. In a mature and robust network, the possibility of semantic variations is much more open in the weakly linked derivative nodes, as opposed to the primordial nodes.

For the in-degree distributions, the situation is contrariwise. Going by Figures 1 and 3, the model fitting can be achieved properly by changing the value of η significantly. Further, with a new release of Debian, η actually decreases, a fact whose import is that the most richly linked nodes in the in-degree distribution (which are also the most-dependent nodes) acquire fewer links than they might have, if the power-law trend were to have been adhered to indefinitely. So, from a dynamic perspective, there is a limit up to which these dependent nodes continue to be linked.

Taking these observations together, we realize that the FOSS network is not a static entity. Rather it is a dynamically evolving network, as any standard software network is known to be [32, 33], undergoing continuous additions (even deletions) and modifications across several generations of Debian releases, contributed by the community of free-software developers. So any realistic model should account for this evolutionary aspect of the network distribution, and by now many theoretical models [34–37] have provided such insight into the dynamic evolution of networks. It is known too that scale-free networks emerge through the simultaneous operation of dynamic growth and preferential attachment [34, 38]. The limiting features of such a scale-free distribution ought also to come out naturally through the long-time dynamics.

The top nodes in the out-degree distribution form the irreducible nucleus of the FOSS network. These nodes are the most influential in the network. From the perspective of a continuum model, we look at the frequency distribution of the nodes in the network of outgoing links as a field $\phi(x, t)$ evolving continuously through time t with the saturation in the number of nodes for high values of x, emerging of its own accord from the dynamics. In keeping with this need, we frame an ansatz with a general power-law feature as

$$\phi(x, t) = \left(\frac{x+\lambda}{c}\right)^{\alpha} + \varphi(x, t), \tag{8}$$

in which $\varphi \longrightarrow \eta$ as $t \longrightarrow \infty$. This prescription is compatible with what equation (2) indicates when $\mu = -1$. Under this requirement, the temporal evolution of the network is described by a first-order, linear, nonhomogeneous equation, going as

$$\tau \frac{\partial \phi}{\partial t} = \frac{\partial \phi}{\partial x} - \frac{\alpha}{c^{\alpha}} (x+\lambda)^{\alpha-1}, \tag{9}$$

in which τ is a representative time scale on which the FOSS network evolves appreciably. Now equation (9) already has a power-law property built in it explicitly and is expected, upon being integrated under suitable initial conditions, to make the saturation features of the top nodes appear because of nonhomogeneity. This is the exact reverse of equation (4), which has nonhomogeneity explicitly designed in it, and upon being integrated, leads to a power-law behavior. The general solution of equation (9) can be obtained by the method of characteristics [39], in which we need to solve the equations

$$-\frac{\mathrm{d}t}{\tau} = \frac{\mathrm{d}x}{1} = \frac{\mathrm{d}\phi}{\alpha (x+\lambda)^{\alpha-1} c^{-\alpha}}. \tag{10}$$

The solution of the $\mathrm{d}\phi / \mathrm{d}x$ equation is

$$\phi - \left(\frac{x+\lambda}{c}\right)^{\alpha} = a, \tag{11}$$

while the solution of the $\mathrm{d}x / \mathrm{d}t$ equation is

$$x + \frac{t}{\tau} = b, \tag{12}$$

with both a and b being integration constants. The general solution is to be found under the condition that one characteristic solution of equation (10) is an arbitrary function of the other; that is, $a = f(b)$, with f having to be determined from the initial conditions [39]. So, going by the integral solutions given by equations (11) and (12), the general solution of $\phi(x, t)$ will be

$$f\left(x + \frac{t}{\tau}\right) = \phi - \left(\frac{x+\lambda}{c}\right)^{\alpha}, \tag{13}$$

which, under the initial condition that $\phi = \eta$ at $t = 0$, will characterize the profile of the arbitrary function f as

$$f(z) = \eta - \left(\frac{z+\lambda}{c}\right)^{\alpha}. \tag{14}$$

Hence, the specific solution can be obtained from equation (13) as

$$\phi(x, t) = \eta + \left(\frac{x+\lambda}{c}\right)^{\alpha} - \left[\frac{1}{c}\left(x + \lambda + \frac{t}{\tau}\right)\right]^{\alpha}, \tag{15}$$

and this, under the condition that $\alpha = -2$, will converge to the distribution given by equation (5), for $t \longrightarrow \infty$. The significance of the initial condition is worth stressing here. For a value of x, the evolution starts at $t = 0$ with an initial node count of $\phi = \eta$, which, under all practical circumstances, is set at $\eta = 1$. This is to say that a node appears in the network with x number of links, where previously there existed no node with this particular number of links. As the network evolves, two things continue to happen: first, new nodes are added to the network, and second, already-existing nodes accrete links in greater numbers. The most heavily linked among the latter started as the primary nodes, and at $t = 0$, their number defines the minimum number of independent packages that are absolutely necessary for the FOSS network to evolve subsequently (for $t > 0$) into a robust semantic system. From a semantic perspective, the initial condition can be argued to have an axiomatic character, and the mature network burgeons from it on later time scales. And during the evolution, the entire

network gets dynamically self-organized in such a manner that the eventual static out-degree distribution has its saturation properties at the highly connected end determined by what the initial field was like at $t = 0$.

The asymptotic properties of equation (15) can now be examined, both in the limit of $t \longrightarrow 0$ and in the limit of $t \longrightarrow \infty$. In the former case, the evolution of ϕ will be linear in t for a given value of x and will go as

$$\phi(x, t) \simeq \eta - \alpha \frac{(x+\lambda)^{\alpha-1}}{c^{\alpha}} \left(\frac{t}{\tau}\right), \tag{16}$$

in which growth is assured only when $\alpha < 0$. This linearity of early growth reflects the assumption of a linear growth of the number of nodes with time [40].

While the temporal evolution obeys linearity on early time scales, in the opposite limit of $t \longrightarrow \infty$, the evolution shifts asymptotically to a power-law trend going as

$$\phi(x, t) - \eta - \left(\frac{x+\lambda}{c}\right)^{\alpha} \simeq -\frac{1}{c^{\alpha}} \left(\frac{t}{\tau}\right)^{\alpha}. \tag{17}$$

Naturally, convergence toward a steady state, as it has been given by the condition in the left-hand side of the foregoing relation, will be possible only when $\alpha < 0$, a requirement that is satisfied by Zipf's law ($\alpha = -2$). Free and open-source software has been known to have its dynamic processes driven by power laws [20, 21], which is a clear sign that long memory prevails in this kind of system.

Now from the steady-state form of the degree distribution, as it is given by equation (2), we can set down for $\mu = -1$ a similar relation for the time-dependent field $\phi \equiv \phi(x, t)$ as

$$\phi(x, t) = \eta + \left(\frac{x+\tilde{\lambda}}{\tilde{c}}\right)^{\alpha}, \tag{18}$$

where $\tilde{\lambda}$ and $\tilde{c}$ are "dressed" parameters, defined as $\tilde{\lambda} = \lambda \nu(x, t)$ and $\tilde{c} = c\,\zeta(x, t)$, respectively. The scaling form of the two functions ν and ζ can be determined by equating the right-hand sides of equations (15) and (18). This will lead to

$$\left(\frac{x+\tilde{\lambda}}{\tilde{c}}\right)^{\alpha} = \left(\frac{x+\lambda}{c}\right)^{\alpha} - \left[\frac{1}{c}\left(x + \lambda + \frac{t}{\tau}\right)\right]^{\alpha}. \tag{19}$$

For scales of $x \gg \lambda$ (typically $x \gtrsim 10$), a converging power-series expansion of increasingly higher orders of λ / x can be carried out with the help of equation (19). The zeroth-order condition will deliver

the scaling profile of ζ as

$$\zeta(x, t) = \left[1 - \left(1 + \frac{t}{x\tau}\right)^{\alpha}\right]^{-1/\alpha}. \tag{20}$$

This function bears time-translational properties, and at a given scale of x, it causes the degree distribution to shift across the x–ϕ plot through time. However, it is also not difficult to see that when $\alpha = -2$, there is a convergence toward $\zeta = 1$ (the steady state limit) as $t \longrightarrow \infty$. And when $x \longrightarrow \infty$, on any finite time scale, $\zeta \longrightarrow 0$. This is why the count of the most heavily connected nodes (for which x has a high value) stays nearly the same ($\phi = \eta$) at all times, a fact that is borne out by the out-degree distributions in Figures 2 and 4. The saturation scale of x for such behavior is given by equation (6). A related fact that also emerges is that time-translation of the degree distribution becomes steadily more pronounced as we move away from $x \sim x_{\text{sat}}$ toward the lower limit of $x = 1$ (the least number of links that a node can possess). Consequently, as the temporal evolution progresses, the out-degree distribution assumes a slanted appearance, with a negative slope in the x–ϕ plane, something shown clearly in Figures 2 and 4. The model fitting in these two plots indicates that the value of c increases with time. This is how it should be, going by the form of the scaling function $\zeta(x, t)$, if we are careful to observe that c in both the plots is to be viewed as $\tilde{c}$, to account for its time-dependent variation.

Information regarding the time-translational properties of the poorly connected nodes is contained in the scaling function $\nu(x, t)$. However, a look at the left-hand side of equation (19) reveals that ν is coupled to ζ, and this nonlinear coupling causes complications. Going back to the power-series expansion in λ / x, as it can be obtained from equation (19), we may suppose that just as the zeroth-order in the series has yielded a proper scaling form for ζ, the higher orders in the series will bring forth a similar form for ν. And indeed we do obtain such a solution, going as $\nu^k = \zeta^{\alpha}\left[1 - (1 + t / x\tau)^{\alpha - k}\right]$, with k being the order of the expansion in the power series. But this result is misleading because the parameter λ and the scaling function $\nu(x, t)$ are influential only when $\lambda \gtrsim x$, with x assuming arbitrarily small values in the continuum model. Therefore, the correct approach here is not to take a series expansion in λ / x, but rather in x / λ, with a proper convergence of the series taking place for higher orders in x / λ. The zeroth-order term of this series gives the scaling form $\nu^{\alpha} = \zeta^{\alpha}\,[1 - (1 + t / \lambda\tau)^{\alpha}]$. The primary difficulty with this result is that the true functional dependence of ζ in this case is not known. This is certainly not going to be the function that is implied by equation (20), because this form of ζ is valid only on scales where $x \gg \lambda$.

Considering everything, the clear message derived from the common pattern exhibited by the two generations of out-degree distributions is that the value of λ has a significant bearing on the number of the preponderant but sparsely connected nodes, a fact that is described by equation (7). In the continuum picture of the degree distribution, λ is the theoretical lower bound of the number of links that the most weakly linked nodes may possess (which saves ϕ from suffering a divergence as $x \longrightarrow 0$, as equation (5) shows). Through the evolutionary growth of the network, an increase in the value of λ suggests that these poorly linked nodes become incrementally relevant to the system by contributing more links in the out-directed network. Now these poorly connected nodes in the out-degree distribution are also the most profusely linked nodes in the in-directed network. Figures 1 and 3 show that for these nodes the value of η decreases with the evolution of the FOSS network. So while these nodes become progressively more relevant as members of the out-directed network (a condition quantified by increasing values of λ), as members of the in-directed network they become progressively less dependent (quantified by decreasing values of η). Analyzing the data of all the six generations of Debian, it is seen for the out-directed network that the value of λ remains nearly the same up to the fourth release, Etch, but grows noticeably thereafter for the next two releases, Lenny and Squeeze. Figures 5 and 6 show, respectively, the in-degree and the out-degree distributions of the release Squeeze.

In contrast, in the in-directed network, the value of λ grows quickly for the early releases and then saturates in the Lenny and Squeeze releases. Remembering that in the in-directed network the most poorly linked nodes are actually the parent nodes of the entire network, we conclude that even these nodes become dependent on other nodes to a small extent. Taken as a whole, as time increases, the interdependency character of the entire network becomes more firmly established, with even the relatively unimportant nodes showing a tendency to contribute outwardly directed links.

Quantitative support in favor of this claim comes from the dynamics of the out-directed network. In this case, the total number of nodes $N_{\text{out}}(t)$ at any given point of time t can be obtained by evaluating the integral

$$N_{\text{out}}(t) = \int_1^{x_{\text{m}}} \phi(x, t)\, \mathrm{d}\, x. \tag{21}$$

The limits of this integral are decided by the limits on the number of links that the nodes possess, 1 being the lower limit and x_{m} being the upper (maximum) limit. The integral in equation (21) can be solved by taking the profile of $\phi(x, t)$ given by equation (15), for $\alpha = -2$.

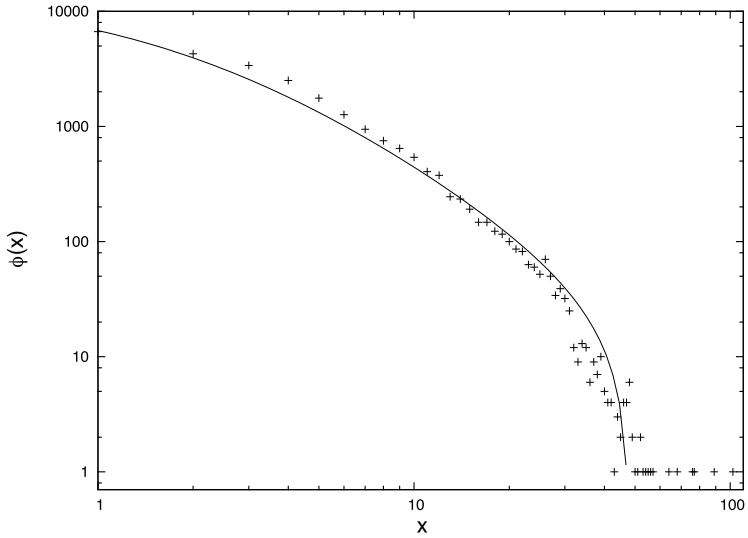

Figure 5. On large scales of x in the network of incoming links of the latest stable release, Squeeze, the saturation of the degree distribution is fitted by the parameter value $\eta = -28$. When x is small, the fit is obtained for $\lambda = 2.2$. For this plot, $c \simeq 265$.

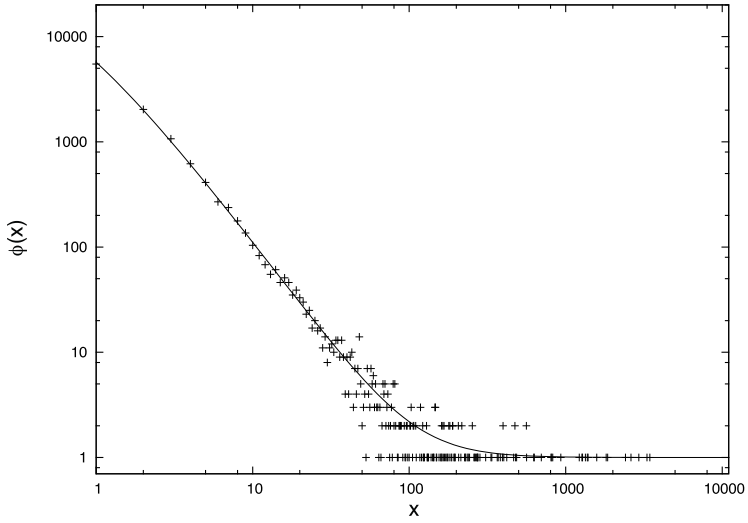

Figure 6. The out-degree distribution of the latest stable release, Squeeze, is in agreement with what the dynamic model predicts. The values of λ and c increase, as expected, to $\lambda = 0.45$ and $c \simeq 110$. The richest node in this distribution has 12 470 links.

Noting that $x_{\mathrm{m}} \gg 1$ (typically $x_{\mathrm{m}} \sim 10^4$) for the out-directed network, the total number of nodes at any time can be estimated as

$$N_{\mathrm{out}}(t) \simeq \eta\, x_{\mathrm{m}} + \frac{c^2}{1+\lambda} - c^2 \left(1 + \lambda + \frac{t}{\tau}\right)^{-1}. \tag{22}$$

On moderate time scales, the last two terms in the right-hand side of equation (22) are roughly equal. So the dominant contribution comes from the first term (the saturation term), as a consequence of which we can set down $N_{\mathrm{out}} \sim x_m$. This argument becomes progressively more correct for large values of x_m, that is, for later releases of Debian.

For the out-degree distribution in the Etch release, $x_m \simeq 9000$, while in the Lenny release, the corresponding number is about 10 000. Using these values from both the releases of Debian, the respective count of N_{out} can be made for the two successive generations. These values of N_{out} represent the number of nodes that contribute at least one link in the out-directed network. In the case of the Etch release, the number of software packages contributing to the out-directed network is counted to be about 8700 (which is closely comparable to the estimated value of $N_{\mathrm{out}} \sim x_m \simeq 9000$), and in the case of the Lenny release, the total count of the out-directed nodes is about 11 000 (which can be favorably compared once again to $N_{\mathrm{out}} \sim x_m \simeq 10\,000$). As a fraction of the total number of nodes, these actual counts indicate that the number of nodes in the out-directed network increases by 0.3% from the Etch release to the Lenny release. This validates the contention that with each passing generation, the network becomes incrementally more robust in terms of out-degree contributions coming from an increasingly greater number of nodes. The values pertaining to the latest stable release, Squeeze, also go along with this trend. In this case the actual count of the out-directed nodes is about 14 000, a number that is again comparable with the estimate of $N_{\mathrm{out}} \sim x_m \simeq 12\,000$. In keeping with the predicted trend, the fraction of nodes contributing out-directed links in this release increases by 1.2%. We also note with curiosity that in these last three Debian releases, Etch, Lenny, and Squeeze, the total number of software packages in both the in-directed and out-directed networks is roughly twice the value of x_m in the out-directed network.

The overall growth of the network, however, slowly grinds to a halt on long time scales. This conclusion cannot be missed in equation (22), which suggests that the total number of nodes increases with time, but approaches a finite stationary value when $t \longrightarrow \infty$, with x_m remaining finite. This inclination of the network to saturate toward a finite-size end can be explained in terms of the finite semantic

possibilities associated with each of the nodes. The extent of making creative use of the existing semantic possibilities of even the most intensely linked of the top nodes is limited. Since most of the nodes in the network depend on such top nodes, there must then be a terminal stage in the growth of the network. Unless novel creative elements in semantic terms are continuously added to the top nodes, the value of x_m will remain constrained within a certain bound, and saturation will happen. So saturation in the network is a consequence of the limit to the various ways in which original functions in the top nodes can be invoked by the derivative nodes. An illustration of this argument is to be seen in Figure 7, which plots actual values taken from all the Debian releases. This plot tracks the growth of the total number of nodes in the out-directed network. All the members of this network contribute at least one out-directed link, and so meaning (the semantic context) is seen to flow out of these nodes. Therefore, these nodes are the bearers of original axioms. That the growth of this entire out-directed network saturates toward a limiting value for the

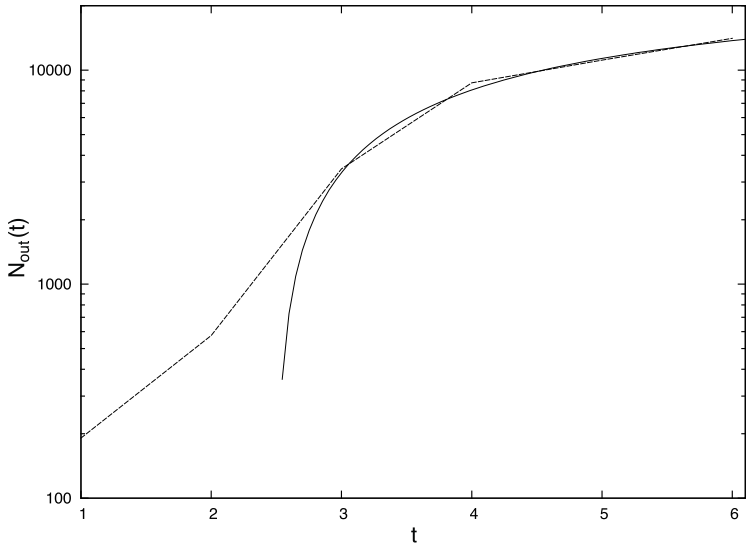

Figure 7. The broken dotted curve plots the growth of the total number of nodes in the out-directed network over six generations of Debian. The continuous curve, following equation (23), gives the fitting function of the data points. The fit, indicating a power-law approach toward saturation, agrees well for the later releases of Debian (the third release onward). In this plot t is a scaled time, marking the generation number. The parameter values are $A = 29\,000$, $B = 113\,000$, and $C = 1.4$, which are compatible when viewed in terms of η, λ, c, and x_m, as they have been set in Figures 2, 4, and 6.

later releases of Debian is quite obvious from the trend indicated in Figure 7. The data curve is fitted by a general form of equation (22), going as

$$N_{\text{out}}(t) \simeq A - \frac{B}{C+t}, \tag{23}$$

where $A = \eta x_m + c^2 (1+\lambda)^{-2}$, $B = c^2 \tau$, and $C = (1+\lambda)\tau$. The implication of the foregoing expression is that the long-time approach toward the terminal stage in the growth of the network is like a power law. From the fitting function, this looks very much true for the later releases of Debian. Now saturation in the growth of the axiom-bearing nodes (with out-directed links) means that the growth of the network of in-directed nodes will also saturate in tandem. The semantic flow in the entire network terminates at these nodes, and as such these "terminal" nodes are also indicators of saturation.

5. Concluding Remarks

This work is based on the networks of direct dependencies in the component-based software Debian. A deeper understanding of dependency-based semantic features can be had on introducing the notions of strong dependencies and package sensitivity, which are instrumental in distinguishing transitive dependencies from conjunctive and disjunctive dependencies [23]. We note that direct and strong dependencies generally tend to be correlated [23]. These features may have a bearing on redundancy in the operating system and its robustness against failure. We may also mention in passing that network structures in component-based software are determined by specific fields, with "depends," which is the basis of this study, being just one of such fields ("conflicts," for instance, being another). A particular field may give rise to specific features in the network, characteristic of itself only.

The mathematical model developed in this work makes a quantitative distinction between the incoming and outgoing distributions in the Debian GNU/Linux network. Similar features are known to exist in the degree distributions of other scale-free networks, and with the mathematical framework applied here, it should become possible to study the saturation properties and the specific directional characteristics of scale-free networks in general. To take an example, the degree distributions of the World Wide Web and Debian appear to be the converse of each other. And so what looks like an in-degree distribution for one is the out-degree distribution for the other, and vice

versa [1]. The model provided here is general enough to capture the specific features of the two different cases by a suitable tuning of the parameters.

Acknowledgments

We thank J. K. Bhattacharjee, A. Kumar, P. Majumdar, V. M. Yakovenko, and S. Zacchiroli for helpful remarks.

References

[1] S. N. Dorogovtsev and J. F. F. Mendes, *Evolution of Networks: From Biological Nets to the Internet and WWW*, New York: Oxford University Press, 2003.

[2] M. Newman, A.-L. Barabási, and D. J. Watts (eds.), *The Structure and Dynamics of Networks*, Princeton: Princeton University Press, 2006.

[3] A. Barrat, M. Barthélemy, and A. Vespignani, *Dynamical Processes on Complex Networks*, Cambridge: Cambridge University Press, 2008.

[4] S. N. Dorogovtsev and J. F. F. Mendes, "Evolution of Networks." arxiv.org/abs/cond-mat/0106144.

[5] S. H. Strogatz, "Exploring Complex Networks," *Nature*, **410**, 2001 pp. 268–276. doi:10.1038/35065725.

[6] R. Albert and A.-L. Barabási, "Statistical Mechanics of Complex Networks," *Reviews of Modern Physics*, **74**(1), 2002 pp. 47–97. doi:10.1103/RevModPhys.74.47.

[7] M. E. J. Newman, "The Structure and Function of Complex Networks," *Society for Industrial and Applied Mathematics Review*, **45**(2), 2003 pp. 167–256. doi:10.1137/S003614450342480.

[8] L. A. N. Amaral and J. M. Ottino, "Complex Networks," *The European Physical Journal B*, **38**(2), 2004 pp. 147–162. doi:10.1140/epjb/e2004-00110-5.

[9] M. E. J. Newman, "Complex Systems: A Survey." arxiv.org/abs/1112.1440.

[10] R. Albert, H. Jeong, and A.-L. Barabási, "Internet: Diameter of the World-Wide Web," *Nature*, **401**, 1999 pp. 130–131. doi:10.1038/43601.

[11] A. Chatterjee and B. K. Chakrabarti (eds.), *Econophysics of Markets and Business Networks*, New York: Springer, 2007.

[12] R. F. i Cancho, R. V. Solé, and R. Köhler, "Patterns in Syntactic Dependency Networks," *Physical Review E*, **69**, 2004 p. 051915. doi:10.1103/PhysRevE.69.051915.

[13] R. F. i Cancho, "Euclidean Distance between Syntactically Linked Words," *Physical Review E*, **70**, 2004 p. 056135. doi:10.1103/PhysRevE.70.056135.

[14] M. Steyvers and J. B. Tenenbaum, "The Large-Scale Structure of Semantic Networks: Statistical Analyses and a Model of Semantic Growth," *Cognitive Science*, **29**(1), 2005 pp. 41–78. doi:10.1207/s15516709cog2901_3.

[15] R. F. i Cancho, C. Janssen, and R. V. Solé, "Topology of Technology Graphs: Small World Patterns in Electronic Circuits," *Physical Review E*, **64**(4), 2001 p. 046119. doi:10.1103/PhysRevE.64.046119.

[16] S. Valverde, R. F. Cancho, and R. V. Solé, "Scale-Free Networks from Optimal Design," *Europhysics Letters*, **60**(4), 2002 p. 512. doi:10.1209/epl/i2002-00248-2.

[17] S. Valverde and R. V. Solé, "Hierarchical Small Worlds in Software Architecture." arxiv.org/abs/cond-mat/0307278.

[18] D. Challet and A. Lombardoni, "Bug Propagation and Debugging in Asymmetric Software Structures," *Physical Review E*, **70**, 2004 p. 046109. doi:10.1103/PhysRevE.70.046109.

[19] N. LaBelle and E. Wallingford, "Inter-Package Dependency Networks in Open-Source Software." arxiv.org/abs/cs.SE/0411096.

[20] D. Challet and Y. L. Du, "Microscopic Model of Software Bug Dynamics: Closed Source versus Open Source," *International Journal of Reliability, Quality and Safety Engineering*, **12**(6), 2005 p. 521. doi:10.1142/S0218539305001999.

[21] D. Challet and S. Valverde, "Fat Tails, Long Memory, Maturity and Aging in Open-Source Software Projects." arxiv.org/abs/0802.3170.

[22] D. J. Watts and S. H. Strogatz, "Collective Dynamics of 'Small-World' Networks," *Nature*, **393**, 1998 pp. 440–442. doi:10.1038/30918.

[23] P. Abate, J. Boender, R. di Cosmo, and S. Zacchiroli, "Strong Dependencies between Software Components," in *2009 3rd International Symposium on Empirical Software Engineering and Measurement (ESEM 2009)*, Lake Buena Vista, FL, 2009 p. 89.

[24] T. Maillart, D. Sornette, S. Spaeth, and G. von Krogh, "Empirical Tests of Zipf's Law Mechanism in Open Source Linux Distribution," *Physical Review Letters*, **101**(21), 2008 p. 218701. doi:10.1103/PhysRevLett.101.218701.

[25] G. K. Zipf, *Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology*, Cambridge, MA: Addison-Wesley Press, 1949.

[26] A.-L. Barabási and H. E. Stanley, *Fractal Concepts in Surface Growth*, New York: Press Syndicate of the University of Cambridge, 1995.

[27] M. E. J. Newman, S. H. Strogatz, and D. J. Watts, "Random Graphs with Arbitrary Degree Distributions and Their Applications," *Physical Review E*, **64**(2), 2001 p. 026118. doi:10.1103/PhysRevE.64.026118.

[28] R. Nair, G. Nagarjuna, and A. K. Ray, "Features of Complex Networks in a Free-Software Operating System," *Journal of Physics: Conference Series*, **365**(1), 2012 p. 012058. doi:10.1088/1742-6596/365/1/012058.

[29] E. W. Montroll, "Social Dynamics and the Quantifying of Social Forces," *Proceedings of the National Academy of Sciences*, **75**(10), 1978 pp. 4633–4637.

[30] S. H. Strogatz, *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*, Reading, MA: Addison-Wesley Publishing, 1994.

[31] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener, "Graph Structure in the Web," *Computer Networks*, **33**(1–6), 2000 pp. 309–320. doi:10.1016/S1389-1286(00)00083-9.

[32] C. R. Myers, "Software Systems as Complex Networks: Structure, Function, and Evolvability of Software Collaboration Graphs," *Physical Review E*, **68**(4), 2003 p. 046116. doi:10.1103/PhysRevE.68.046116.

[33] A. A. Gorshenev and Yu. M. Pis'mak, "Punctuated Equilibrium in Software Evolution," *Physical Review E*, **70**(6), 2004 p. 067103. doi:10.1103/PhysRevE.70.067103.

[34] A.-L. Barabási and R. Albert, "Emergence of Scaling in Random Networks," *Science*, **286**, 1999 pp. 509–512. doi:10.1126/science.286.5439.509.

[35] P. L. Krapivsky, S. Redner, and F. Levyraz, "Connectivity of Growing Random Networks," *Physical Review Letters*, **85**(21), 2000 pp. 4629–4632. doi:10.1103/PhysRevLett.85.4629.

[36] S. N. Dorogovtsev, J. F. F. Mendes, and A. N. Samukhin, "Structure of Growing Networks with Preferential Linking," *Physical Review Letters*, **85**(21), 2000 pp. 4633–4636. doi:10.1103/PhysRevLett.85.4633.

[37] P. L. Krapivsky and S. Redner, "Network Growth by Copying," *Physical Review E*, **71**(3), 2005 p. 036118. doi:10.1103/PhysRevE.71.036118.

[38] A.-L. Barabási, R. Albert, and H. Jeong, "Mean-Field Theory for Scale-Free Random Networks," *Physica A: Statistical Mechanics and Its Applications*, **272**(1–2), 1999 pp. 173–187. doi:10.1016/S0378-4371(99)00291-5.

[39] L. Debnath, *Nonlinear Partial Differential Equations for Scientists and Engineers*, Boston: Birkhäuser, 1997.

[40] S. Valverde and R. V. Solé, "Logarithmic Growth Dynamics in Software Networks," *Europhysics Letters*, **72**(5), 2005 p. 858. doi:10.1209/epl/i2005-10314-9.