# Complex networks in the curriculum of computer engineers

Árpád Horváth
Óbuda University, Center for Innovation and Education
H-8000 Székesfehérvár, Budai út 45., Hungary
Email: horvath.arpad@roik.bmf.hu
and
Zoltán Trócsányi
University of Debrecen and Institute of Nuclear Research of the Hungarian Academy of Sciences
H-4001 Debrecen P.O.Box 51, Hungary
Email: Z.Trocsanyi@atomki.hu

***Abstract*—Recent investigations in the field of complex networks include the analysis of distributions of connections in particular networks and the clustering properties of networks. In the first part of my presentation I analyse a network: the dependency network of the software packages of the Ubuntu distribution of GNU/Linux. It is a directed network, and I analyse the degree distribution of the network for in-degrees (connections directed into a node, the dependent packages), out-degrees (connections directed away from a node) and plain-degrees. I also analyse the clustering properties of the network. In the second part, I discuss the teaching of these topics at the college level. Using the NetworkX modul for the Python programming language students can easily investigate these properties of the package dependency network.**

## I. Introduction

The networks are often used as a synonym of the graphs in the field of sociology and some other fields of the science. Networks contains some kinds of entities. A pair of entity can be connected or not. The entities are called nodes, vertices or points, the connections are called as edges.

Complex networks are big networks with a structure not easy to describe. The aim of the science of the complex networks is to study the properties of real networks. The earliest investigations of networks happened in the field of sociology, they investigated the acquaintanence network of people.

There are a lot of networks in the fields of the enigneering as the World Wide Web, the Internet, whose investigation brings the networks in the focus recently. In biology and medicine the network of protein interactions, the food chain or to forecast the spreading of a disease the acquaintanence and sexual networks are important. Some example can be found in the Table I.

Summarized properties of many studied networks and the methods of the investigations can be found in several papers can be reached on the Web [1], [2].

## II. Definitions

Networks can be directed or undirected. In undirected networks the nodes at the two endpoints of an edge has not the same role in the connection. The pages of the World Wide Web (as a nodes of a networks) can link one to other. These links are the edges of the network. A page linking to the other has other role in this connection as the linked page.

The function and the growth of a network is influenced by the degree distribution of the network. The $k_i$ degree of the $i$-th node is the number of nodes connecting this node. The in-degree and out-degree are defined in a directed network similarly. In-degree includes only the in-connections, where the arrow lead to the given node, and out degree includes only the out-connections.

The $p$ *degree distribution* is a function. $p(k)$ is the probability of a randomly choosen node has the degree $k$, or

$$p(k) = Pr(\text{degree of a randomly choosen node} = k). \quad (1)$$

In directed networks we can investigate the distribution of in-degree and out-degree as well. Many real networks belong to the scale-free networks, defined elsewhere in the article. In this networks there are nodes with degrees orders of magnitude larger as the average degree. The diagram of this distributions is difficult to analyse, as of the large statistical fluctuation at large degrees. In this case the plot of the *cumulative degree distribution* can be useful. The $P(k)$ value of the cumulative degree distribution gives the probability of a randomly choosen node having degree *larger* then $k$:

$$P(k) = Pr(\text{degree of a randomly choosen node} \geq k). \quad (2)$$

Usually in real networks the clustering is large. In the example of the network of acquaintanence, where two people is connected if they have met, this means that two acquaintance of a people know each other with bigger probability, as two randomly choosen people: for example an Indian maharaja and a Hungarian university student.

To measure this property was introduced the clustering coefficient, which shows for a node the ratio of the number of existing edges between its neighbours and the number of the all possible edges between the neighbours. This is defined for undirected networks, so the directed networks need to transform to undirected to investigate them.

Table I
REAL NETWORKS

| nodes | edge exists if... | type |
|---|---|---|
| people | they have met each other | undirected |
| web pages | there is link from one to other | directed |
| routers of Internet | there is wire between them | undirected |
| scientific papers | citation | directed |
| proteins | they participate in the same interaction | undirected |
| words | they are synonyms according to a given dictionary | undirected |
| mathematicians | authors of the same paper | undirected |
| actors | actors of the same movie | undirected |

There can be maximally $k_i(k_i-1)/2$ edges between the $k_i$ neighbours of the $i$th node, in the case they form a complete graph as subgraph, so the $C_i$ *clustering coefficient of node* $i$ is

$$C_i = \frac{2E_i}{k_i(k_i-1)}, \quad (3)$$

where $E_i$ is the actual number of the edges between the neighbours of the edge. The $C$ *clustering coefficient of the network* can be defined as the arithmetic mean of the clustering coefficients of the nodes:

$$C = \frac{1}{N}\sum_{i=1}^{N} C_i \quad (4)$$

## III. PROPERTIES OF REAL NETWORKS AND NETWORK MODELS

The cumulative degree distribution of some real networks can be found in Figure 1. Diagrams use doubly logarithmic scale except e).

The doubly logarithmic scales help us to recognize the power-law function, because if we use such scales the diagram of power-law functions will be strait lines. The cumulative degree distributions in Figure 1 can be approximated by a power-law function for large degrees (except (e)). The importance of the power-law function is the fact, that in networks with degree distributions approximated with power-law there are nodes with extraordinary large degrees, these nodes are called hubs. In networks with degree distributions decreasing as an exponential function there are no hubs.

The networks, whose degree distribution decreasing as a power-law function for large degrees are called *scale-free networks*. It is easy to show, that in this case the cumulative degree distribution decreases as a power-law function with one bigger exponent, because for get the cumulative distribution we need to integrate the original distribution. The scale-free networks can be defined with the equations

$$p(k) \sim k^{-\gamma} \quad (5)$$
$$P(k) \sim k^{-(\gamma-1)} \quad (6)$$

The name scale-free is come from a property apply only for power-law function. Power-law functions are the only group of $f$ functions, that for every $c$ number there is a $k$ number, with that the next equation is true.

$$f(c\cdot x) = k\cdot f(x) \quad c,k\in\mathbb{R}. \quad (7)$$

Figure 2. Paul Erdős, Alfréd Rényi and Béla Bollobás, three Hungarian mathematicians who studied random graphs.

Scale-free networks are common in the world of complex networks. Albert-Laszlo Barabasi and Reka Albert recognized this fact, when they investigated the structure of the Web in 1999. With the help of the programmer Hawoong Jeong they mapped the home pages in the domain of their workplace, the Notre Dame University (nd.edu). They regarded the web pages as nodes, which were connected if one page linked to the other. The collected data have given the power-law nature.

It is time to speak about the models of the complex networks. The first frequently used model is the random graph model. This model In this model the edges of an undirected network are drawn randomly, so it is a good base to determine whether a property of the network is a sign of order or it occures in the random graph as well. In one of its variations, the $N$ number of nodes and a $p$ probability is given. In the model we go through all edge pairs, and we connect them with $p$ probability.

The properties of the model was studied first by Paul Erdős and Alfréd Rényi from 1959, and later by Béla Bollobás (Fig. 2). The random graphs are usually called Erdős-Rényi graph.

In a graph with $N$ edges for a node there can be maximum $N-1$ edges to other nodes. (In this article graphs means simple graph. In simple graph there can be no multiple edges between two nodes, and has no edges from a node to itself.) If we create the edges with $p$ probability in a random graph, than the estimated value of the degree of a node is $(N-1)p$. Than the degree distribution follows Poisson distribution with $(N-1)p$ expected value. This distribution is decrease as an exponential function for large $k$ values, which do not allows the network to have nodes with degree more larger than the expected value. There are no hubs in random networks.

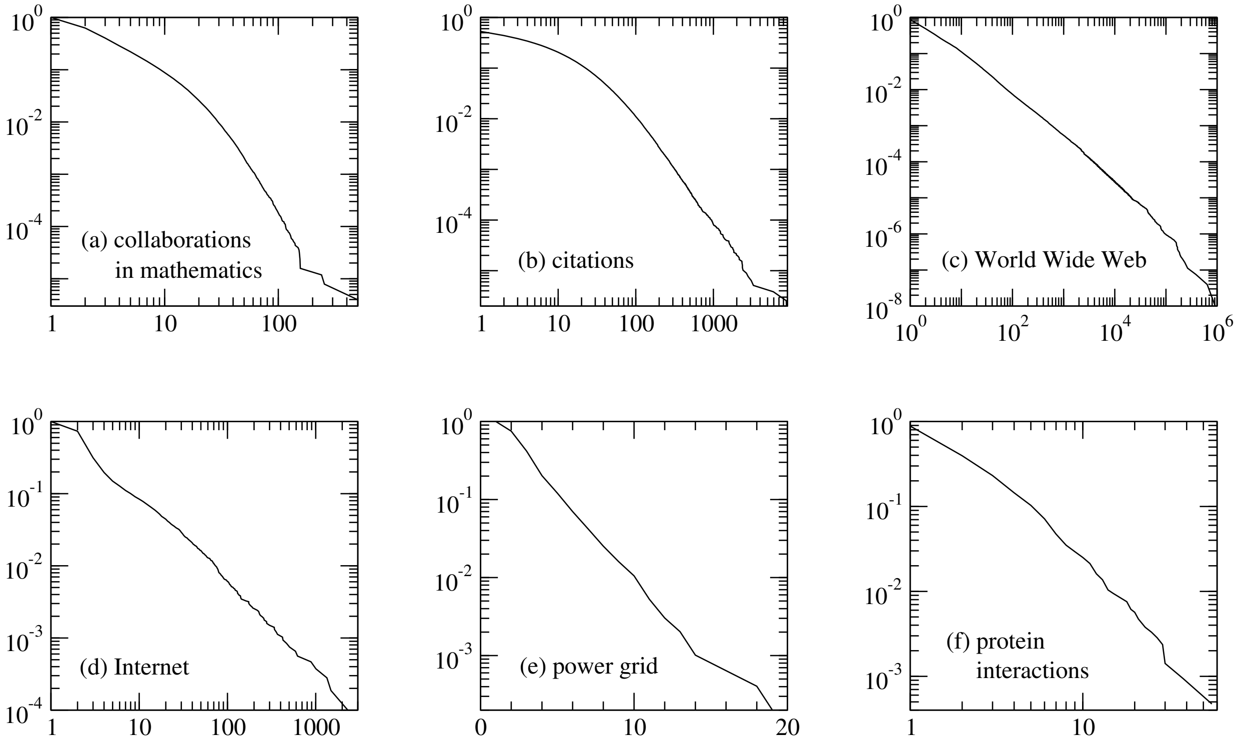


Figure 1. Cumulative degree distributions for six different networks. The horizontal axis for each panel is vertex degree $k$ (or in-degree for the citation and Web networks, which are directed) and the vertical axis is the cumulative probability distribution of degrees, i.e., the fraction of vertices that have degree greater than or equal to $k$. The networks shown are: (a) the collaboration network of mathematicians; (b) citations between papers; (c) a subset of the World Wide Web; (d) the Internet at the level of autonomous systems; (e) the power grid of the western United States; (f) the interaction network of proteins in the yeast. This figure is from the article of M. Newman, references and more details can be found there. [2]

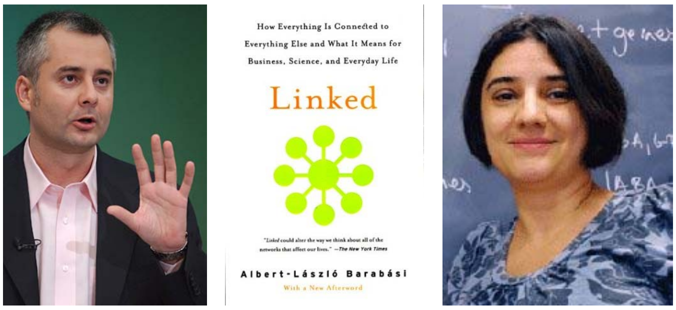


Figure 3. Albert-László Barabási, its book *Linked* [3] and Réka Albert. They have introduced the concept of scale-free networks, and developed a model creating scale-free networks. The book is about the recent progress in the field of complex networks.

The random model was not an acceptable model for the World Wide Web, because they degree distribution differs radically. They completed the random model with two things:

A) The network is growing.
B) Preferential attachment: new nodes attach to the nodes with larger degree with higher probability.

They have found, that neithe the growing, nor the preferential attachment can explain the scale-free degree distributioni without the other condition. The probability of the attachment need to be a linear function of the degree, otherwise the scale-free distribution will not appear. The exponents other than one destroy the power-law character.

In the Barabási–Albert model, they have developed, as a start network we are allowed to choose the network we want. In each step, one new node are created, and it attach to $m$ older nodes. The $m$ is a parameter of the model. For the nodes it is attach to the probability of the attachment is directly proportional to the degree.

The evolution of the network according to the model is shown in Fig. 4. In this case we start with a cycle of three nodes (a). This network is homogen, all of the nodes have two neighbours. We choose the edges created in each steps $m = 2$. In the next step the model choose all of the node pairs with equal probability to attach to them. In this case it chosed the (1, 2) pair of nodes. In the new network there is one node pair of nodes with degree 3. Althout this node pair, the (1,2) pair has the largest probability to attach to it, but there is four pair having one 3-degree-node and one 2-degree-node in: (2,3), (2,0), (1,3), (1,0). In our example we choose a pair of the latter four (c). In the last step we attached the new node to the old nodes with largest degree and second largest degree (d). At this step we can conjecture that the early nodes have bigger chance to get a lot of neighbours, so to become a hub, and the latest ones have usually only the $m$ neighbours they get at its creation.

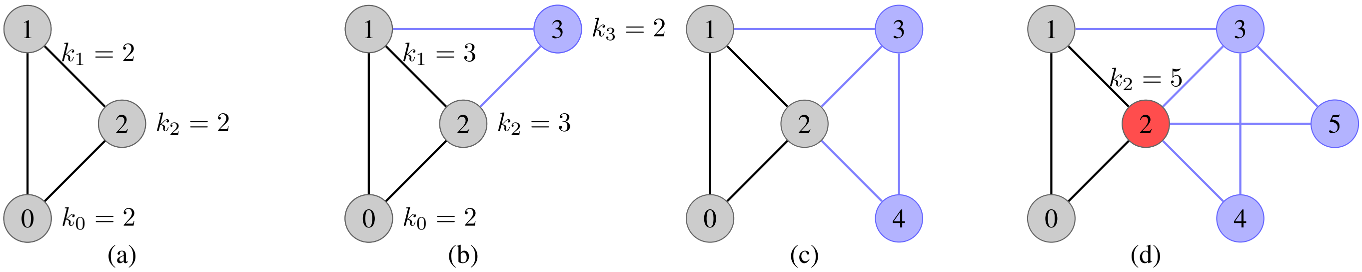


Figure 4. The evolution of the network in the Barabási–Albert model. Explanation is in the text.

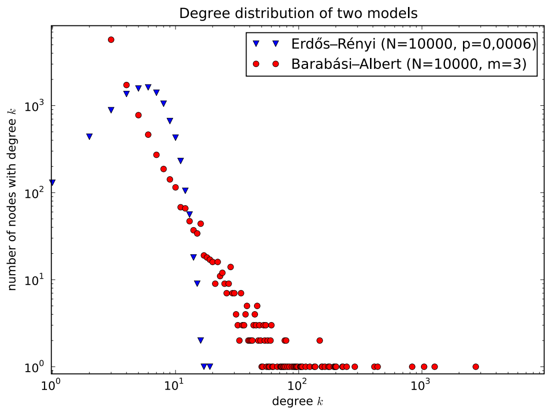


Figure 5. The degree distributions of the random graphs and a network we got from the Barabási–Albert model. On the vertical axis there is the frequency of the degree instead of probability.

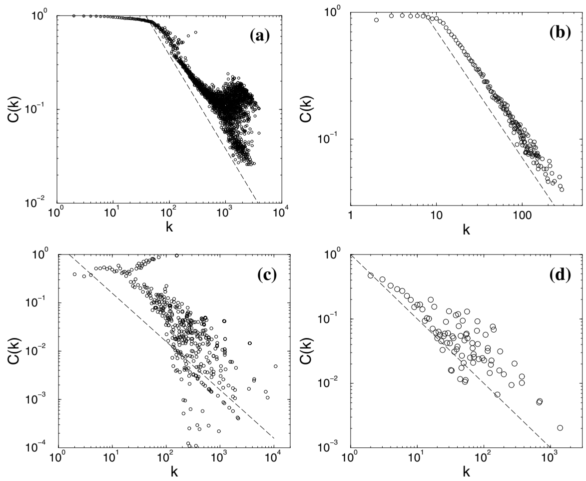


Figure 6. The dependency of clustering coefficient from the degree in four real networks. **(a)** Actors (acted in the same movie according to the `www.IMDB.com` database). ($N = 392\,340$) **(b)** English words (listed as synonyms in a dictionary, $N = 182\,853$) **(c)** A part of the World Wide Web (`www.nd.edu`) ($N = 325\,729$) **(d)** Internet at the Autonomous System level, each node represents a domain ($N = 65\,520$) [4]

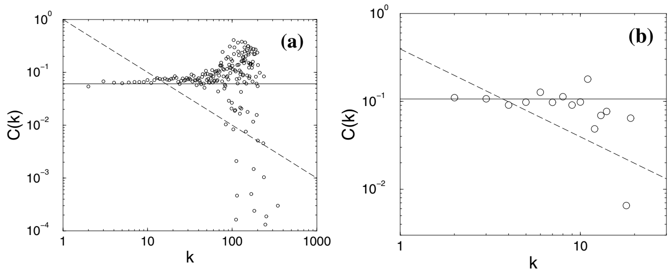


Figure 7. The dependency of clustering coefficient from the degree in two real networks. These are two networks at which the costs of the creation of an edge depends on the distance of the nodes. **(a)** Internet at router level ($N = 260657$) **(b)** Power grid ($N = 4941$) [4]

In the Fig. 5 there is the distribution of the random graph and the network we got from the Barabasi–Albert model. Both network has 10000 nodes and about 30000 edges, the mean of the degrees is about 6. The figure is plotted with doubly logarithmic scales. The largest degree is roughly three times larger than the average in the random graph, but nearly fivehundred time larger in the case of the network we got from the Barabási–Albert model.

In real networks there are some order of magnitude larger clustering coefficients that it is in the random graphs or in the Barabási–Albert model. In many real networks the clustering coefficient decreases with the degree, and it is nearly inversely proportional to it. [4] These networks are called *hierarchical networks*. In other cases the clustering coefficient does not depends on the degree. These latter is true mainly for networks for which the geographical locations of the nodes have large effect on the cost of the edge. For example in the World Wide Web there can be link to a web page located on the other continent without effort, but to build a connection between two routers located in the two sides of the Ocean would be horrible cost.

How can we imagine the hierarchical networks? We should not imagine it as a tree. The nodes of the trees have zero clustering coefficient, because there is – by definition – no cycles in trees. If two neighbours of a node would be connected, there would be a cycle. The hierarchical topology can be imagine with the hierarchical model developed by Albert-László Barabási, Erzsébet Ravasz and Tamás Vicsek [4]. As we see in Fig. 8 at the step 0 the network is a complete graph with 5 nodes. In each steps we create four copies of the network. We place the original one into the middle and we connect the central nodes of the original network with the „peripherial nodes” of the copies.

We can see the properties of the hierarchical model in Fig.

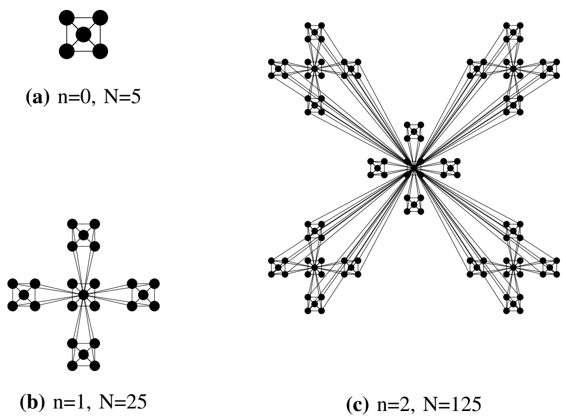


Figure 8. Some step of the hierarchical model. [4]

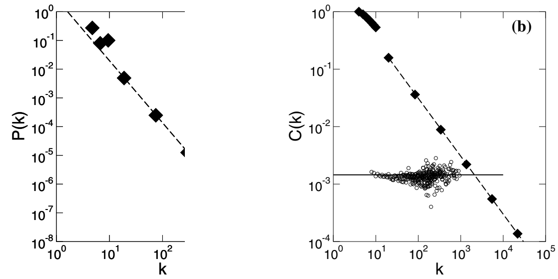


Figure 9. The properties of the hierarchical model for $n = 5^7$. The hierarchical model (a) build a scale-free network and (b) its clustering coefficient is nearly inversely proportional to the degree. On the panel (b) the data of the Barabási–Albert model are plotted with circles (○). [4]

9. This model creates a scale-free network, like the Barabási-Albert model, but unlike the network come from the Barabási–Albert model, in this network the clustering coefficients of the nodes fall inversely proportional with degree, so it creates a hierarchical network.

## IV. The properties of the package dependency network

We have developed a program to analyze the package dependency network of the GNU/Linux operational system.

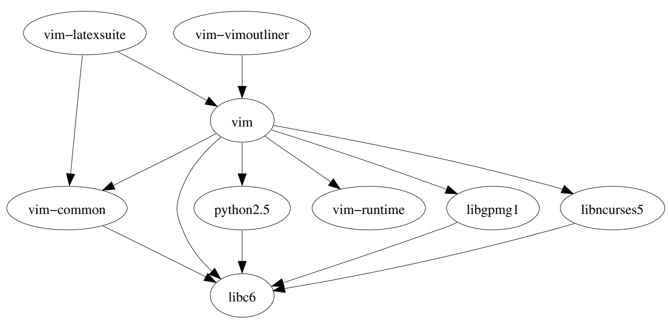


Figure 10. A part of the package dependency network: the surroundings of the package of the Vim text editor. This plot includes the package with the highest degree, the libc6 package, the standard library of the C programming language.

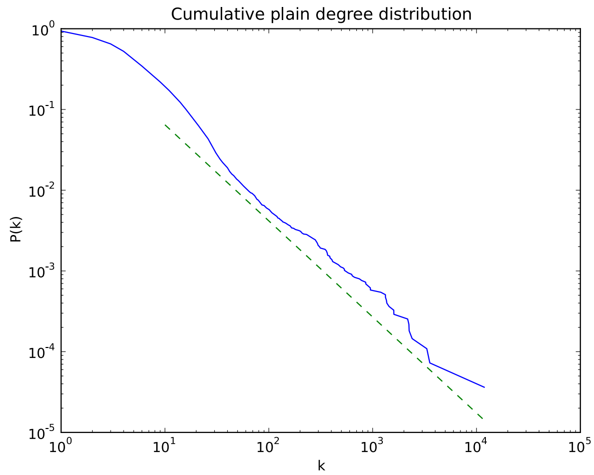


Figure 11. The cumulative degree distribution of the package dependency network (continuous). and the power-law function with the exponent $-1,19$ given by the maximum likelihood method (dashed). $\left(\gamma = 2,19 \pm 0,14 \,{}^{+0,1}_{-0,12}\right)$

The GNU/Linux system has a lot of distribution, that has software packages of its own. These software packets have two prevalent format. One of them are the *deb* packet format developed by the Debian distribution team [5] and used many other distributions include Ubuntu. The packages can be reached on optical disks (CD, DVD) or via repository on the Internet. One package can depend on other packages, that means without them it is not able to work or just partially. This dependences are stored in other files than software packages, so we need to download only this files to create the network of the packages. These dependencies can be treated by the APT package managing tool. This tool can search the dependencies of a package, and install this package with all the other packages it needs or with a removal of a package all the dependent packages to remove. The packages compose a directed network. We defined the direction of the edges to direct form the dependent package to the package it depends on. A part of the package dependency network can be seen in Fig. 10, the package of the Vim text editor with its neighbours. We can observe, that we need the version of 2.5 of the Python programming language (python2.5 package) and the standard library of the C programming language to install the Vim package and the vim package needs to install other packages as well.

In this article we investigated the package dependency network of the Jaunty (9.04) version of Ubuntu distribution. The date of the stored state we investigated is November 3, 2009. The stored files can be found on the web page `http://mail.roik.bmf.hu/complex_networks/packages/archive/`

In this network there number of nodes and edges are $N = 27554, M = 126540$. The network is not connected, the 93.14% of the nodes belongs to the largest component. The diameter of the largest component is 13, so all of the two nodes can be reached from one to the other through 13 edges.

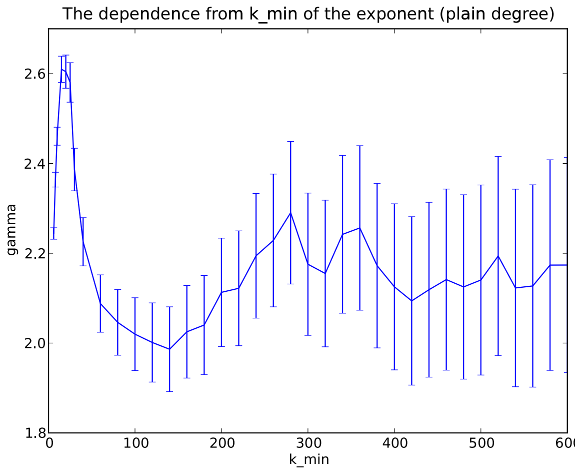


Figure 12. The calculated values of the exponent as a funtion of $k_{\min}$.

The cumulative degree distribution of the network can be seen in Fig. 11. We can see without any calculation, that the scale-free model better fits the distribution than the random graph model.

We can approximate the $\gamma > 0$ absolute value of the network with the maximum likelihood method with its standard deviation [6]:

$$\gamma = 1 + N \left[ \sum_{k_i > k_{\min}} \ln \frac{k_i}{k_{\min}} \right]^{-1}, \quad (8)$$

$$\sigma = \frac{\gamma - 1}{\sqrt{N}} \quad (9)$$

The equations are true for continuous power-law distribution, but with a small modification is acceptable for discrete power-law distribution [7]. We need to use a half smaller $k_{\min}$ value in the equation than the smallest integer value we want to take into account to achive a more accurate value. So we use the equation below.

$$\gamma = 1 + N \left[ \sum_{k_i > k_{\min}} \ln \frac{k_i}{k_{\min} - 0.5} \right]^{-1} \qquad k_{\min} \in \mathbb{Z} \quad (10)$$

The equation gives the exponent and its standard deviation exactly, if the the values really come from a power law distribution. In our case we have not proven the power law nature. The strict analysis of the distribution needs deeper statistical knowledge [7], we think that the proof of the power law nature is worth to be done only with students with such an interest.

In other cases we propose to plot the dependence of $\gamma$ on the minimal degree ($k_{\min}$) as we done in the Fig. 12. As the plot shows the statistical error is very small, but this is not the real uncertainity of the exponent. As we can see, the value of the exponent will be stable around $k_{\min}= 240$. The exponent is worth to interpret only for $k_{\min}$ values above 240. At the uncertainity of the exponent beside the $\gamma$ statistical error we need to include the fluctuation of the $\gamma$ values: the systematic uncertainity. With $k_{\min}= 240$ we get $\gamma = 2.19$ and $\sigma = 0.14$ from the equations (10) and (9). The systematic uncertainity is calculated from the differencies of the values from the value at 240.

Table II

THE PACKAGES WITH THE LARGEST IN-DEGREE ($k_{\text{in}} > 1400$) AS OF NOVEMBER 3, 2009. $k_{\text{in}}$: THE IN-DEGREE, $k_{\text{in}}$': THE DEGREE AT JULY 18, 2008 (UBUNTU 8.04). THE DEGREE OF THE LIBXEXT6 PACKAGE HAS DECREASED, SO IT BECAME THE 15TH, HOWEVER IT WAS THE 8TH EARLIER.

| $k_{\text{in}}$ | $k_{\text{in}}$' | package | description |
|---|---|---|---|
| 11866 | 11113 | libc6 | C shared libraries |
| 3548 | 3230 | libgcc1 | Libraries of C compiler |
| 3319 | 3109 | libstdc++6 | Standard C++ library |
| 2399 | 1940 | zlib1g | Compression library |
| 2243 | 1985 | libglib2.0-0 | The GLib library |
| 2229 | 1929 | perl | Perl language |
| 2170 | 2696 | libx11-6 | X11 client-side library |
| 1595 | 1381 | libgtk2.0-0 | The GTK+ graphical user interface library |
| 1571 | 1296 | python | Python language |
| 1427 | 1279 | libpango1.0-0 | Layout and rendering of internationalized text |
| (...) | | | |
| 1195 | 1865 | libxext6 | X11 miscellaneous extension library |

$$240 \le k_{\min} \le 600 \qquad \Rightarrow \qquad 2.07 \le \gamma \le 2.29$$

The largest difference in the negative direction is 0.12, in the positive direction is 0.10, so

$$\gamma = 2.19 \pm 0.14 \ {}^{+0.1}_{-0.12}$$

Using cumulative distribution, we get the exponent $-\gamma + 1 = -1.19$. Plotting the power law function with that exponent and the cumulative distribution we can check, that the cumulative distribution is really nearly parallel to the power law function $k^{-1.19}$.

We can find the the nodes (packages) with the largest in- and out degrees, and we can compare them with the average of the in- and out-degrees. The Table II compares two states of the network coming from two distinct date. There were notable changes in the order of the packages, and there were a node, which degree has decreased as well.

The average of the in-degree is equal to the ratio of the number of edges and the number of nodes

$$\langle k_{\text{in}} \rangle = M/N = 4.592.$$

However the largest in-degree is 11868, more than 2500 times larger.

The clustering coefficient was defined for undirected networks, so we have needed to transform our directed network into undirected. The clustering coefficient of the network is 0.308. We can compare it with the values coming from the network models. If the network would be a random graph, the $p$ probability of edges, and also the $C$ clustering coefficient would be

$$p = \frac{M}{N(N-1)/2} = C = 0.000333. \quad (11)$$

Table III
PACKAGES WITH THE LARGEST IN-DEGREE ($k_{\mathrm{out}} > 70$).

| $k_{\mathrm{out}}$ | package | description |
|---|---|---|
| 133 | ichthux-desktop | Desktop for Christians |
| 120 | ubuntu-netbook-remix | |
| 113 | ubuntu-desktop | Ubuntu with GNOME desktop environment |
| 111 | ubuntustudio-desktop | |
| 103 | texlive-full | Meta package pulling in all components of TeX Live |
| 83 | libjifty-perl | Programkönyvtár a Jifty webkeretrendszerhez |
| 74 | xubuntu-desktop | Ubuntu with XFCE desktop environment |
| 74 | brdesktop-gnome | Brazil GNOME asztali környezet |
| 71 | rhythmbox | Music player for GNOME |

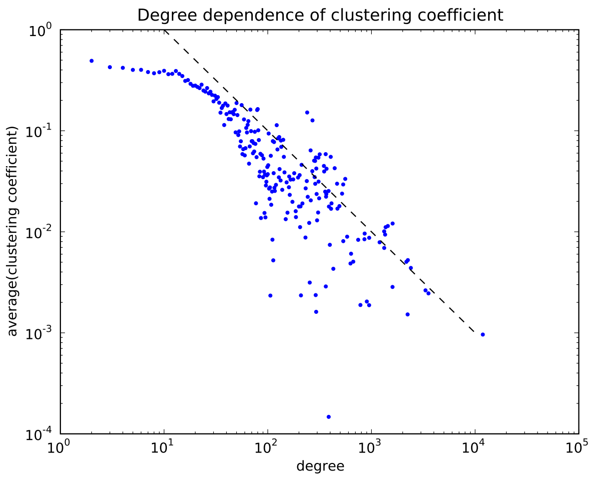


Figure 13. The dependence of the average clustering coefficient on the degree (dots) and the power law function with the exponent $-1$ (dashed line)

In the Barabási–Albert modell the mean degree is $2m$. So we calculated first the mean degree of the package dependency network, and we have choosen the rounded value of the half of the mean degree. As the mean degree is

$$\langle k \rangle = 2M/N = 9.1849, \text{ we have choosen } m = 5$$

for the parameter of the model. With the parameters $N = 27554$, $m = 5$ we got a network with 137745 edges and the clustering coefficient 0.0032. The parameters of the two models give us a clustering coefficient 2–3 order of magnitude larger, than the package dependency network. The clustering coefficient of the hierarchical model is $C_h = 0.743$ [4], which is in the same order of magnitude as the coefficient of the dependency network.

We have plotted the clustering coefficient of the nodes with given degree in the function of degree (Fig. 13). The dashed line is the line of the power law function with the exponent $-1$. The function does not contradict the degree dependency of the other networks. Our network can be called hierarchical network.

There is a node with very small clustering coefficient, can be seen in the bottom of the figure. This is the point of the package `locales` with degree 387. This depends only on the packages `libc6` or `libc6.1` and the translation packages of the languages depend on it, as the packages for Hungarian language `language-pack-hu-base`, `language-pack-gnome-hu-base`, `language-pack-kde-hu-base`. The clustering coefficient of `locales` is 0.000147, there is only 11 edges between its 387 neighbours. 8 edges of the 11 is a dependency on the `libc6` package.

## V. COMPLEX NETWORKS IN THE EDUCATION

One aim of the program development was, that student could analyze a real network created by the computer in the same time. In our center in Székesfehérvár there is two rooms, where the students can use Linux distributions using *deb* packages (Debian and Ubuntu). The program use the `apt` and `apt_pkg` moduls of the Python to get the dependencies, and the `NetworkX` modul [8] to create and analyze the package dependency network and to compare with the models. There is some Hungarian video tutorial and document helping to learn the usage of the NetworkX modul: `http://mail.roik.bmf.hu/complex_networks/mozi`

We have choosen the Python languages because

- it is easy to learn,
- it is included in the GNU/Linux distributions,
- it has modules to analyze networks and to plot functions,
- it can be used interactively, so we get results immediately when we write the commands. This is very useful in learning smaller algorithms and in analizing networks.

The NetworkX modul is easy to use, it contains the basic functions for analizing and modelling networks. We can plot the created networks with the Pylab modul. The Pylab modul allows us to plot functions as the degree distribution. The functions of Pylab is similar to the functions of the MATLAB language. We recommend to use the IPython interactive shell with the option `-pylab`. So the plotting of function and use of mathematical functions are easier as with the standard Python shell.

As the NetworkX modul is based on Python, some calculations (as to get the diameter) needs long time. If we use time critical calculations, it is worth to try the IGraph library written in C [9]. IGraph has developed to handle very large networks. The Python modul based on IGraph have allowed us to make simulations of disease spreading and network evolution parallel. It allows us to write the CPU and memory critical parts of the program in C or C++ as a part of students work.

Unfortunately the usage of NetworkX and IGraph is quite different. So we are inspired to use IGraph instead of NetworkX, but to use the IGraph with ease we need to do some more program.

We introduced the teaching of complex networks as a part of an optional course. In this course the students learn the methods to analize networks, the models of the networks and the properties of the given networks, and they compare it with the real networks.

Our experience is that the students can fulfil such an exercises at home, as to find the $p$ probability above which the giant component appears in Erdős–Rényi graphs, and how it depends this $p$ on the number of nodes, $N$, or to make a program of the modified Barabási–Albert models, and the analysis of the network he get.

## VI. Conclusion

We studied the package dependency network, and we found that it is scale-free hierarchical network, sharing some properties with other real networks. We found that the Python language with some modules (NetworkX or IGrpah) suits for this analysis, these are handy for teaching this field.

We think, that the teaching of the complex networks are possible and useful in some area of the higher education as informatics, physics, sociology, engeneering and biology, or in informatic courses of secondary schools as well.

## References


[1] R. Albert and A. Barabasi, "Statistical mechanics of complex networks," *REVIEWS OF MODERN PHYSICS*, vol. 74, no. 1, pp. 47–97, JAN 2002. [Online]. Available: http://arxiv.org/abs/cond-mat/0106096

[2] M. E. J. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, p. 167, 2003. [Online]. Available: http://arxiv.org/abs/cond-mat/0303516

[3] A.-L. Barabási, *Linked: The New Science of Networks*. Perseus, Cambridge, MA, 2002.

[4] E. Ravasz and A. Barabasi, "Hierarchical organization in complex networks," *PHYSICAL REVIEW E*, vol. 67, no. 026112, FEB 2003.

[5] [Online]. Available: http://www.python.org/

[6] M. E. J. Newman, "Power laws, pareto distributions and zipf's law," *Contemporary Physics*, vol. 46, p. 323, 2005. [Online]. Available: http://arxiv.org/abs/cond-mat/0412004

[7] A. Clauset, C. R. Shalizi, and M. E. J. Newman, "Power-law distributions in empirical data," 2007. [Online]. Available: http://arxiv.org/abs/0706.1062

[8] A. Hagberg, D. Schult, and P. Swart, "NetworkX: High productivity software for complex networks," 2004–. [Online]. Available: http://www.webcitation.org/5aMSYmJUi

[9] G. Csárdi and T. Nepusz, "Igraph," 2003–. [Online]. Available: http://igraph.sourceforge.net/