Just how dense are dense graphs in the real world? A methodological note

Guy Melançon

Just how dense are dense graphs in the real world?

A methodological note

Guy Melancon
LIRMM UMR CNRS 5506, France
Guy.Melancon@lirmm.fr

ABSTRACT

This methodological note focuses on the edge density of real world examples of networks. The edge density is a parameter of interest typically when putting up user studies in an effort to prove the robustness or superiority of a novel graph visualization technique. We survey many real world examples all being of equal interest in Information Visualization, and draw a list of conclusions on how to tune edge density when randomly generating graphs in order to build artificial though realistic examples.

Categories and Subject Descriptors

H.5.2 [**Information Interfaces**]: User Interfaces—*Evaluation/Methodology, Benchmarking*; G.2.2 [**Discrete Mathematics**]: Graph Theory—*Graph Algorithms*; G. [**Probability and Statistics**]: Random generation

Keywords

Interface Evaluation, Information Visualization, Graph Models, Edge Density, Random Generation, Real World Examples

1. INTRODUCTION

Graphs provide a useful mathematical tool for modeling various real world phenomenon. People participating to a same social activity, companies competing or collaborating in a given industrial sector, routers exchanging packets over the internet, or proteins involved in a given process of the living cell are examples of "networks" that can be modeled using graphs. They form a network because of the interactions taking place between the different actors: people, companies, routers or proteins.

In this paper, we will use the term "network" to denote the real world entity that usually maps to a graph after it is modeled. We will reserve the word "graph" to denote the mathematical construction itself. Actors of a network are usually mapped to nodes of the graph. Put differently, nodes are placeholders for actors to which one can attach various attributes (labels, ordinal or numerical values, etc.) in order to reflect properties of the modeled network. Links between actors of the network are mapped to edges, formally defined as pairs of nodes (or ordered couples if direction is relevant). Again, attributes obtained from a description of the studied network can be attached to edges of the graph.

The intent of this methodological note is to look at a particular measure of a graph that is being used when constructing artificial examples of real world networks, namely the *edge density*. We do not argue about what definition should be adopted, but insist on the way the edge density should be used – whatever the definition may be – when generating artificial example graphs in the context of Information Visualization. The construction of artificial example graphs often follow "random" generation processes. (That is, the randomness of these constructions more often comes from the fact that the underlying generation algorithm is unpredictable. This is radically different from being able to prove formally that the algorithm generates all graphs of the considered class with equal probability.) The class from which graphs are drawn can be controlled by varying parameters such as the number of edges, for instance.

Being able to build artificial examples of networks can be necessary. Algorithms are often designed to be able to deal with any graph possessing a number of given properties. Another way of looking at the question is the following. The list of desired properties defines a class of graphs and the hope is that the algorithm will perform as expected with any candidate graph of the class under consideration. A good example could be that of a drawing algorithm (for planar graphs or directed acyclic graphs, for instance). Testing it against a suite of example graphs may provide confidence on the algorithm's robustness or performance. In doing so, the designers should take care in testing the algorithm with a selection of cases sampling the set of graphs it has been designed for. One good strategy here is to use a generation algorithm that will draw graphs uniformly at random (that is all graphs of the considered class should be drawn with equal probability).

Another important situation where artificial networks must be constructed is when putting up usability experiments. In order to avoid biases and to make sure that users' performances are judged equally, one will often prefer an artificial dataset to avoid domain specific knowledge from interfering with the experiment. Testing a new graph navigation technique is a typical example where an artificial example will be required to comply with real world characteristics. For in-

BELIV 2006 Venice, Italy

stance, because layout algorithms typically exploit the link structure, it is essential that the constructed graphs have a topology that mimics that of the real world networks for which the navigation technique has been designed. Finally, because real world networks sometimes can be huge, it might be necessary to generate examples on a smaller scale but that nevertheless capture real world properties.

The concept of a random graph, as defined by Erdös and Rényi [13, 14] (see also [4]), has recently been challenged by more focused classes of graphs found in the real world. Indeed, the relevance of "small world networks" and "scale-free networks" has been assessed by the work of Barabàsi, Watts, Strogatz and others (see [31, 3, 32, 26] for instance; see also [5, 11]). It is nevertheless custom to simply say that "graphs have been generated randomly" when describing the process by which the example graphs have been obtained. What is kept silent here, is that restrictions on the class of all possible graphs have helped the designer to focus on properties of "real-world" examples. One such restriction consists in putting an upper bound on the number of edges of a graph. A useful way to pilot this parameter is to impose the constructed graphs to have a given edge density.

However, the edge density depends on the number of nodes of a graph and must be used with caution to avoid constructing non-realistic examples. We aim at exploring the notion of edge density and provide methodological guidelines to help designers when faced with the problem of "randomly generating example graphs".

2. EDGE DENSITY OF GRAPHS

2.1 Possible definitions

The density of a graph should be a real number reflecting just how many edges it contains. Let us denote a graph as $G = (V, E)$ where V is the set of nodes and E is the set of edges. We shall denote the number of nodes and edges as $n = |V|$ and $m = |E|$. We shall moreover only consider simple graphs, that is undirected graphs without any loop connecting a node to itself. *Theoretically speaking*, a graph is considered dense if its number of edges is *close to* n^2. We shall however adopt a radically different point of view, since graphs with that many edges certainly are virtually absent from reality as far as Information Visualization is concerned. In other words, although a graph with n nodes but "only" $20\,n$ edges does not have many edges when compared to n^2, it does so when compared to networks found in the real world.

One definition of edge density that is often used is to compare the number of edges m with the number of nodes n contained in the graph by computing the ratio:

$$d_\ell = m/n. \tag{1}$$

This measure is often seen as natural because most drawing algorithms will fail to produce readable (no edge crossing) representations of graphs having more than $\sim 4n$ edges [25]. That is, although simple graphs have $n(n-1)/4$ edges on average, only those having a number of edges proportional to n are taken as test candidates for most layout algorithms. (This statement holds when one considers graphs on n nodes where each edge is selected with probability $1/2$. More generally, if the probability of selecting an edge is p, then this number is $p \cdot n(n-1)/2$. See [4] or [5, Chap. 1].) This assumption is present in the Graph Drawing community and holds for most drawing algorithms producing node-link diagrams. As a particular case, planar graphs can have at most $3n - 6$ edges (see [23, Section 1.5] for instance). The assumption is however irrelevant once we consider other types of representations such as matrix representations that can easily hold graphs having a quadratic number of edges.

Ghoniem *et al.* [20] argue that, from an information visualization point of view, a better choice is to compute the ratio:

$$d = \sqrt{m/n^2}. \tag{2}$$

Their main argument against d_ℓ is:

> [although d_ℓ is] *topologically meaningful,* [it is] *not scale invariant since the number of potential edges increases in the square of the number of nodes* [20, Sect. 3.2] (see also [19, 21]).

Indeed, the density value d_ℓ for simple graphs, as defined by Eq. (1), varies over the interval $[0, \frac{n-1}{2}]$ and thus depends on n. The density d_ℓ mapped to $[0, n-1]$ for directed graphs or $[0, n]$ if graphs contain loops. This is somehow undesirable as we would prefer the density value to vary on a fixed interval, whatever the number of nodes in the graph is. Eq. (2) achieves this by mapping the edge density of a graph to the interval $[0, \frac{1}{\sqrt{2}})$ when considering simple graphs. Directed graphs are mapped to $[0, 1)$ or to $[0, 1]$ if loops are allowed. This last remark is of importance when it comes to deciding how the possible d values should be interpolated.

What we want to emphasize here is that even though Eq. (2) maps the edge density to a fixed interval $[0, \frac{1}{\sqrt{2}})$ whatever the number n of nodes in the graph, claiming its scale invariance is somewhat wrong. In other words, *the density cannot be used as a tuning parameter independently from* n. Any of these two definitions can be used depending on their utility or convenience. We shall refer to d_ℓ as the "linear density" and to d as the "square root density".

2.2 Looking closer at density

Let us now look at how the number of edges m in a graph varies according to its density. This is actually a function depending on two parameters, d and n. The curves in Figure 1 show how the number of edges grow as d varies on $[0, \frac{1}{\sqrt{2}})$.

Hence, for instance, a graph on 200 nodes having a density of $d_1 = 0.1$ has 400 edges, whereas for the same density a graph with 500 has 2500 edges. When the density goes to $d_2 = 0.2$, the number of edges respectively increases to 1600 (for a 200 nodes graph), and 10000 edges (for a 500 nodes graph). Observe that in each case, the number of edges has been multiplied by 4 which was predictable since the number of edges is given by the identity $m = d^2 \cdot n^2$ and that $d_2 = 2 \cdot d_1$.

However, although in each case the number of edges has increased identically by a factor of 4, the number of edges when compared to the number of nodes has been affected in a radically different manner. Indeed, for $d = 0.1$ a graph of $n = 200$ nodes has $2n = 400$ edges. When $n = 500$, the number of edges already is equal to 5 times the number of nodes of the graph. When d increases to $d = 0.2$, the number of edges of a 200 node graph goes to 1600, which

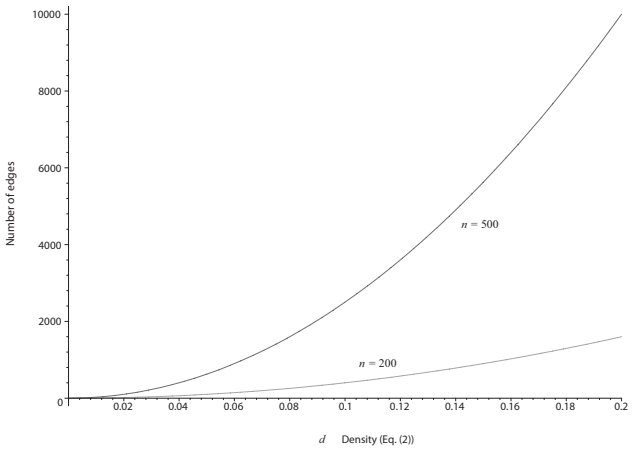

Figure 1: The curve shows how the number of edges of a graph grows depending on its density d (Eq. (2)). The lower curve corresponds to $n = 200$ while the upper curve is for $n = 500$.

is 8 times its number of nodes. A graph of 500 nodes and density $d = 0.2$ should have 20 times more edges than nodes which is rarely seen in the real world (see section 3). Even if realistic, such a ratio is rather high and it might be unreasonable to use a node-link layout to visualize such a graph. Put differently, it might be pointless to compare the performance of a navigation technique (or that of a user using the technique) against a node-link layout algorithm on a graph having $20 \cdot n$ edges, that is, with $d_\ell = 20$. The node-link layout is almost sure to lose (but this, of course, depends on the type of graph being visualized and/or navigated). This statement actually already is true for lower values of d_ℓ.

Figure 2 shows snapshots of ramdom simple graphs with fixed density and increasing size (number of nodes). Each drawing has been rescaled (for each value of n) so that the size of nodes compare. As a consequence, the densities d_ℓ and d also relates to visual density or opacity inside the circular drawing of the graphs. By contrast, Figure 3 shows graphs with increasing size all having the same$d$ density. Images have been scaled so that nodes of all graphs have the same size. As a consequence, any part inside the circular drawing of these graph relatively have the same "visual" density.

We ought to look at how the d_ℓ density is affected by the increase in d. In other words, we wish to look at the number of edges of a graph G as a multiple of n, that is $m = k \cdot n$ (or $d_\ell(G) = k$) as d varies. Our argument is simple. In order to test an algorithm or a navigation technique's robustness with respect to the size and number of edges of a graph, we should consider letting m grow as $m_0 = n$, $m_1 = 2n$, $m_2 = 3n$, and so on up to a given order of magnitude. A majority of the examples we collected confirm that indeed networks have a linear number of edges with $d_\ell \ll n$. Again, it only makes sense to test most layout algorithms on graphs with $k \cdot n$ edges with rather low values of k (see section 3). The relation between d and d_ℓ is straightforward:

$$d_\ell = d^2 \cdot n \tag{3}$$

from which we deduce:

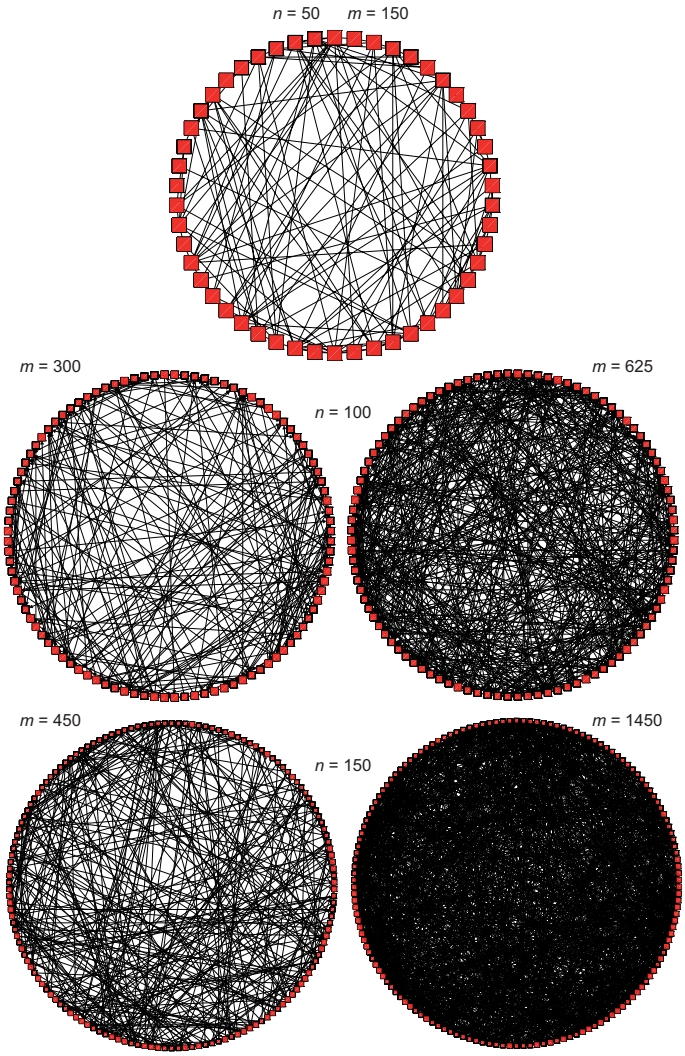

Figure 2: The picture gives a visual impression of how density d_ℓ (left) and density d increase as n grows. We use fixed density $d_\ell = 3$ and $d = 0.25$, so that the number of edges coincide for $n = 50$.

From Eq. (3), we see that in order to map to graphs with $n, 2n, 3n, \ldots, kn, \ldots$ edges, the "square root" density d should vary over the set

$$\{\sqrt{\frac{1}{n}}, \sqrt{\frac{2}{n}}, \sqrt{\frac{3}{n}}, \ldots \sqrt{\frac{k}{n}}, \ldots\}, \tag{4}$$

showing that the relevant values for d actually depend on n. That is, as n grows, d should be constrained to lower values in order to map to realistic graphs, as confirmed by Figure 4. Indeed, a graph with $30n$ edges has density $d \sim 0.25$ when $n = 500$ whereas the density goes up to $d \sim 0.39$ for a graph with $n = 200$ nodes.

3. REAL WORLD EXAMPLES

We list here example datasets we have collected from various places and covering as many application domains as possible. We plan to continue this effort and publicize the

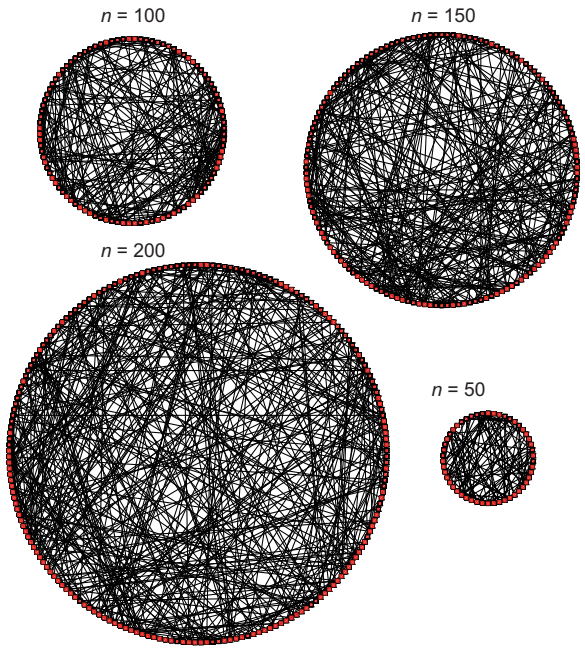

Figure 3: The picture gives an impression of how density d_ℓ remains visually "constant" as n grows. The example graphs were built using $d_\ell = 3$ and $n = 50, 100, 150, 200$.

available statistics on the web[1]. Note that not all networks are freely available. Whenever possible, we provide information to locate and get the example networks. In each case, we give the square root density d and the linear density d_ℓ of the graph underlying the considered example network.

3.1 Small Worlds

Many networks found in the real world share the so-called "small world" property. The expression "small world" refers to the fact that networks often organize into communities, themselves being small worlds organized into sub-communities, and so on. This organization into communities rely on close relationships of people belonging to a same subgroup. In other words, people having common acquaintances are likely to already know each other. Or put more formally, nodes having common neighbors are likely to already be connected by a link. The interested reader will want to browse Watts' novel [33] or read more specialized literature [32, 27, 5, 11, 29].

We also list here graphs that do not exactly fall within this category, but that more exactly are "scale-free". That is they are organized around a few nodes with a very high degree. Although it is possible to find scale-free graphs that are also small worlds, some graphs are scale-free but not small world. That is, scale-free graphs can be community-less. This is the case for the yeast protein interaction networks, for example. The interested reader will want to look at [5, 11].

[1]Anyone wishing to share data should feel free to contact us. Note that, as far as edge density is concerned, we do not require the actual dataset but only need to know about the number of nodes and edges.

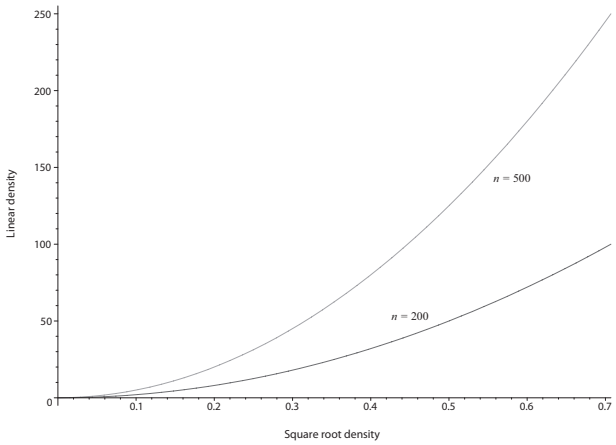

Figure 4: The curve shows how the "linear density" d_ℓ (Eq. (1)) relates to the "square root" density d (Eq. (2)).

Dataset (source)	n	m	d_ℓ	d
Yeast protein interaction [30]	985	899	0.91	0.0304
IV London 2004 Co-authorship graph [8]	1160	2013	1.74	0.0387
Internet routers [7]	11174	23409	2.09	0.0137
Word association (web forms) [10]	2975	6981	2.35	0.0281
Internet routers [7]	10900	31180	2.86	0.0162
InfoVis Contest 2004 co-citation graph [9]	743	2219	2.99	0.0634
Yeast protein interaction [35]	3833	11942	3.16	0.0285
Painters dataset [34]	502	2486	4.95	0.099
Co-authorship graph [6]	1506	7766	5.16	0.0585
Email network [12]	59912	447543	7.47	0.0112
Word association (nouns in dictionary) [17]	51511	392142	7.61	0.0122
Peer-to-peer (Kazaa) file sharing [22]	3403	30555	8.98	0.0514
InfoVis Contest 2004 co-authorship graph [9]	1953	17970	9.20	0.0686
Word association (verbs in dictionary) [16]	9043	101603	11.24	0.0352
Social network (IMDB) [2]	419	5651	13.49	0.1794
Passenger air traffic 2000 [1]	1148	16523	14.39	0.112
Web pages [18]	589	15120	25.67	0.2088
Constraint programming [18]	240	6300	26.25	0.33
Peer-to-peer (web) file sharing [22]	6049	1866271	308.53	0.2258

The table lists the values n, m, d, and d_ℓ for graphs coming from different application domains. All graphs have a single connected component.

- The data (n and m) related to yeast protein interaction was extracted from the cited papers. The number of nodes (proteins) and edges (interactions) strongly

depends on the underlying biological question. In any case however, we can expect the linear density of such graphs to remain below $d_\ell \leq 4$.

- The word association networks were obtained from the authors of the cited papers. One example was obtained through web forms. People were asked to spontaneously respond to a list of suggested words drawn at random. The two other networks were built from the *Robert* French dictionary. In one case the network was restricted to nouns. There was an (undirected) edge between two words if the first appeared in the definition of the other. The other network was built from verbs.
- One dataset was used for the InfoVis Contest in 2004. All data is available from the InfoVis Contest Repository website [15]. The graphs considered here were borrowed from [9] (see also `www.cs.ubc.ca/~tmm/papers/contest04/entry.html`).
- The email network data is available at `www.theo-physik.uni-kiel.de/~ebel/`.
- The peer-to-peer file sharing data concerns the *largest connected component* of the graph describing the exchange of a single file between peers.

The graphs are sorted according to their d_ℓ values in order to make it clear that:

- Most graphs have a rather low linear density value below $d_\ell \leq 10$.
- Most graphs have a d value below 10% with an average value of 6% for the considered examples.
- The d density *does not necessarily increase with* d_ℓ: it all depends on the value of n.
- the example with $d_\ell = 308.53$ stands as an exception.

Ghoniem uses a dataset coming from constraint programming which probably explains why the authors of [20, 21] conducted their experiment using graphs with density values $d = 0.2, 0.4$ and 0.6. Note that a value of $d = 0.6$ is rather high knowing that d actually varies over $[0, \frac{1}{\sqrt{2}})$ (and not over $[0, 1]$ as the authors assumed). Also, interpolating at equidistant values over the interval is questionable. Ghoniem's experiments were based on graphs of size $n = 20, 50$ and 100. Even if their example graphs are relatively small, using a density of $d = 0.4$ or $d = 0.6$ for $n = 100$ corresponds to a linear density of $d_\ell = 16$ or $d_\ell = 36$ which is rather high when compared to the real life datasets we list (Ghoniem himself only reaches a value of $d_\ell \sim 26$). A study wishing to establish a clear comparison between matrix representations and other node-link representations should concentrate on examples with a much lower d_ℓ density. Indeed, force-directed layout algorithms actually are amongst the only available techniques to deal with graphs not having any particular topological properties. However, they often fail to easily produce readable layouts when $d_\ell > 4$. The case of the painters dataset [34] required the design of a specific force model based on the work of Noack [28]. Boutin *et al.* [6] specifically designed a strategy to filter edges out of a graph in order to help the identification of clusters.

The same type of error was observed in [24]. In order to assess of the usability and superiority of their navigation technique, the authors conducted a usability study using an artificial email network. To build their network example, the authors first set its size to $n = 200$ and density at $d = 30\%$. (That is, $m = 3600$.) This is rather high considering that it corresponds to a linear density of $d_\ell = 18$, which does not compare to the example reported in [12]. A number of edges equal to $m = 1500$ (or $d = 13.72\%$) would have been more acurate.

3.2 Web crawls

The following networks have been obtained by web crawlers, in an attempt to get a better view of the traffic taking place over several regions of the world. As expected the networks have a huge size. The incredibly large size of these examples undoubtedly show that d cannot be used without any reference to n. All data is publicly available from the Laboratory for Web Algorithmics website (see `law.dsi.unimi.it`).

World region (crawl year)	n	m	d_ℓ	d
India (2004)	1382908	16917053	12.23	0.0030
United Kingdom (2002)	18520486	298113762	16.10	0.00093
Europe (2005)	862664	19235140	22.29	0.0051
United Kingdom (2005)	39459925	936364282	23.73	0.00077
Indochina (2004)	7414866	194109311	26.18	0.0019
Italy (2004)	41291594	1150725436	27.87	0.00082
Slovakia (2005)	50636154	1949412601	38.50	0.00087

4. CONCLUSIVE REMARKS

4.1 So what density should I use when building my example graphs ?

What we have emphasized here is that when looking at examples from the real world, it seems that the linear density d_ℓ is a much better descriptor of the complexity of networks with respect to the application domain. That is, d_ℓ seems relatively constant through all examples of a given application domain when compared to the $[0, \frac{n-1}{2}]$ theoretical range. For example, graphs extracted from the internet – hyperlinks between web pages or physical connections between internet routers – most of the times have a linear density that can exceed 10 to easily reach 20. Other examples – social networks, co-authorship, graphs from linguistics – all fall below the threshold $d_\ell \leq 10$. It also seems that many examples remain under $d_\ell \leq 5$. Note however that different types of graphs, with varying density, can be considered within a same application domain. Looking at classes of a Java library for instance, graphs with different density can be obtained by looking either at the inheritance structure or call graphs[2]. In other words, different questions arising

[2]We wish to thank one of the anonymous referees for pointing out this relevant observation and providing the example.

in a same application domain might correspond to different types of graphs. This is not really surprising as task and data types indeed are related. We will nevertheless talk about an "application domain" meaning a type of graph associated with a set of relevant tasks.

In other words, when considering a given application domain, you may well select a density d_ℓ and work with that density independently from n – that is, you could use the same d_ℓ for all example graphs you generate, on all cardinalities. On the contrary, when using the density measure d, you should select d depending on the size of the example you wish to generate as indicated by Eq. (4).

However, it may be more convenient to use the density d to tune the algorithms used to generate example graphs because it varies over a fixed interval. The value to be used should then be chosen according to a target application domain, and should be computed from the adequate d_ℓ value. More precisely, suppose one wants to randomly generate a graph to test a technique developed for the navigation of internet graphs while using artificial example graphs of a size smaller than the real world web crawls. Let d and d_ℓ be the densities computed from a real world example graph with given n and m ($d = \sqrt{m/n^2}$, $d_\ell = m/n$). Let q be such that the size of the example to be generated will be $N = n/q$. Because we want the linear density to remain constant, $d_\ell^{(N)} = d_\ell$, we must have $M = m/q$, so we find $d^{(N)} = d \cdot \sqrt{q}$. In other words, as we generate examples of size smaller than real world examples we can allow the density d to increase only by a square root factor.

Figure 5 illustrates how the density d varies as q increases when mimicking the 2004 web crawl on India. When generating a graph with $N = 1000$ nodes, that is if $q \sim 10^3$, we can use a density $d \sim 0.09$ close to 30 times that of the original real world example. Starting from Ebel's email network with $n = 59912$ to build a $N = 200$ nodes example graph, we compute the above transformation with $q \sim 300$ so we can use a density close to $\sqrt{300} \cdot 0.0112 = 0.194$, leading to a graph with about 1500 edges.

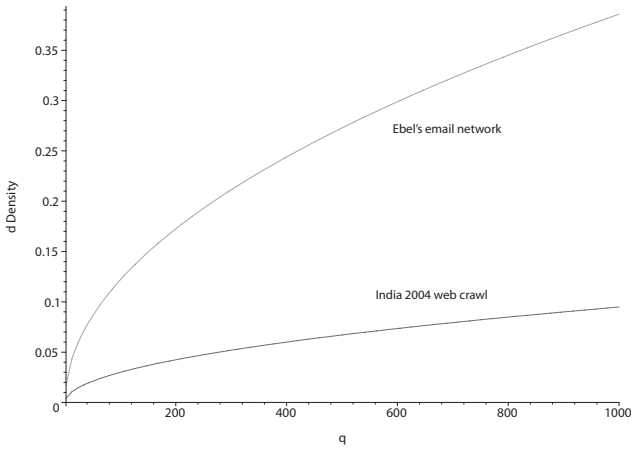

Figure 5: The curve shows how the density of example graphs vary as q increases, based on the real world web crawl for India in 2004 ($n = 1382908$, $m = 16917053$).

4.2 Models of random graphs

We have made a point explaining how the density d or d_ℓ should be used in order to construct "realistic", though artificial, example graphs. Properly tuning the density is however often not enough. Different algorithms will typically produce different types of graph because they *model* different classes of networks. Properly choosing a model has a definite impact on the possible uses of the artificial examples and thus on the conclusion one can draw from the experiment that was conducted using these artificial graphs, or on the robustness of a drawing algorithm.

For example, it is certainly wrong to simply draw graphs at random using the Erdos-Rényi model when designing a user experiment trying to assess of the usability of a visual navigation technique. Indeed, Erdos-Rényi graphs have a more or less constant node degree simply because edges are drawn between nodes without any specific pattern. However, most examples found in the real world are much more inhomogeneous and show properties that radically differ from the Erdos-Rényi graphs. Barabàsi and others extensively study small world networks or scale-free networks through their degree distribution. Scale-free graphs, for example, typically show power law degree distribution. Barabàsi suggested that these networks arise from a pattern he calls "preferential attachements". That is, new actors will preferably get connected to high degree nodes, thus leading to a "rich get richer" pattern. The book by Bornholdt and Schuster [5] surveys Barabàsi preferential attachement model (with an interesting chapter by Bollobàs) and many other useful and "realistic" model that should be preferred to Erdos-Rényi when building test cases. The interested reader will want to look at the recent books [5] and [11]. A comparison of all available models and algorithms for generating random graphs is out of the scope of this short note and merits to be adressed separately. Again, our ambition here was to focus our discussion on the *edge density* of graphs and on how it should be used when building "realistic" examples.

5. REFERENCES

[1] M. Amiel, G. Melançon, and C. Rozenblat. Réseaux multi-niveaux : l'exemple des échanges aériens mondiaux. *M@ppemonde*, 78, 2005.

[2] D. Auber, Y. Chiricota, F. Jourdan, and G. Melançon. Multiscale navigation of small world networks. In *IEEE Symposium on Information Visualisation*, pages 75–81, Seattle, GA, USA, 2003. IEEE Computer Science Press.

[3] A.-L. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 286:509–512, 1999.

[4] B. Bollobás. *Random Graphs*. Academic Press, London, 1985.

[5] S. Bornholdt and G. Schuster, editors. *Handbook of Graphs and Networks: From the Genome to the Internet*. Wiley-VCH, 2003.

[6] F. Boutin, J. Thièvre, and M. Hascoët. Focus-based filtering + clustering technique for power-law networks with small world phenomenon. In R. F. Erbacher, J. C. Roberts, M. T. Gröhn, and K. Börner, editors, *SPIE Electronic Imaging / Visualization and Data Analysis*, volume 6060, San Jose, California, 2006. SPIE IS&T.

[7] Q. Chen, H. Chang, R. Govindan, and S. Jamin. The

origin of power laws in internet topologies revisited. In *IEEE INFOCOM*, Anchorage, Alaska, 2001. IEEE Communications Society.

[8] Y. Chiricota, 2006. Personal communication.

[9] M. Delest, T. Munzner, D. Auber, and J.-P. Domenger. Exploring infovis publication history with tulip (2nd place - infovis contest). In *IEEE Symposium on Information Visualization*, page 110. IEEE Computer Society, 2004.

[10] D. Dion, D. Auber, B. Leblanc, and G. Melançon. Graphe d'associations verbales : élaboration et visualisation. In *Cognitique : vers une informatique plus cognitive et sociale*, pages 223–232. Cépaduès-Editions, 2003.

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

[12] H. Ebel, L.-I. Mielsch, and S. Bornholdt. Scale-free topology of e-mail networks. *Physics Reviews E*, 66(035103(R)), 2002.

[13] P. Erdos and A. Renyi. On random graphs. *Publ. Math. Debrecen*, 6:290–297, 1959.

[14] P. Erdos and A. Rényi. On the evolution of random graphs. *Publ. Math. Inst. Hungar. Acad. Sci.*, 5:17–61, 1960.

[15] J.-D. Fekete, G. Grinstein, and C. Plaisant. "ieee infovis 2004 contest", the history of infovis (www.cs.umd.edu/hcil/iv04contest), 2004.

[16] B. Gaume, 2003. Personal communication.

[17] B. Gaume, N. Hathout, and P. Muller. Désambiguïsation par proximité structurelle. In *Conférence Traitement Automatique du Langage Naturel (TALN'2004)*, pages 205–214, Fez, Maroc, 2004. ATALA.

[18] M. Ghoniem. *Outils de visualisation et d'aide à la mise au point de programmes avec contraintes*. Phd, Université de Nantes, 2005.

[19] M. Ghoniem, J.-D. Fekete, and P. Castagliola. Comparaison de la lisibilité des graphes en représentation noeuds-liens et matricielle. In *IHM 2004*, ACM International Comference Proceedings Series, pages 77–84, Namur, Belgique, 2004. ACM Press.

[20] M. Ghoniem, J.-D. Fekete, and P. Castagliola. A comparison of the readability of graphs using node-link and matrix-based representations, 2004.

[21] M. Ghoniem, J.-D. Fekete, and P. Castagliola. On the readability of graphs using node-link and matrix-based representations: a controlled experiment and statistical analysis. *Information Visualization*, 4(2):114–135, 2005.

[22] A. Iamnitchi, M. Ripeanu, and I. T. Foster. Small-world file-sharing communities. In *IEEE INFOCOM*. IEEE Communications Society, 2004.

[23] D. Jungnickel. *Graphs, Networks and Algorithms*. Springer Verlag, 1999.

[24] B. Lee, C. S. Parr, C. Plaisant, B. B. Bederson, V. D. Veklser, W. D. Gray, and C. Kotfila. Treeplus: Interactive exploration of networks with enhanced tree layouts. *IEEE Transactions on Visualization and Computer Graphics, Special Issue on Visual Analytics*, To appear.

[25] G. Melançon and I. Herman. Dag drawing from an information visualization perspective. In W. d. Leeuw and R. v. Liere, editors, *Joint Eurographics and IEEE TCVG Symposium on Visualization (Data Visualization '00)*, pages 3–13, Amsterdam, 2000. Springer-Verlag.

[26] M. Newman, D. Watts, and S. Strogatz. Random graph models of social networks. *Proceedings of the National Academy of Sciences*, 99:2566–2572, 2002.

[27] M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45:167–256, 2003.

[28] A. Noack. An energy model for visual graph clustering. In *11th International Symposium on Graph Drawing (GD 2003)*, volume 2912 of *Lecture Notes in Computer Science*, pages 425–436, Perugia, Italy, 2003.

[29] J. Park and M. E. J. Newman. The statistical mechanics of networks. *Physics Reviews E*, 70, 2004.

[30] A. Wagner. How the global structure of protein interaction networks evolves. *Proceedings of the Royal Society London B*, 270:457–466, 2003.

[31] D. Watts and S. H. Strogatz. Collective dynamics of "small-world" networks. *Nature*, 393:440–442, 1998.

[32] D. J. Watts. *Small Worlds*. Princeton University Press, 1999.

[33] D. J. Watts. *Six Degrees: The Science of a Connected Age*. W. W. Norton & Company, 2004.

[34] J. v. Wijk and F. v. Ham. Interactive visualization of small world graphs. In T. Munzner and M. Ward, editors, *IEEE Symposium on Information Visualisation*, Austin, TX, USA, 2004. IEEE Computer Science press.

[35] S. Wuchty, A.-L. Barabasi, and M. T. Ferdig. Stable evolutionary signal in a yeast protein interaction network. *BMC Evolutionary Biology*, 6(8), 2006.