Criado, R., Flores, J., González-Vasco, M. I., & Pello, J. (2007). Choosing a leader on a complex network. *Journal of Computational and Applied Mathematics*, 204(1), 10-17.

DOI: 10.1016/j.cam.2006.04.024

Choosing a leader on a complex network☆

R. Criado*, J. Flores, M.I. González-Vasco, J. Pello

Departamento de Matemáticas y Física Aplicadas y CC.NN., Universidad Rey Juan Carlos, c/ Tulipán s/n, 28933 Móstoles, Madrid, Spain

Abstract

In many real life applications a group of people interact through a communication network, mathematically modelled as a connected graph linking each element of the group. These participants may have diverse objectives and play very different roles depending on their knowledge and privileges. We focus on a particular scenario, in which a certain node is absolutely essential for completing the intended task. Moreover, if a technical failure results in disconnection of a participant to this leader node, this participant can no longer take part in the group's performance.

In this setting a sound choice of the underlying network topology could minimize the damage caused by random or provoked technical failures. We study different criteria for choosing suitable communication networks, from the point of view of both efficiency and robustness.

MSC: 90B18; 68P20

Keywords: Network efficiency; Network vulnerability; Communication network; Technical failure; Intentional attack

1. Introduction

Computer networks are nowadays not only mere highways through which information flows, but, moreover, they allow for the distributed performance of many tasks that can only be carried out in a cooperative way. When evaluating complex networks it is often assumed that all nodes in the underlying graph are equally important. This assumption is however not fulfilled in many practical applications. Also, it is often assumed that point to point communication is at hand (that is, participants are displayed in a complete graph). However, this is not very realistic and in any case expensive in terms of connections and computing resources.

We consider here a (somewhat more realistic) scenario, in which a certain task is carried out by a set of n participants, one of which, referred to as the *leader*, is essential for the success of the performance. For the sake of simplicity, let us assume the leader is to produce a certain value and distribute it to all other participants. Examples of such situations arise in many real life applications; message passing from a central server to a group of users, key distribution for private communication, etc. Moreover, we will assume that the underlying network is any connected graph and explore how the network topology may influence the final success of the protocol if technical failures may occur.

☆ Partially supported by PPR-2004-16 from Universidad Rey Juan Carlos.

* Corresponding author. Tel.: +34 91 6647445; fax: +34 91 4887338.

E-mail addresses: regino.criado@urjc.es (R. Criado), julio.flores@urjc.es (J. Flores), mariaisabel.vasco@urjc.es (M.I. González-Vasco), javier.pello@urjc.es (J. Pello).

Thus, our goal is to give reasoned criteria for, given a communication graph, singling out a node to represent a privileged participant. In addition, this criteria could also prove useful for comparing in this sense two such networks: that is, deciding which one will yield higher efficiency/robustness provided any of its nodes is eligible as a leader.

2. Preliminaries

Let us start by reviewing some well known graph parameters that could help motivating our study. So called *vulnerability* and *efficiency* of graphs are two concepts strongly related to the study of the structural properties of a complex network representing a system. The concept of vulnerability helps to measure the response of complex networks subjected to attacks on nodes and edges and it allows to spot the critical component of a network in order to improve its security. On the other hand, the concept of efficiency plays the role of measuring the ability of complex networks for the exchange and propagation of information on a global and local scale and its response for the spread of perturbations in diverse applications.

Intuitively, if G is a graph with a finite number n of nodes, the vulnerability $v(G)$ of G is a number $0 \leqslant v(G) \leqslant 1$ such that $v(G') \geqslant v(G)$ if G is obtained from G' by adding edges. So, the key property of a vulnerability measure is that it should never increase by adding edges. The rationale behind this assertion is that such an addition can only reinforce the structure of the graph.

Many definitions [1,7,9] are consistent with the intuitive idea of vulnerability and have been used in different contexts. However, as seen in [2] most of them fail to reflect intuitive properties of some concrete networks of practical relevance. Recently, in [2], the authors establish a new definition of vulnerability

$$v^{**}(G) = \exp\left\{\frac{\sigma}{n} + n - |E| - 2 + \frac{2}{n}\right\}, \tag{1}$$

where $|E|$ is the number of edges in the graph and $\sigma = ((1/n)\sum_{i=1}^{n}(gr_i - (2|E|/n))^2)^{1/2}$ is the standard deviation of the degree distribution, and they obtain its main properties, showing that this is an accurate and computable definition of network vulnerability which is directly connected with its topology.

A somewhat dual notion is that of the *efficiency* of a network. Smith [8], and Latora and Marchiori [6], introduced a definition of the *efficiency* in an attempt to reflect how fast information flows over a network. Therefore it is natural to assume that the efficiency ε_{ij} in the communication between node i and j is inversely proportional to the shortest distance between such nodes, and therefore it is natural to set $\varepsilon_{ij} = (1/d_{ij})$. With this definition we get $d_{ij} = \infty$ when there is no path in the graph between i and j, and consequently $\varepsilon_{ij} = 0$. In this fashion [6,4,5], we can find the definition of the efficiency of a graph G as

$$E^{+}(G) = \frac{1}{n(n-1)} \sum_{i,j \in G, i \neq j} \varepsilon_{ij} = \frac{1}{n(n-1)} \sum_{i,j \in G, i \neq j} \frac{1}{d_{ij}}. \tag{2}$$

The main drawback of this definition is that it does not reflect connectedness properties. In [3] the authors introduce another efficiency function $E^{\bullet}(\cdot)$ of a network G that is connection-sensitive

$$E^{\bullet}(G) = \left(\prod_{i,j \in G, i \neq j} \frac{1}{d_{ij}}\right)^{(1/n(n-1))}. \tag{3}$$

Other interesting results about this function can be seen in [3].

Even though the above notions provide sound guidelines for choosing a suitable communication network, they treat all network nodes equally and thus do not seem suitable for the kind of applications considered here.

3. Network selection in the presence of a leader

Let us now consider that we deal with complex networks designed for protocols where a single leader node generates a certain value which is handed on to the rest. In this setting, the topology of the underlying graph determines how the generated value reaches every participant. Recall that, as noted in the introduction, we assume that the failure of the leader, for causes either intentional or unintentional, completely prevents the protocol from succeeding. On the other

hand, if it is one of the other participants who suffers a failure, the impact of this failure can be measured by the number of other participants that will not be able to receive the computed value, as parties having another communication path will not be affected. In this sense, we will assume, from now on, that the leader is not subject to any kind of failure or attack, since, in such circumstance, the protocol would be totally disrupted no matter what the topology of the graph or the position of the leader within such topology.

Thus, our measure of resistance of a graph against an attack will be based on determining how well placed is the leader within the graph. A leader in a good spot will be such that the failure of any other node will not, on average, cause too many other nodes to be deprived from the computed value due to broken communications.

As in the previous section, let d_{ij} be the length of the shortest path connecting nodes i and j.

Proposition 1. *Let i be the leader on a given tree G. Then the average number of nodes disconnected from i due to the failure of a random node (other than i) is*

$$\frac{1}{n-1}\sum_{j\in G} d_{ij}.$$

Proof. Since we are assuming that all nodes (other than i) are equally prone to failure, the probability of any one of them failing is $1/(n-1)$, and then the average number of disconnected nodes is

$$\sum_{j\neq i}\frac{1}{n-1}n_j,$$

where n_j is the number of nodes "downstream" from j (including j). Precisely because G is a tree, there is a unique simple path from the leader i to any other node j, so we can define an ordering in G by $j_0 \preccurlyeq j_1$ if j_1 is "downstream" from j_0, that is, the shortest path from i to j_1 passes through j_0. With this ordering, the previous formula can be expressed as

$$\sum_{j\neq i}\frac{1}{n-1}n_j = \frac{1}{n-1}\sum_{j\neq i}\sum_{j\preccurlyeq k} 1 = \frac{1}{n-1}\sum_{k\neq i}\sum_{j\preccurlyeq k} 1.$$

Now the inner sum matches exactly the distance from the leader i to the node k, and the proof is done.

This way, the expression

$$\sum_{j\neq i} d_{ij}$$

becomes a measure of the vulnerability of the graph against random attacks. (The factor $1/(n-1)$ can be taken out since it is constant for a given number of nodes.)

In a similar fashion, we can give a measure of the efficiency of a graph with the (simple) formula

$$\max\{d_{ij} \mid j \neq i\}$$

that computes the maximum distance from the leader to any other node. Since the target value, once generated, must be transferred to the rest of participants following the underlying graph, this maximum distance gives us an idea of how long it takes for the constructed value to arrive to every destination.

We can, in fact, combine these two definitions into some kind of p-distance, in order to give them a similar and uniform treatment afterwards.

Definition 2. Let G be a tree graph and let i be one of its nodes. For every $1 \leqslant p < \infty$, we define

$$D_p(i) = \left(\sum_{j\in G} d_{ij}^p\right)^{1/p}.$$

For $p = \infty$, we define

$$D_\infty(i) = \max_{j \in G} d_{ij}.$$

With this definition, $D_1(i)$ becomes the vulnerability of the graph when the node i is chosen as leader, as seen before, while $D_\infty(i)$ is an inverse measure of the efficiency with said leader (a higher value means less efficiency). The measures $D_p(i)$ for $1 < p < \infty$ could intuitively be understood as "intermediate" measures, approaching $D_1(i)$ when p goes to 1 and approaching $D_\infty(i)$ when p goes to infinity, so that picking a certain $1 < p < \infty$ will result in a "balanced" measure that takes into account both efficiency and vulnerability to some extent. However, as we will see in the last section of this paper, this is not always the case for every possible network.

A good property that these measures have is that it can be proved that the node where the minimum is attained can be found rather easily, just by comparing adjacent nodes and keeping the best one.

Lemma 3. *Let G be a tree graph and let a, b and c be distinct nodes in G such that b is linked to both a and c. Then*

$$D_p(b) < \max\{D_p(a), D_p(c)\}$$

for every $1 \leqslant p \leqslant \infty$.

Proof. Since we assume that G is a tree, for any node x of G, exactly one of a, b or c will be the nearest node to x of those three, because cycles are not allowed. Thus, we can define three subtrees S_a, S_b and S_c of G given by

$$S_z = \{x \in G \mid z \text{ is the nearest to } x \text{ amongst } a, b \text{ and } c\},$$

where $z \in \{a, b, c\}$. Given the tree structure of G, the sets S_a, S_b and S_c determine a finite partition of G.

Pick now $z \in \{a, b, c\}$ and $x \in S_z$, and let α_x be the distance $\alpha_x = d_{xz}$. Assume that $1 \leqslant p < \infty$; then it is clear that

$$D_p(a) = \left(\sum_{x \in S_a} \alpha_x^p + \sum_{x \in S_b} (\alpha_x + 1)^p + \sum_{x \in S_c} (\alpha_x + 2)^p \right)^{1/p},$$

$$D_p(b) = \left(\sum_{x \in S_a} (\alpha_x + 1)^p + \sum_{x \in S_b} \alpha_x^p + \sum_{x \in S_c} (\alpha_x + 1)^p \right)^{1/p},$$

$$D_p(c) = \left(\sum_{x \in S_a} (\alpha_x + 2)^p + \sum_{x \in S_b} (\alpha_x + 1)^p + \sum_{x \in S_c} \alpha_x^p \right)^{1/p}.$$

If we consider the vector $\alpha \in \mathbb{R}^n$ as $\alpha = (\alpha_x)_{x \in G}$, the above values can be seen as the p-norms

$$D_p(a) = \|\alpha + (0, 1, 2)\|_p,$$
$$D_p(b) = \|\alpha + (1, 0, 1)\|_p,$$
$$D_p(c) = \|\alpha + (2, 1, 0)\|_p,$$

where (λ, μ, ν) stands for the vector in $\mathbb{R}^n$ that has a constant λ over the first $\operatorname{card}(S_1)$ coordinates, then μ over the next $\operatorname{card}(S_2)$ coordinates and finally ν over the last $\operatorname{card}(S_3)$ coordinates.

Now the triangle inequality of the p-norms yields that

$$\begin{aligned} 2D_p(b) &= \|2\alpha + 2(1, 0, 1)\|_p = \|2\alpha + (2, 0, 2)\|_p \\ &\leqslant \|\alpha + (0, 0, 2)\|_p + \|\alpha + (2, 0, 0)\|_p \end{aligned}$$

and the fact that α is in the positive cone of $\mathbb{R}^n$ gives

$$2D_p(b) < \|\alpha + (0, 1, 2)\|_p + \|\alpha + (2, 1, 0)\|_p = D_p(a) + D_p(c).$$

This forces $D_p(b)$ to be less that either $D_p(a)$ or $D_p(c)$, as we wanted.

The proof is similar if $p = \infty$. □

Proposition 4. *Let G be a tree graph and let $x_1, x_2, \ldots, x_m$ be distinct consecutive nodes in G. Then*

$$D_p(x_i) < \max\{D(x_1), D(x_m)\}$$

for every $2 \leqslant i \leqslant m - 1$.

Proof. This fact can be proved inductively on m by application of Lemma 3. □

This result leaves us with with a simple way of finding the optimum for a given D_p, which can be graphically described as "follow the slope". The algorithm begins by randomly choosing a node in the graph. Lemma 3 shows that at most one adjacent node can have a lower D_p, so we visit every adjacent node until we find one with a lower D_p. When that happens, we move on to that node and repeat the process with its neighbours. If no such node exists, Proposition 4 tells us that we have found a minimum.

Another nice consequence of Proposition 4 is the following.

Proposition 5. *Let G be a tree graph and let $1 \leqslant p \leqslant \infty$. Then there are at most two minima for D_p and, if there are actually two, they are adjacent.*

Proof. Let a and c be distinct nodes of G at which the minimum for D_p is attained,

$$D_p(a) = D_p(c) = \min_{x \in G} D_p(x).$$

If a and c were not adjacent, the path from a to c would contain (at least) another node b which, by Proposition 4, would satisfy

$$D_p(b) < \max\{D_p(a), D_p(c)\} = \min_{x \in G} D_p(x),$$

which is a contradiction. This proves that all minima must be adjacent to each other; since G is a tree, this is impossible for more than two nodes.

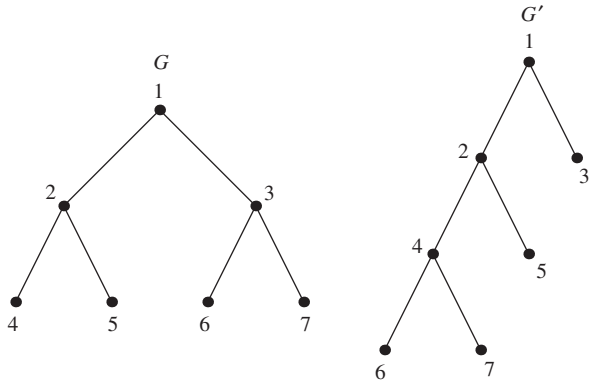
These measures provide more information than those already known in the literature (see Section 2), since, as mentioned, they take into account the fact that now a node in the graph plays a distinguished role, while the preexisting measures give no special relevance to any node.

For instance, in the simple case of a star graph of n nodes S_n, the function v^{**} recalled in Section 2 converges to 1 as n grows to infinity. On the other hand, for the line tree of n nodes l_n we have that $v^{**}(l_n)$ converges to e^{-1} as n grows. Thus, roughly speaking, v^{**} indicates that line trees are more robust than stars.

Intuitively, however, if we are to place an "invulnerable" leader at a certain node, it is clear that a star configuration is much more preferable, as disconnection of any other node only affects itself. In contrast, a failure on any node of a line tree will always bring about the disconnection of several participants.

The method set forth above shows that for a given tree with an invulnerable leader the D_1-measures on the different nodes help us to choose an optimal location for the leader. Another related problem is the following: assume that we have two possible tree networks for a given number of nodes and that we must select one of them and then pick a leader

in it. Which one is to be preferred from the point of view of robustness? We want to show how the D_1 measures can play a useful role in helping us decide. To make our point clear we will consider the following two trees G and G'.



The vulnerability function v^{**} was designed to treat this problem (at a more general level) and yet seems of little help in the present situation. Indeed if we calculate the v^{**}—vulnerability of both G and G' we obtain the same value. This is certainly to be expected since G and G' have the same number of nodes and their corresponding incidence degrees coincide. Thus, according to v^{**}, there is no reason why one should choose G over G', or conversely.

The use of D_1-measures can shed some light about why we should prefer G' over G. Indeed, if $D_1^G(i)$ and $D_1^{G'}(i)$ represent the D_1-measures of the node i for G and G', respectively, we get the following values:

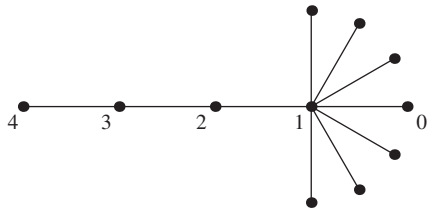
$$D_1^G(1) = 10, \quad D_1^{G'}(1) = 12,$$
$$D_1^G(2) = 11, \quad D_1^{G'}(2) = 9,$$
$$D_1^G(3) = 11, \quad D_1^{G'}(4) = 10,$$
$$D_1^{G'}(5) = 14.$$

Proposition 4, along with the algorithm described afterwards, prove that node 1 is the best possible leader for graph G, while node 2 is the best one for graph G'. So if we are to choose one of these networks and then a leader within that network, we should select G' and then take node 2 as the leader, as it has the lowest D_1 vulnerability of all possibilities. In this sense, the measures D_p can be turned from a local point of view, in which they gauge how good a node is if chosen as leader, into global comparison functions of network topologies, so we could say that a graph is better than another one when the best leader for the former graph gives a lower D_p than the best leader for the latter.

4. Behaviour of D_p for different $p \in [1, \infty]$

So far we have focused in studying the properties of the measures D_p for a fixed p, that is, we pick some $p \in [1, \infty]$ and then decide which is the best node in a graph for that p, or compare two different graphs according to their respective best node choices. Now we are going to take a look into how these measures behave when p changes.

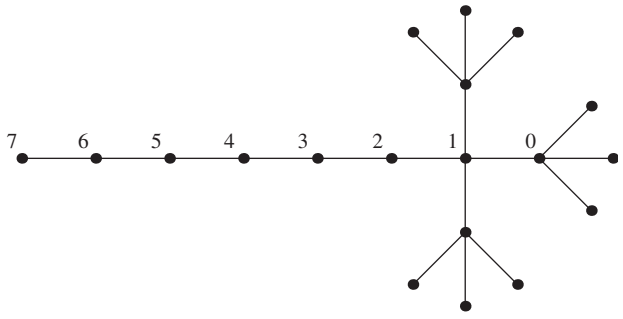
A first observation that must be made is that the optimum node in a graph need not be the same for different values of p. This is best shown with an example. Consider the following graph.



If our aim is to minimize the expected loss of communication in the event of a node failure, and pay no attention to efficiency ($p = 1$), it is easy to check that the best choice for a leader is node 1 ($D_1(0) = 22$; $D_1(1) = 13$; $D_1(2) = 18$);

intuitively, this is so because then at worst three nodes (2, 3 and 4) will be disconnected with a single failure, while for any other leader the failure of node 1 itself would cause massive disruption. On the other hand, if we are absolutely certain that no node can ever fail and therefore want to maximize efficiency ($p = \infty$), the best place for the leader is node 2, since that reduces the longest distance to two hops ($D_\infty(1) = D_\infty(3) = 3$; $D_\infty(2) = 2$).

Actually, in many cases this is what happens when the optima for $p = 1$ and $p = \infty$ do not coincide. Assume that the optimum for $p = 1$ is some node x_1, and that the optimum for $p = \infty$ is another node x_n. Since we are dealing with trees, there will be a unique simple path from x_1 to x_n, which we will call $x_1, x_2, \ldots, x_n$. Then x_1 will not only be the optimum for $p = 1$ but also for a whole interval $[1, p_1]$, where $1 \leqslant p_1$; after that, the optimum will usually switch to x_2 for another interval $[p_1, p_2]$, then to x_3 and so on until the optimum node is finally x_n for all p in the interval $[p_{n-1}, \infty]$. The "switching points" $p_1, p_2, \ldots, p_{n-1}$ are exactly those at which there are two simultaneous optimum nodes; for the rest of values, the optimum is unique. Let us see an example that illustrates this situation:



First of all, it is easy to see that $D_\infty(2) = D_\infty(4) = 5$ and $D_\infty(3) = 4$, so node 3 is the optimum leader for $p = \infty$. For other values of p, it can be checked that

$$D_p(0)^p - D_p(1)^p = (7^p - 2^p) + 6(3^p - 2^p) > 0,$$

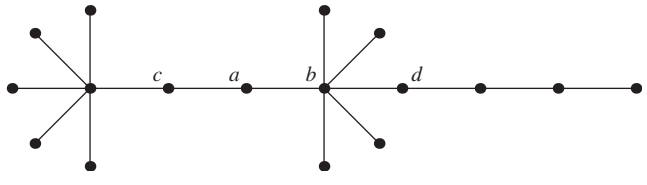
so node 1 is always a better leader than node 0 (or any of the other nodes with which it is symmetric, for that matter); on the other hand,

$$D_p(4)^p - D_p(3)^p = 9(5^p - 4^p) + 2(4^p - 3^p) > 0,$$

so again node 3 is a better leader than node 4 for any $p \in [1, \infty)$, following Proposition 4. What we are going to see is that the optimum moves from node 1 for $p = 1$ to node 3 for $p = \infty$, being node 2 for an intermediate range of p. Indeed, the expression $D_p(1)^p - D_p(2)^p$, as a function of p, is negative up to a value p_1 which numerical computations find close to 2.804, and it is positive afterwards, and the same happens to the function $D_p(2)^p - D_p(3)^p$ with respect to some p_2 close to 9.656. This means that the optimum is node 1 for values of p in $[1, p_1]$, then it switches forth to node 2 up to p_2 and afterwards it finally becomes node 3. At the "switching points" p_1 and p_2, there are simultaneously two best leaders.

In this situation, intermediate values of p give a leader node that, while possibly not the absolutely best choice vulnerability-wise, may still have good robustness properties and also pay some attention to efficiency. In this example, for instance, choosing node 2 instead of node 1 will result in an efficiency improvement at the cost of robustness (since we are left with the chance of a failure at node 1 causing the disconnection of 12 other nodes); the final decision must always be made taking into account factors such as the probability of failure or the cost associated with long communication paths. This compromise between robustness and efficiency can then be transferred into the model by way of p, with smaller values of p giving more weight to vulnerability and larger values focusing on efficiency.

However, there are other networks in which the ideal leader does not behave in such a monotone way with respect to the choice of p, so the previous consideration does not apply. Take for example the following graph:



If we take $p = 1$ or $p = \infty$, we have that

$$D_1(a) = 41, \quad D_\infty(a) = 5,$$
$$D_1(b) = 40, \quad D_\infty(b) = 4,$$
$$D_1(d) = 49, \quad D_\infty(d) = 5,$$

so node b is the optimum for both vulnerability and efficiency. On the other hand, if we take $p = 2$, the results are quite different:

$$D_2(c) = \sqrt{148},$$
$$D_2(a) = \sqrt{121} = 11,$$
$$D_2(b) = \sqrt{128}.$$

In this case, then, we cannot say that a value of p in the interval $(1, \infty)$, such as $p = 2$, provides a leader that keeps a balance between robustness and efficiency; if it did, it should lead us to node b anyway, since it is the optimum for both criteria.

This rather simple graph shows us that we cannot blindly assume that taking intermediate values of $p \in (1, \infty)$ implies finding a leader node with a compromise between robustness and efficiency. In the example we had seen before, however, this was exactly what happened, and such a p was a good way of making a trade-off between these parameters. The problem remains open to give conditions for a graph to have this behaviour.

References

[1] Y. Bar-Yam, Dynamics of Complex Systems, Addison-Wesley, New York, 1997.

[2] R. Criado, J. Flores, B. Hernández-Bermejo, J. Pello, M. Romance, Effective measurement of network vulnerability under random and intentional attacks, Journal of Mathematical Modelling and Algorithms 4 (3) (2005) 307–316.

[3] R. Criado, A. García del Amo, B. Hernández-Bermejo, M. Romance, New results on computable efficiency and its stability for complex networks, Journal of Computational and Applied Mathematics 192 (1) (2006) 59–74.

[4] V. Latora, M. Marchiori, Efficient behaviour of small-world networks, Phys. Rev. Lett. 87 (2001) 198701.

[5] V. Latora, M. Marchior, Economic small-world behaviour in weighted networks, Eur. Phys. J. B 32 (2003) 249–263.

[6] V. Latora, M. Marchiori, How the science of complex networks can help developing strategies against terrorism, Chaos, Solitons and Fractals 20 (2004) 69–75.

[7] M.E.J. Newman, The structure and function of complex networks, SIAM Review 45 (2003) 167–256.

[8] J.E. Smith, Characterizing computer performance with a single number, Commun. ACM 31 (1998) 1202–1206.

[9] S.H. Strogatz, Exploring complex networks, Nature 410 (2001) 268–276.