STRUCTURE AND DYNAMICS OF COMPLEX SYSTEMS

A thesis submitted for the degree of Doctor of Philosophy

by

Bernard Kujawski

School of Information Systems, Computing and Mathematics,
Brunel University, Uxbridge, Middlesex, UB8 3PH, United Kingdom

September, 2008

Abstract

A family of navigation algorithms for packet transport in complex networks is introduced. These algorithms use deterministic and probabilistic rules which depend, in different ways, on the degree of the node, packet flow and the temporal properties of packet delivery and distribution. On scale-free networks all our algorithms can handle a larger load than the random walk algorithm. I examined the fluctuation properties of packet traffic on scale-free networks and random graphs using random diffusion and a locally navigated diffusive motion with preferred edges. I found that preferential behaviour in either the topology or in the dynamics leads to the scaling of fluctuations of the number of packets passing nodes and the number of packets flowing along edges, respectively. I showed that the absence of any preference results in the absence of scaling. Broad distributions of the return times at nodes and edges illustrate that the basis of the observed scaling is the cooperative behaviour between groups of nodes or edges.

I presented an empirical study of the networks created by users within internet news groups and forums and showed that they organise themselves into scale-free trees. The structure of these trees depends on the topic under discussion; specialist topics have trees with a short shallow structure whereas more universal topics are discussed widely and have a deeper tree structure. The correlation function of activity shows long range correlations connected

with the users' daily routines.

I presented an analysis of empirical data on the arrival and discharge times at a UK Accident and Emergency (A&E) department. I found that discharges rates vary with the workload and that the distribution of the length of stay has a fat tail. A sand pile model is introduced to show that the A&E department is a driven self-organised system. In my model I used a variable input space to mimic the queuing discipline related to different types of patients presenting to the department.

Acknowledgements

I would like to thank my supervisor, **Geoff J. Rodgers**, Pro-Vice-Chancellor (Research) at Brunel University for his inspirational help and support through my studies. He has opened my eyes to bold and long-sighted projects.

This work would not have been finished without the help of **Bosiljka Tadić**. Her endless patience and hours of explanations to all my questions increased my insight into complex systems.

I am grateful to my former supervisor, **Janusz Hołyst**, without whom I would not be where I am now. He was the first who has excited my curiosity in complex networks.

I want to thank all my friends that gave me support, inspiration and a lot of fun at Brunel University, especially Pierpaolo Vivo and Rodrigo Megaides. I will never forget great guys, who left something special in my mind and were a part of my greatest holidays in Ljubljana: Milovan Šuvakov and Zoran (Kiza) Levnajic.

And finally, I thank my beloved wife **Marta** for taking care of me, for her love she was sharing with me each day we spent here together.

Publications

Chapter 1

- *Local Information Based Algorithms for Packet Transport in Complex Networks* B. Kujawski, B. Tadić, and G.J. Rodgers, In V.N. Alexnandrov et al., editor, ICCS 2006, volume **3993** of Lecture Notes in Computer Science, pages 1024-1031, Berlin, 2006, Springer.

Chapter 2

- *Preferential behaviour and scaling in diffusive dynamics on networks* B. Kujawski, B. Tadić and G. J. Rodgers, Preferential behaviour and scaling in diffusive dynamics on networks, New J. Phys. **9**, 154 (2007).

- *Origin of Scaling on Networks, Structural Inhomogeneity and Preference in Dynamical Behaviour* B. Kujawski, B. Tadić, and G.J. Rodgers, Origin of scaling on networks, structural inhomogeneity, and preference in dynamical behaviour , Proc. SPIE **6601**, 660107 (2007).

Chapter 3

- *Growing trees in internet news groups and forums* B. Kujawski, J. Hołyst and G. J. Rodgers, Growing trees in internet news groups and forums, Phys. Rev. E **76**, 036103 (2007).

Chapter 4

- *Self Organised Criticality in an Accident and Emergency Department* A. Hellervik, B. Kujawski, G. J. Rodgers and T. P. Young, to be published.

Contents

List of Figures

LIST OF FIGURES

Introduction

Contemporary scientists have to move outside their original realm of studies and challenge themselves with multidisciplinary problems. We observe today a trend to cross the edges of our original scientific background in multidisciplinary teams, tackling problems spanning natural and social science and studying broad models with applications in such different subjects as physics, biology, finance and social psychology.

One of these models is without doubt the network model, which considers the structure built up by the constituents (nodes or vertices), and the interconnections between them (links or edges). The internal features of both elements are usually not studied and the theory is focused on the *structure* and *function* of networks. Random graphs considered by Erdős and Rényi [1] were the first approximation to real networks. However, it took almost 40 years for scientists to find that the real networks are much more sophisticated than the random graphs. This was due to the development in computing and capacities of databases that allowed researchers to grapple with large real datasets. The access to them resulted in the rapid growth of studies of the real networks, such as World Wide Web (WWW) [2, 3, 4, 5, 6, 7], Internet [8, 9, 10, 11], movie actor collaboration network [12, 13, 14, 15, 16], science collaboration graph [17, 18, 19, 22], the web of human sexual contacts [23], cellular networks [24, 25, 26], internet discussion

networks [27, 28, 29, 30, 31], phone call networks [32, 33, 34], social networks [36], and gene networks [35].

The results obtained from these works have shown that real networks are not regular nor random at all. This is quite surprising that the most ubiquitous type of network is a scale-free network. It means that in most cases a real network observed in different scales is similar to itself (self-similarity is a well known property of fractals [38]). This non-trivial topology has interested many scientists who breathed a new life into the graph theory and developed it into the modern theory of networks.

There are a few factors supporting the popularity of the theory of networks, but first of all one has to mention that an enormous number of real systems can be represented as a set of nodes and links. A link between two nodes can represent a link existing in reality (Internet on the router level) or a virtual link (a hyperlink in World Wide Web page or a relation between species in a foodweb). Second, classic physical models are suitable for regular structures and particles. Applied to finance, economics or social sciences they use mostly a statistical approach where each constituent is on average identical, such as an agent based model. The network model is a powerful tool, which can handle irregularity, individualisation of constituents and locally oriented connectivity. Third, the basic concepts of properties such as a preferential attachment [49], growth and evolution [87], small-world [37, 12], clustering [49], assortativity and disassortativity [20, 21] can be adopted in a variety of fields, spanning from genetics [35] to the internet [8, 9, 10, 11].

The study of the theory of networks is divided into two main groups: the study of structure and the study of dynamics on networks.

The most important characteristics of a network topology is its degree distribution, where a degree is a number of links connected to a single node.

INTRODUCTION

We say a network is scale-free when its degree distribution is power law $p(k) \sim k^{-\alpha}$. A random graph is described by a Poisson distribution. Real networks are often enormously large. Thus, useful characteristics were found to describe their properties, such as the length of the shortest path, clustering coefficient, dimension, motifs, centrality, betweenness or assortativity. For example, the last parameter, assortativity, does a great job separating human networks from all other types. It is characterised by a Pearson coefficient, which in the case of human networks is positive and negative in all other cases [20, 21]. The study of structure helps us to classify networks, however the study of growth and evolution models [49, 88, 57, 51] give us insight into the processes leading to a certain type of network. It is fascinating that very elementary ingredients such as the growth and preferential attachment [49] lead us to such complex result as a scale-free network.

The behaviour of dynamical processes taking place on networks is very complicated. First of all there are many different types of dynamics such as random walk, navigated diffusion, percolation or transport. Second, for most of them the results depend on the type of underlying network. Thus, scientists focus mostly on a very elementary process (random walk) or study dynamics for a representative type of a network (random graph or Albert - Barabási model). Large improvements in our understanding of the dynamics of networks were made for diffusion [41], navigation [62, 65, 66], jamming[42, 62], transport[73], noise and fluctuations [79, 80, 81, 82, 83, 84, 85, 86].

This thesis focuses mostly on complex networks and three chapters of it are devoted to this subject. First of all we focus on a transport process on networks and the influence of a navigation algorithm on its properties. For the transport we understand the process of sending any type of item from a place of origin to a place of destination. In networks these places

are nodes and the transport process takes place along links. An item stands typically for an idea, particle, charge or a packet and the name refers to the real process, which is considered. Here, we will describe it as a packet.

A random walk is the basic navigation algorithm and is often used as a reference for other algorithms. It describes a diffusive dynamics of a packet from an origin to the destination. This transport process is not optimised in any way. Thus, all transport properties depend on the type of the underlying network. All other algorithms have optimised performance, such as shortest delivery time or jamming threshold. They can be divided into global and local ones. Global algorithms are optimised in the sense of finding minimum or maximum value of desired property for the whole network, for instance the minimum number of hops between two nodes or the maximum network load. In the case of local algorithms we optimise the network performance in a local scale. For instance, we find a node in the neighbourhood of node i with the highest connectivity and thus, we increase chances of a walker to reach the destination faster. The local minimum or maximum can overlap with the global one but it does not overlap in general and the local navigation algorithms are not as good as the global ones. However, there are cases where we are unable to retrieve information about the whole system and the global algorithms cannot be used. Moreover, even in the case of complete knowledge about the system, its size can limit an implementation of some algorithms and this is due their computational complexity. Because of that, local navigation algorithms can be helpful in both cases. They explore usually the nearest neighbourhood of a node, which can be assumed as knowledge of a constituent about the whole system. Thus, these algorithms optimise a subsystem of size $n_i = k_i$ for $i = 1, 2, \ldots N$ which is typically much smaller than the size of the network and therefore much faster to compute. In this way, despite of

their limitations, local navigation algorithms in some applications are worth considering. In Chapter 1 we investigate four local navigation algorithms.

The basic characteristic that describes a transport process on a network is the load time series. This is a number of packets that are processed in the network in given time t. The average value of load stands for the network capacity. When the load increases in time constantly, it indicates that the network is jamming. For most navigation algorithms, the power spectrum of the load follows a power law relation with exponent β ranging from -2 to -1, which stand for no correlations (diffusive transport) and long-range correlations, respectively. The power spectrum of a load time series is very similar to a power spectrum of noise. For instance, in acoustics the power spectrum of a signal time series described by $P(f) = af^{-2}$, where f is a signal frequency, is called the *brown noise* and is characteristic for the sounds of nature such as a waterfall. Because of this similarity the network load is often called the noise.

This property focuses on the network on a macro scale. But one can ask a question about the load time series for each network node. This approach is called multichannel analysis and is well known in discrete systems, where each part of the system can be distinguished. For instance, it was applied for agent based model and to the stock market [84]. For each noise time series recorded at node i one can obtain the average number of flowing packets $< h_i >$ and relate it to the dispersion σ_i. We would like to stress here that we use the term dispersion in sense of a standard deviation. This terminology is commonly used in the literature following one of the first works in this subject [79]. If you plot σ_i against $< h_i >$ for each node i you will obtain a scaling relation $\sigma \sim < f >^{\mu}$, where $0.5 \leq \mu \leq 1$. Chapter 2 is devoted to the origin of this scaling relation.

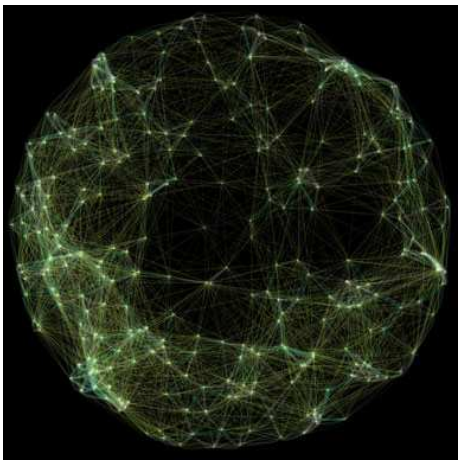
The internet gives us an unprecedented opportunity to study the interaction of human beings. This comes mostly because of the possibility to track internet users' activity through internet databases and web pages. Nowadays, there are plenty of ways in which people communicate with each other such as emails, Skype, Facebook, news groups or forums. Particularly, the last two methods are very interesting because of the open access to them and the huge number of users. We wrote a computer program which enables us to download these internet discussions from both sources. First of all, we chose the biggest Polish news portal *www.onet.pl*, where forum users are very active. The second internet discussions source was a students' news group, available only to students of Warsaw University of Technology. We analysed these systems as a network of messages sent by users. These discussions form the tree-like structures, where a topic of each discussion is a root node. In Chapter 3 we investigate in detail their structures and temporal distributions as well as the activity of the internet discussions' users.

In Chapter 4 we move outside the scope of the network theory and we deal with the behaviour of hospital personnel in an A&E department. We focus on two characteristics of this system: the discharge / arrival rate against number of patients in the department and the length of stay distribution. The former is a linearly growing curve indicating that the discharge / arrival rate depends on the workload. This means the department staff adjusts itself to the number of patients in the department. This property places the A&E department within the scope of self-organised systems. Furthermore, the fat tail of the length of stay distribution supports remarkably this claim. Thus, we decided to model the behaviour of personnel at the A&E department using a sand pile model, which is a well known example of the self-organised system. However, we modified this model by introducing an idea of the

variable input space to cope with the different cases of emergency observed at the A&E department.

Chapter 1

Navigation on Networks



We introduce four algorithms for packet transport in complex networks. These algorithms use deterministic rules which depend, in different ways, on the degree of the node, the number of packets posted down each edge, the mean delivery time of packets sent down each edge to each destination and the time since an edge last transmitted a packet. On scale-free networks all our algorithms are considerably more efficient and can handle a larger load than the random walk algorithm. We consider in detail various attributes of our algorithms, for instance we show that an algorithm that bases its decisions on the mean delivery time jams unless it incorporates information about the degree of the destination node.

1.1 Introduction

Complex networks can be used to model a wide range of physical and technological systems. One of the most interesting dynamical problems on networks is transport, which can give us some insight into the transport of information in technology based communication networks like the internet [8], the World Wide Web [2],[39] or phone call networks [40]. Here we use the term *transport* to mean the transport of particles, which are packets in a network. Thus our model falls within the Network Layer of the OSI Reference Model and the algorithms described in section 3 are routing algorithms that belong to the Network Layer of the OSI Reference Model. Of particular interest is the phenomenon of load in a network, as a function of the rate of packet creation R, which has been investigated for models of communication networks [42],[43],[44] and in real networks [45].

Typically the problem of transport is investigated using either a random walk algorithm [42], or the shortest path algorithm used by most internet protocols. The difficulty with these approaches is that random walk algorithm is very inefficient for transport in technology based communication networks and shortest path algorithm requires, for its implementation, information about all connections in a network.

Thus, we would like to study in this chapter the local navigation algorithms, that could benefit from a simple local network structure and display an efficiency much greater than the random walk algorithm. Particularly, we would like to explore the local mean delivery time property. We assume that the shortest paths are also the quickest ones. Thus, when a node sends a packet it should choose one of its neighbours with the shortest mean delivery time. We believe that this property can be useful in creating an efficient navigation rule.

One can argue that a mean delivery time of node i is not a local property. This is due to the fact that node i needs all information from the destination nodes about the arrival times of the packets. But this information can be passed to node i also by a packet. This demand is not an artificial one, for instance in the TCP internet protocol a destination router sends back to a sender information about the correctness of received data. In our algorithms it would be information on the time of delivery.

In this paper we focus on algorithms that use local information about the topology, along with information about the flux of packets between neighbours, the link load and the time taken to deliver packets. We propose four algorithms that use some or all of these properties to deliver packets in a network.

In section 2 we describe the algorithm that we use to perform numerical simulations of our models. In section 3 we discuss the algorithms that packets use to find their destinations and in section 4 we show our results. In section 5 we summarise our results.

1.2 The Program

A program was written to simulate packet transport on a network that does not depend on the size of the network or its topology. At the beginning of the program an external file with the adjacency matrix of the network is read in. We focus on the internet and consequently we treat nodes in our network as if they are routers. The connections between the routers have the same capacity for all networks. Such a model can not only be used to model internet packet transport but also for a range of transport networks in which the nodes have local routing information.

Each node:

- Generates a new packet with probability $r = R/N$ and with a randomly chosen destination, where R is a fixed rate for the whole network, and N is the number of nodes in network.
- Stores packets in a queue, which has maximum length of $L = 1000$. Packets are despatched from the queue in a first in, first out (FIFO) order.
- Sends packets to its neighbours.

Each node has information about:

- The address of all its neighbours (they have unique indices j).
- The degree of its neighbours - $k(i)$.
- Flow through all its neighbours, which is measured by
  - The number of packets posted down each edge to neighbour i - the Link Load - $C(i)$.
  - The number of packets sent through neighbour i, which have reached their destination - $N_P(i)$.
  - The sum of the delivery times of all the packets sent through neighbour i that have reached their destination - $T_P(i)$.
  - The time interval since an edge last transmitted a packet to neighbour i and current time step - $\Delta T(i)$.

The index i enumerates each neighbour of node k and each node keeps all the statistics about its neighbours. Quantities $C(i)$, $N_P(i)$, $T_P(i)$ and $\Delta T(i)$ describe node i from the perspective of node k. Each node is described by its

neighbours and all properties can be different for all neighbours that describe node i.

The initialisation part of the program sets up the network topology, the nodes and all the tables used by them. Inside the main loop a time step is incremented, and within that a loop over all nodes calculates and updates the statistics. The loop over all nodes includes three basic routines, which are run for each node; generating new packets, checking its queue for packets with its address and sending packets to its neighbours. Each node generates a packet with a randomly chosen destination with probability R/N. The node checks its own queue for packets addressed to itself. When it finds one of these it deletes it from the queue and updates the statistics $N_P(i)$ and $T_P(i)$ for all the nodes on the packet's path. Each packet keeps track of its own path. The node sends packets to its neighbours by taking the first packet in its queue and checking the packet's destination address. If the packet is addressed to one of its neighbour, the node will send it to the neighbour. If it is not, the node will use the *algorithm* to find where to send the packet. During this posting step the $C(i)$ property is updated. When node k sends packets to node i, the number of sent packets $C(i)$ increases. After this loop over all the nodes is completed, the quantities $\Delta T(i)$ and the mean delivery time of packets sent down each edge $N_P(i)/T_P(i)$ are updated for all nodes.

1.3 Algorithms

The most important element in transport is the rule that determines the direction in which a packet is sent. A transport network without a rule is a random walk network. We call this rule the *algorithm*. It describes how nodes deal with packets and should help packets to get to their destination.

Not all algorithms help packets to reach destinations, poor algorithms can easily be worse than the random walk algorithm. All algorithms considered in this paper work with deterministic rules.

As we mentioned before, we are mainly interested in exploring potential application of the local mean delivery time property. In our basic algorithm when a node k is attempting to send a packet it finds one of its neighbours with the smallest mean delivery time $S_k = \min \left[\frac{T_P(i)}{N_P(i)}\right]_{i=1\ldots n}$. To verify its properties we have run a set of simulations for this algorithm and we have found that a network is easily jamming even for a small value of the input rate R. The reason for that is twofold. First of all, links that were found in the initial phase of the simulation as inefficient (i.e. very long time of the first delivery) have no possibility to improve their performance. Secondly, large nodes are usually better in delivering packets and their mean delivery time is smaller than for small nodes. This property in connection with deterministic rule leads to the state when two large nodes send packets only between themselves, trap packets and jam the network. These two side effects of applying the local mean delivery time lead us to the inclusion of another properties and creation four navigation rules that will cope with the problems described above and keep our algorithms deterministic.

The *shortest time*(ST) algorithm is our basic algorithm that uses information about the mean delivery time $T_P(i)/N_P(i)$ and the time interval between the last packet that came to node i and actual time step. The ST algorithm finds the minimum value

$$S_k = \min \left[\frac{T_P(i)}{N_P(i)} \frac{1}{\Delta T(i)}\right]_{i=1\ldots n} \tag{1.1}$$

in order to determine which node to send the packet to. The idea of this algorithm is to try and find the minimum travel time for each packet between source and destination. At the start of the simulation S is equal to

0 for all neighbours. Because the update of $T_P(i)/N_P(i)$ only occurs when a packet arrives at its destination, it can take a number of time steps before $T_P(i)/N_P(i)$ becomes non-zero. The inclusion of the reciprocal of $\Delta T(i)$ in S is a response to the first problem described above, it ensures that the algorithm does not get into a state where it never sends a packet down certain links which have a large mean delivery time. This state is particularly likely to occur at the start of the simulation. The inclusion of the reciprocal of $\Delta T(i)$ in S also prevents overcrowding when a node finds a node which is clearly better than all its other neighbours. Hence, because of the inclusion of $\Delta T(i)$ more nodes take part in the transport and in this way the large node does not become overcrowded. Because the algorithm with $T_P(i)/N_P(i)$ is looking for minimum delivery time we call it the *shortest time* (ST) algorithm. To start this algorithm, and the STD algorithm, which we will introduce shortly, we use the random walk algorithm. This is due to initial values of the mean delivery time which are set to 0 in the beginning. Once all values in the neighbourhood of node i are greater than 0 (at least one delivered packet through each neighbour), node i starts to use an appropriate algorithm. This means in the initial stage the random walk and the algorithm coexist, but in the time scale of our simulations the influence of this stage can be neglected. Without this initial random walk procedure both the ST and the STD algorithms would jam almost immediately.

Our computer simulations have shown that the inclusion of $1/\Delta T(i)$ helps only for small values of the input rate R. The effect of overcrowded large nodes is present and is still the main problem to overcome. Thus, we introduce to our algorithm information about the local topology, the neighbours' degrees. This idea of incorporating information about the degree of nodes in the transport algorithm was discussed in [46] and [47]. In these papers

models were introduced in which nodes were selected at a rate proportional to a power of their degree. It was found that the most efficient algorithm was one in which the probability of selecting a node of degree k was proportional to $1/k$ [46, 47].

The *shortest time and degree* (STD) algorithm is a modification of the ST algorithm. It uses information about a neighbour's degree which helps packets avoid the nodes with the largest degree. The STD algorithm is defined by

$$S_k = \min \left[\frac{T_P(i)}{N_P(i)} \frac{1}{\Delta T(i)} k(i) \right]_{i=1\ldots n} \tag{1.2}$$

where $k(i)$ is a degree of node i and $k(i) > 1$. This last assumption allows the algorithm to avoid dead-end nodes. A node with degree $k = 1$ can only receive a packet that is addressed to itself. The STD algorithm uses both temporal properties and also information about the local connectivity. For transport in a scale-free network the most important nodes are those with the largest degree. But because their neighbours send these nodes a large number of packets, the queues at these nodes can become overcrowded. Information about the degree helps the algorithm to avoid these nodes, but it does not mean that they are not used.

We introduce here the CDT algorithm, *connections, degree and shortest time*, which incorporates information about the link load $C(i)$. Thanks to this property we can avoid the random walk initial procedure. It is similar to STD algorithm, it tries to minimise the delivery time and benefits from the degree property to avoid overcrowding. The CDT algorithm is defined by

$$S_k = \min \left[\frac{T_P(i)}{N_P(i)} \frac{1}{\Delta T(i)} C(i) k(i) \right]_{i=1\ldots n} \qquad \text{with } k(i) > 1. \tag{1.3}$$

For the CDT algorithm we begin at $\frac{T_P(i)}{N_P(i)}\frac{1}{\Delta T(i)} = 1$.

We are going to test the algorithms described above by comparing them to an algorithm without the mean time delivery property. We call it *connections and degree* (CD) algorithm and it is defined by

$$S_k = \min[C(i)k(i)]_{i=1\ldots n} \tag{1.4}$$

where $C(i)$ is a number of packets that node k sends to node i. CD algorithm uses information about the link load $C(i)$ and the degree, however the link load property is used here only to make it deterministic implementation of the algorithm described in [46, 47]. That very simple navigation rule was designed to avoid large nodes, which are mostly responsible for jamming. In that algorithm a node i sends a packet to one of its nighbours with the probability $p \sim 1/k(j)^{\beta}$, where $k(j)$ is a degree of its neigbour. Our CD algorithm represents the case for $\beta = 1$, which was found to be most efficient in dealing with jamming in the scale-free networks. The probabilistic version of of CD algorithm called D algorithm is used in Chapter 2. The CD algorithm does not need the initial random walk procedure, in the beginning S equals 0 and $C(i)$ is updated almost immediately. For this navigation rule there is no property that can be minimised, unlike in the ST and STD algorithms where the delivery time is expected to minimise.

Additionally, we compare performance and efficiency of our navigation algorithms with the random walk and the shortest path algorithm. We use Dijkstra's algorithm [48] to find the shortest paths between nodes in the network and all packets are navigated through them from the origins to their destinations.

We use the learning property to describe behaviour of an algorithm in the beginning. By learning we mean the proportion of links whose value of S has changed since $t = 0$. The CD and CDT algorithms learn the most quickly.

After 5000 time steps they tried 95% of the links. This is because the link load, $C(i)$, changes when a packet is sent down it whereas $T_P(i)/N_P(i)$, used by the ST and STD algorithms, only changes when a packet sent down it gets to its destination. That is why the ST and STD algorithms need the random walk starting procedure. With this procedure after 5000 time steps 35% of links were tried. For the ST algorithm without the random walk starting procedure it was 5%. As we have mentioned before for ST and STD, a node i starts to use a navigation rule once all its links have been used at least once. Thus, in the initial state two navigation rules operate effectively, the random walk and a navigation algorithm. However, our simulations are long enough to ensure there is no influence of the initial state for our findings. The speed of learning is important because when a network learns slowly, the network only uses a small proportion of its links for transport over a long period of time, which means that the network is easily jammed when a region of the network becomes overcrowded.

1.4 Results

We consider transport on the Barabási and Albert model of a network [49] with $N = 1000$ nodes and $m = 2$. The parameter m is the number of links of a new node that are added to network. When $m = 2$ the network includes loops and has relative small number of connections. Our research shows that this network jams for lower values of the posting rate than networks with $m = 1$ or $m = 3$ and higher. When it is necessary we support our analysis and conduct additional simulations on the same type of network with $N = 500$ and $N = 100$ nodes to reveal important size effects. In this work we use a posting rate of $R = 0.1$. This means that each node creates

a packet with probability R/N. We use one value of the posting rate R for all algorithms. Hence, the particular choice of $R = 0.1$ is set by the free flow regime of the least efficient random walk algorithm. Moreover, we are not going to study here the queuing effect and we do not want to disturb our studies by the effect of short queues. Thus for the queue length $L = 1000$ we choose $R = 0.1$ to be sure that queues are not overcrowded throughout a whole simulation. However, one should be aware that some additional effect for the properties studied here may occur for higher posting rate and different network sizes.

The number of time steps for our simulations is $6M$. Due to the long transient time of the STD algorithm we extended its simulation up to $8M$ time steps to confirm its stability. We present results for the STD, CD and CDT algorithms. We do not consider the ST algorithm any further because it is not stable and easily jams, even for a very low posting rate R.

In figure 1.1 we show the load time series, which is defined as the number of packets that are in the network at time step t. The figure presents results for the final $500,000$ time steps of the simulations for the STD, CD and CDT navigation algorithms. All three algorithms are stable, however the load for CDT algorithm displays larger deviations from the mean value than the other two algorithms. In table 1.1 we compare the load mean values and their standard deviations. In the case of STD, CD and CDT algorithms, the STD has the smallest mean load value. As we might expect from Fig. 1.1, the standard deviation for CDT algorithm is the highest one. However, all algorithms compared with the Shortest Path algorithm (SP) display rather poor performance. This result was expected, because any local algorithm can only approach the optimal solution given by the Shortest Path algorithm.

The number of packets in the network can be treated as a noise in the

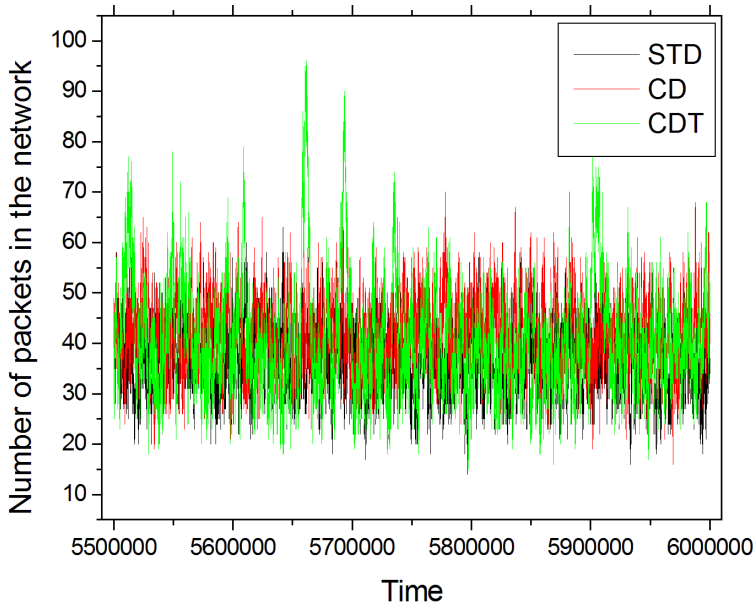


Figure 1.1: Load in the network for the STD, CD and CDT algorithms.

network. The measure of the power spectrum of this noise is an useful indicator of the type of correlations found in the network and is usually given by the power law relation

$$P(f) = af^{-\beta} \tag{1.5}$$

where f is the load frequency. The exponent β takes typically values from 1 to 2, where $\beta = 1$ indicates long-range correlations and $\beta = 2$ the short ones. In Fig. 1.2 we show the power spectrums for STD, CD and CDT algorithms. In all cases, we observe long-range correlations ($\beta < 2$) for frequencies smaller that 10^{-3} and short-range correlations ($\beta = 2$, blue dash line) above this value. In the inset of Fig. 1.2 we present parts of the power spectrums for long-range correlations with fitted curves. The exponents are

Algorithm	Mean load	SD
STD	36.58	6.17
CD	41.70	6.71
CDT	40.33	10.75
SP	0.22	0.47

Table 1.1: The mean load values and their standard deviations for the STD, CD, CDT and the Shortest Path (SP) algorithms. The values are obtained for the time series shown in Fig.1.1.

$\beta = 1.06 \pm 0.15$, $\beta = 1.12 \pm 0.12$ and $\beta = 1.47 \pm 0.18$ for STD, CD and CDT algorithms, respectively. The change in the exponent of the power spectrum for $f = 10^{-3}$ is a finite size effect of the network. We can find in Fig. 1.3 that the change in the exponent β is observed for $f = 1/N$, where N is the size of the network. For a small network $N = 100$ we find almost flat power spectrum (white noise) for $f < 10^{-2}$ which is related to a very small network load. This is due to a very large hub node in this network, which is very efficient in delivering packets, even for the STD algorithm that minimise the use of hub nodes.

We measured the distribution of the waiting time ΔT, the factor we introduced before to prevent long inactivity times of links. This is a local property, which describes waiting time of node i for a packet from node j or more intuitively the directed inactivity time of $i - j$ link. The reciprocal of ΔT is used only by STD and CDT algorithms but we measured it also for CD and SP algorithm for the comparison purposes. The results are shown in figure 1.4. For short waiting times ($\Delta T < 50$) the distributions are flat for all algorithms. Both STD and SP algorithms have clear power law tails of their distributions $p(\Delta T) = a\Delta T^{-b}$ with the exponent $b = 1.5$. However for STD, power law behaviour starts for $\Delta T > 10^2$ and for SP for $\Delta T > 10^4$. In the case of CDT and CD algorithms the distributions decay faster than STD

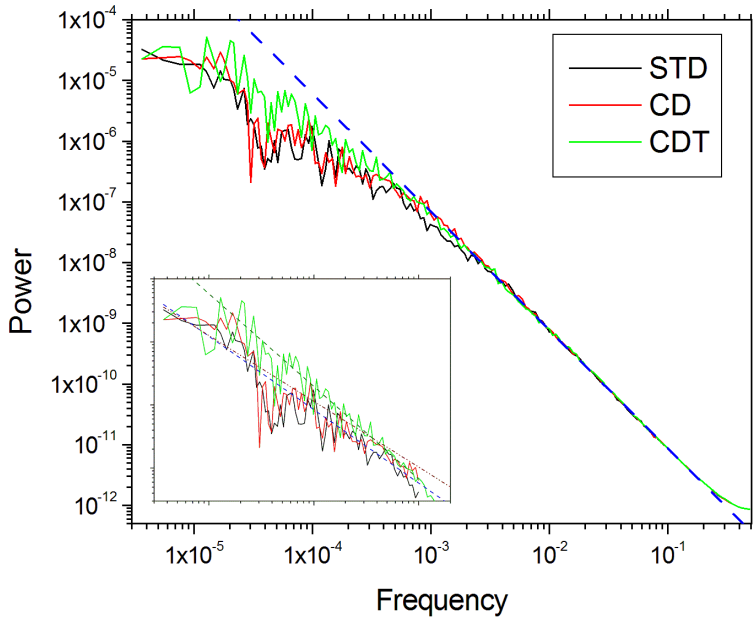


Figure 1.2: The power spectrum of the navigation algorithms for the STD, CD and CDT algorithms. The inset of the figure shows parts of the distributions for $f < 10^{-3}$ and the fitted power law curves $P(f) = af^{-\beta}$ with exponents $\beta = 1.06 \pm 0.15$, $\beta = 1.12 \pm 0.12$ and $\beta = 1.47 \pm 0.18$ for STD, CD and CDT algorithms, respectively.

and SP ones due to the $C(i)$ property. The fastest decay for CDT algorithm is related with incorporation of $C(i)$ and $1/\Delta T(i)$ properties which forces usage of larger number of network links and thus decreases links inactivity time. For the SP algorithm the waiting times are equally probable in a very broad range ($\Delta T < 10^4$).

The distribution of packet delivery time (Fig.1.5) is similar for all the algorithms. However the distribution shows that the number of packets delivered in a short time is different for each algorithm. The packets are delivered quickly most frequently for the STD algorithm due to the $\frac{T_P(i)}{N_P(i)}$ factor, which promotes the fastest delivery paths. For the CDT algorithm the impact of

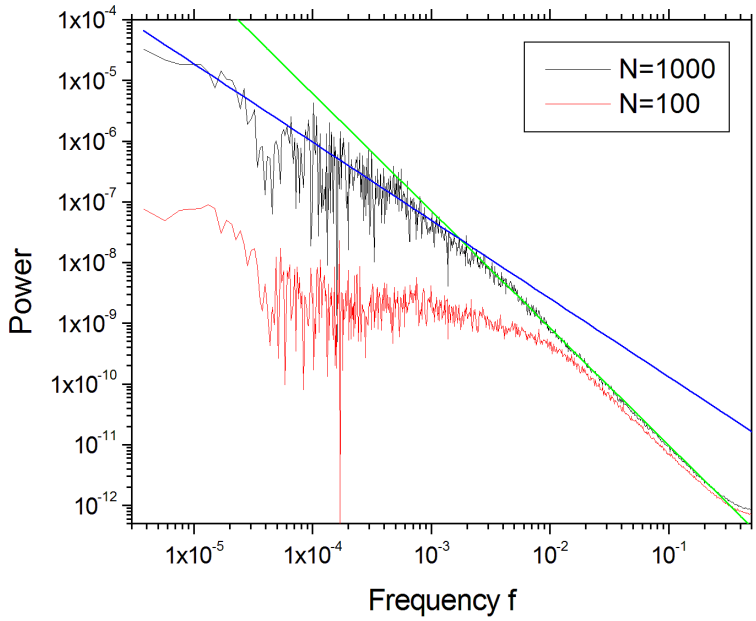


Figure 1.3: The power spectrum of the STD navigation algorithms compared for two network sizes $N = 1000$ and $N = 100$. The two power law curves with exponents $\beta = 1$ and $\beta = 2$ indicate the slopes of the power spectrum.

this factor is decreased by the $C(i)$ property, which tends to equalise the link loads. Thus, the result for the short delivery times is worse than for the STD algorithm. Finally, the CD algorithm displays the worst performance among our three studied algorithms in delivering packets quickly. If we do not consider the sharply decreasing tail, the packets' delivery time distribution for the CD algorithm is mostly flat. This is a reflection of the $C(i)$ property, which tends to distribute packets equally. As we might expect the distribution shifts to the right for a larger network size that is shown in the inset of Fig.1.5. This is due to the larger distances between the places of origin and destination, which make the very long distances more frequent to find.

The next measure we study is the mean delivery time series. This mea-

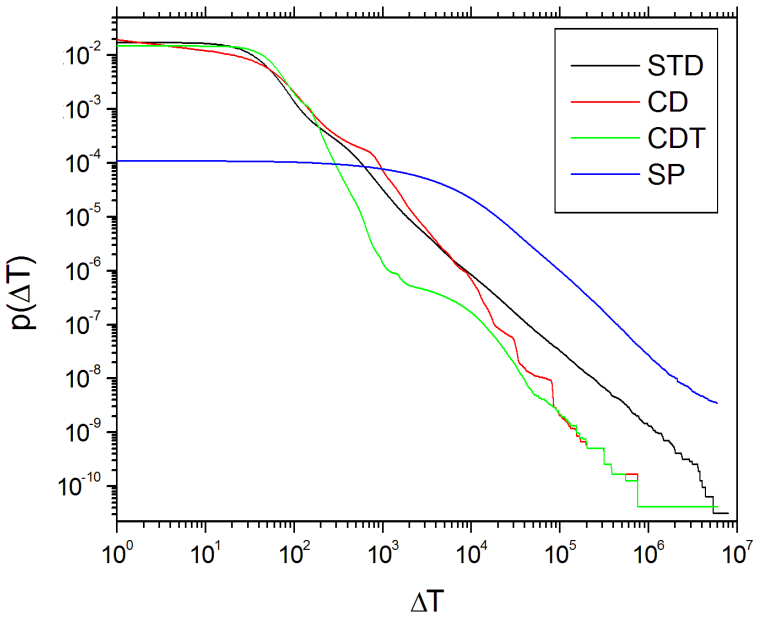


Figure 1.4: The distribution of the time interval ΔT calculated for the STD, CD, CDT and the Shortest Path algorithms.

sure indicates the overall performance of an algorithm and its stability. It is calculated as a sum of all packet delivery times since the beginning, divided by the number of delivered packets. The time series for the mean delivery times are calculated for 6 million time steps and are shown in Fig. 1.6. The CD and CDT algorithms reach stable level of the mean delivery time quickly, but in the case of STD algorithm, the transient time is very long. Thus, for this algorithm we calculated the standard deviation, which is shown in the inset of Fig. 1.6. It was obtained for the subsequent time windows, where each time window contains 80,000 time steps. The inset shows that the stable state for the STD algorithm is reached after 5 million time steps. We compare the performance of all studied algorithms in the Tab. 1.2. The STD, CD and CDT algorithms are outperformed by the Shortest Path algorithms

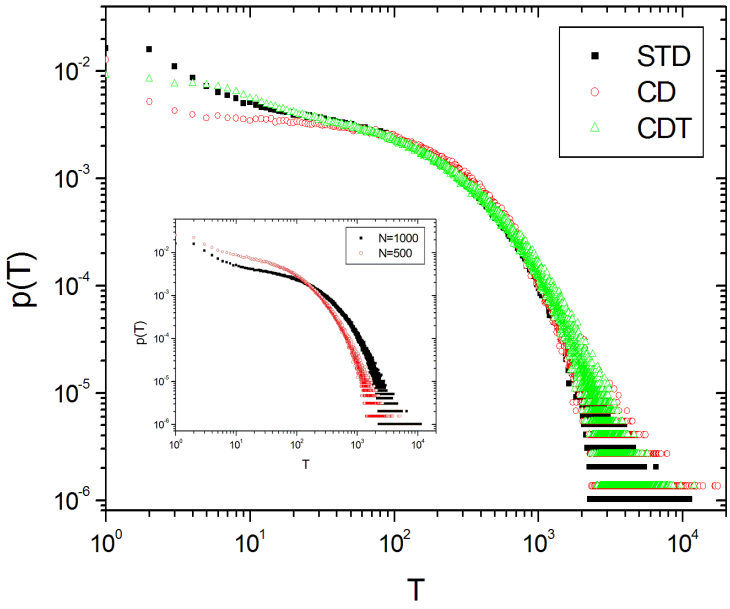


Figure 1.5: The distribution of packet delivery time for the STD, CD and CDT algorithms. The inset of the figure shows the network size effect in the case of the STD algorithm. The mean delivery time grows with the network size which results in the shift of the distribution to the right for a larger network.

significantly. However, this might be expected for any local navigation algorithm and low packets load. In our case it results mainly from the degree property, which tends to avoid the large nodes.

The overall mean delivery time is considerably lower for the STD algorithm than for the CD and CDT algorithms. However, we might expect that this algorithm will minimise its mean delivery time due to the $T_P(i)/N_P(i)$ factor. It does not happen mainly because of the inclusion of $\Delta T(i)$ property. It prevents jamming and supports learning in the initial part of the simulation but it also forces the usage of the infrequent links thereafter. We considered switching it off after a certain number of time steps, however we

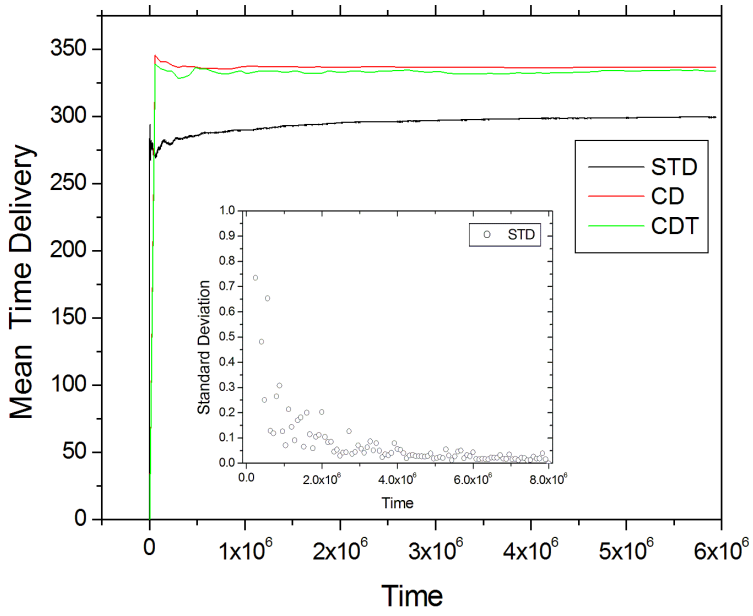


Figure 1.6: The mean delivery time series for the STD, CD and CDT algorithms. All algorithms reach stable state, however in the case of the STD algorithm the transient time is very long. The inset of the figure shows the standard deviation calculated for the STD algorithm mean delivery time series. The algorithm reaches the stable state after 5 million time steps.

encountered another problematic issue, which is an intrinsic weakness of all possible algorithms based on the average shortest time property. The problem lies in a direct link between two hub nodes. The large nodes have a significant number of the possible destination nodes. Thus, a link between them has a very low average delivery time, much lower than any other link directing to a small node. Without the $\Delta T(i)$ property two large nodes would send all packets between themselves and update the mean delivery time of the link only for the packets addressed to the nodes in their closest vicinity. All other packets would travel endless back and forth leading to the jamming

of the hubs.

Algorithm	Mean delivery time	SD
STD	295.91	5.69
CD	336.68	1.12
CDT	333.14	1.35
SP	1.83	0.01

Table 1.2: The mean delivery time values and their standard deviations for the STD, CD, CDT and the Shortest Path (SP) algorithms. The values are obtained from the mean delivery time series shown in Fig.1.6.

1.5 Conclusions

The aim of this chapter was to investigate the possibility of using the shortest delivery time property to navigate packets in a complex network. To do so, we created three local and deterministic navigation algorithms, the ST, STD and CDT, and we studied their properties such as the load and its power spectrum, the link inactivity time ΔT, packets' delivery time and the time series of the mean delivery time. We compared the performances of the algorithms between themselves and with the benchmark Shortest Path algorithm. Additionally we compared the results with the CD navigation algorithm, which is a deterministic version of the algorithm studied in [46] and [47].

We were mainly interested in studying the local navigation algorithms due to the fact that the transportation networks can be extremely large and the knowledge of the network structure may not necessairy be available for each network node. The implementation of the Shortest Path algorithm in the internet network is possible because of the powerful high speed routers and the high capacities of the network backbone structure.

We supposed originally node i, based on the shortest time property $T_P(j)/N_P(j)$, will send packets to its neighbours and update in time the statistics about the delivery time through each neighbour. As a result node i will find local quickest on average node j in its vicinity in delivering packets to the destinations. As we mentioned before, the most significant problem that emerges here is caused by the direct links between large nodes. The shortest time algorithm finds these links locally quickest and the hubs send packets back and forth through such links endlessly, leading to the jamming of the network. Our first solution to this problem was an introduction of $1/\Delta T$ property, which forces use of infrequent links. It occurred however, that this was not enough to prevent jamming of the network. Our next step based on avoiding large nodes strategy through implementation of the degree D property, realised in the STD algorithm. It was natural at this step to compare the results of the STD algorithm with the performance of the algorithm introduced in [46], the CD algorithm. This algorithm uses only information about the degrees of the neighbours of node i, the shortest time property is not implemented here and the $C(j)$ property that counts packets posted down from i to j is only to make it deterministic. The CDT algorithm is a straightforward combination of the CD and ST algorithms, very similar to STD algorithm. While for the CD algorithm the $C(j)$ property is essential to perform, its combination in the CDT algorithm with $1/\Delta T(j)$ property results in high volatility of links usage. Thus, for the CDT algorithm, the ΔT property studied in Fig. 1.4 decays very fast for the long link inactivity times.

The STD navigation algorithm displays the best performance among our studied algorithms. The mean delivery time is shortest for it and because of that the network load is lowest. On the other hand, the transition time when the STD algorithm reaches the stable state is very long. However,

in comparison with the simple local navigation algorithm [46, 47], the STD algorithm is definitely a step forward.

Finally, the necessary inclusion of $1/\Delta T$ and D properties has significant side effects. The learning ability of the shortest time property is considerably slowed down (long transient time), the mean delivery time does not minimise in time and its value for the stable state is high. Thus, our goals were not fully achieved. We believe the shortest time of delivery factor can be successfully applied, but only if the unwanted behaviour of sending back and forth along one link is solved.

Chapter 2

Origin of Scaling on Networks, Structural Inhomogeneity and Preference in Dynamical Behaviour

We examined the fluctuation properties of packet traffic on scale-free networks and random graphs using two different dynamical rules for moving packets: random diffusion and a locally navigated diffusive motion with preferred edges. We found that preferential behaviour in either the topology or in the dynamics lead to the scaling of fluctuations of the number of packets passing nodes and the number of packets flowing along edges, respectively. We show that the absence of any preference results in the absence of scaling, and when scaling occurs

it is non-universal with the scaling exponents depending on the acquisition time window, the network structure and the diffusion rule.

2.1 Introduction

Most real systems exhibit complex dynamical behaviour. An interesting property of complex systems is that the scaling of fluctuations is found in rivers, stock markets, computer networks, World Wide Web (WWW) and electric circuits [79, 83]. This occurs when the average activity $< X_i >$ of the component i of a system is related to the standard deviation (called dispersion in related works) σ_i of its time series by power law behaviour $\sigma_i \sim < X_i >^{\mu}$, where the value of the exponent μ is between 0.5 and 1.0. Initial studies [79] found the exponent μ takes only two values: 0.5 or 1.0, which were related to the internal and the external dynamics, respectively. However, simulations on networks found the value of μ to be dependent on the traffic parameters such as the input rate R, the packet's life time S, or the time window T_{WIN} of data acquisition [81, 86]. Similar dependencies of the acquisition time window are observed in the analysis of the empirical time-series of stock markets [86, 84] and in the gene expression data, where the natural time window is determined by the cell-cycle dynamics [35]. The occurrence of scaling and reasons for its nonuniversality have been the subject of debate [79, 83, 81, 86, 84], with conclusions sometimes obscured by the nature of the empirical data or limitations in the models.

In Chapter 1 four navigation rules for delivering packets in a complex network have been introduced. All these algorithms were the edge-preferred navigation rules, where the preference was based on the properties of a node at the end of the edge, such as the degree and mean delivery time. In this chapter we are mainly going to use a probabilistic version of the CD algo-

rithm, called D algorithm, however without any in-depth search. The reason is that this algorithm is the simplest possible realisation of algorithms introduced in Chapter 1 to observe the scaling of flow fluctuations on links. It does not involve any additional properties such as the shortest path property (see Chapter 1) that might make the outcome results unclear. However, in some particular cases we show also the results for the most efficient algorithm we introduced in previous Chapter, the STD algorithm, for comparison purposes.

By simulating the traffic of packets on an uncorrelated scale-free network with the edge-preferred navigation rules, we show that preferences in either topology (i.e., for nodes on a scale-free network), or in the dynamics (i.e., for dynamically preferred edges), is necessary for the occurrence of the scaling. Furthermore, we have shown that the nonuniversal dependence of the exponent μ on the time window appears to be different for nodes and for edges, and related to waiting time distributions.

In this work we investigate the role of network topology and the role of navigation rules in the occurrence of scaling. For this purpose we studied in parallel:

- traffic on a scale-free network and on a random graph;
- random diffusion and a probabilistic edge-preferred local navigation rule on both networks;
- fluctuating time series recorded at nodes, $\{h_i(t)\}$, and at edges, $\{f_{ij}(t)\}$, of the network within a specified time window T_{WIN}.

Therefore we defined two types of the dispersion relations, for the activity of nodes and for flow along the edges:

$$\sigma_i \sim < h_i >^{\mu} \; ; \quad \sigma_{ij} \sim < f_{ij} >^{\mu} \; . \tag{2.1}$$

Our findings in this larger class of network structures and dynamic rules confirmed the necessity of a preference for the scaling of the fluctuations in Eq.(2.1). Furthermore, we demonstrated that the variations of the values of the scaling exponents with the time window, both for nodes and links, are strictly related to the network topology and to the navigation rules.

In Section 2 we introduced the network structures and navigation rules and we briefly described the basic traffic properties. Section 3 focused on the fluctuations of the time series recorded at nodes and at edges from the simulations with the random diffusion on both network types and similarly, in Section 4, the results for the edge-preferred navigation on both network types. The conclusions of our results are given in Section 5.

2.2 Network structures and transport rules

Networks. To investigate the role of network structure on the fluctuating time series we studied a network with a scale-free connectivity distribution and a random graph. In both cases the network consisted of $N = 1000$ nodes and $E = 2N$ edges (links), with average connectivity per node $< k >= 2E/N = 4$. We grew the *scale-free network* with the preferential attachment described in detail in Ref.[87]. In this model for each new node, m new links are added to the network. However, with the probability α a link is created between the new node and one of the older nodes and with probability $1-\alpha$ a new link is created within the network, only between the older nodes. The network with the connectivity $k_i \sim (i/N)^{-1/(1+\alpha)}$ at ith added node emerges, leading to the degree distribution $P(k) \sim k^{-(2+\alpha)}$. In our case $m = 2$, $\alpha = 0.5$

and at least one link connects a new coming node with the network, assuring that there are no disconnected nodes and all destinations are reachable. The *random graph* of the same size and number of edges is made by starting from N nodes from each of which $m = 2$ links are randomly connected to two other nodes. Multiple links are not allowed and we also take care to produce a graph with all nodes connected to the giant cluster. It should be stressed that both networks are uncorrelated and have low clustering coefficient. Once the networks are generated, we consider their structure fixed, and given by the adjacency matrix C_{ij}.

Navigation rules. We simulate the transport of packets on these networks within the traffic model [62, 42]. The packets are created with a rate R and each packet is given a destination address where it is eventually delivered and removed from the network. The packets are moved through the network in parallel using a local navigation rule (Chapter 1, [62]). Packets queue at nodes with the FIFO preference rule. Here we use two strictly local rules to navigate packets towards their specified destinations. These rules are defined by the probability p_{ij}, which a node i forwards a packet towards one of its neighbour nodes j:

$$p_{ij}^{D} = \frac{2}{k_i} - \frac{k_j}{\sum_{j=1}^{N} C_{ij} k_j} \; ; \qquad p_{ij}^{RD} = \frac{1}{\sum_{j=1}^{N} C_{ij}} \; ; \tag{2.2}$$

where C_{ij} is the adjacency matrix and k_j is the degree of node j and subscripts RD and D denote *random diffusion* and *degree-dependent* D-navigation rule, respectively. Note in contrast to the algorithms described in Chapter 1, the D rule in Eq.(2.2) does not imply any in-depth search, i.e., node i does not know addresses of its neighbours and is unable to send directly any package addressed to any of them, similar to the random diffusion. However,

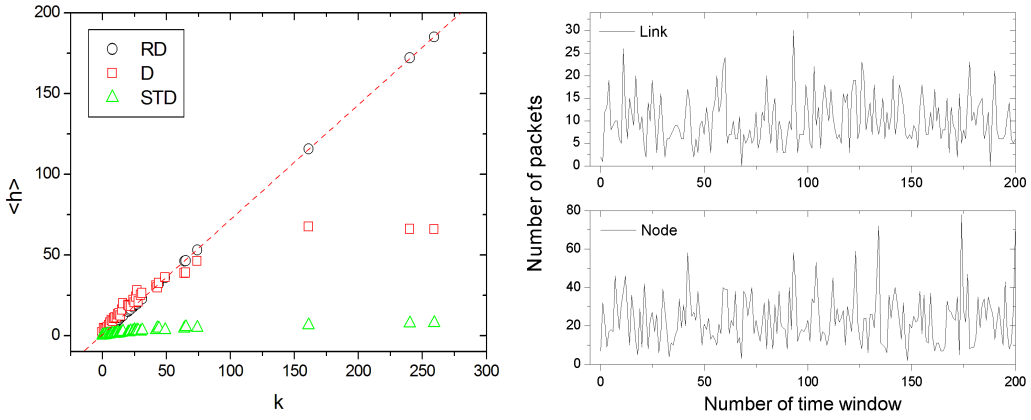
according to the D rule, the *edge linked with the less-connected neighbour node is dynamically preferred.* In effect, the central node loses its topological preference. This is in contrast to the random diffusion, where the number of visits of a random walker to a node is proportional to the node's connectivity [90], i.e., the average number of packets processed by a node i, $< h_i >$ is given by node's degree k_i

$$< h_i > \sim k_i. \tag{2.3}$$

Consequently, the average number of packets processed by a node, as shown in Fig.2.1a, is different in the two dynamic rules. With the edge-preferred local rule D, described above, we observe the dynamic homogeneity of the network, i.e., $< h_i > \sim const$ for large k_i.

The $< h_i >$ (k_i) dependence found for the D navigation rule is very similar for the STD algorithm. This shows that degree property has the largest impact on the performance of this algorithm. The much lower level of $< h_i >$ for the STD algorithm than the D one for the same values of k_i results from much better performance of the STD navigation rule, mainly due to the 1-depth search (see Chapter 1).

In the following we study in detail the fluctuating time series which represent the time fluctuations of the number of packets processed by nodes, $\{h_i(t)\}$, for all $i = 1, 2 \cdots N$ nodes in the network and time fluctuations of the number of packets processed along a link (packet flow), $\{f_{ij}(t)\}$, for all E links (connected pairs ij) in the network. Examples of such time series for the scale-free network and two diffusion rules are shown in Fig. 2.1b.



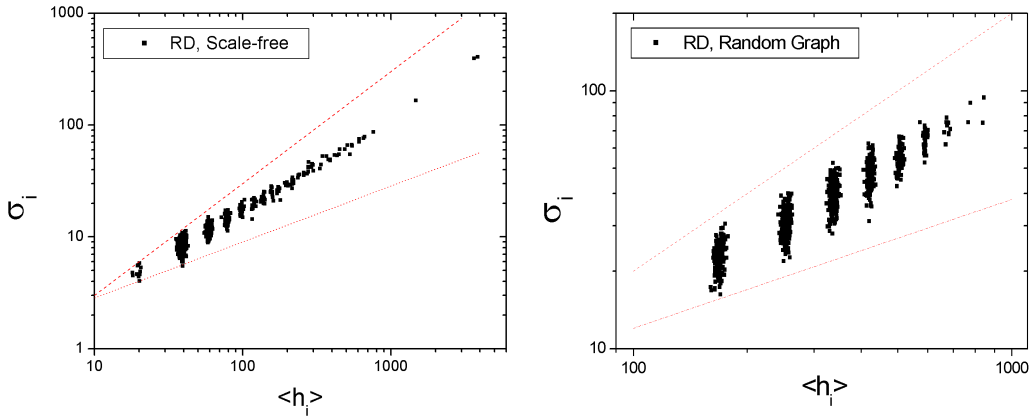
(a) (b)

Figure 2.1: (a) The average number of packets processed by a node against node degree for random diffusion (RD), the D and the STD navigation algorithms (STD introduced in Chapter 1). The plot for the random diffusion is fitted with a linear function. (b) Example of time series recorded at a preferred node with random diffusion rule (bottom) and time series recorded at a preferred edge with the D navigation algorithm for time-window $T_{WIN} = 1000$ steps.

2.3 Scaling of Fluctuations for random diffusion on networks

In this section we investigate the scaling of noise fluctuations $\{h_i(t)\}$ for the random diffusion process on two types of underlying structures, the scale-free network and the random graph, described in the previous section. Fig. 2.2 shows the relation between the dispersion σ_i and the average noise $< h_i >$, where each node i is represented as a point. The plots for the scale-free network and the random graph follow the general scaling relation in Eq.(2.1). In both cases the scaling exponents μ are between the border values 0.5 and 1.0 (indicated by the thin lines).

According to Eq.(2.3) the average noise $< h_i >$ is related to node's degree



(a) (b)

Figure 2.2: Random diffusion: Dispersion σ_i against average $< h_i >$ of the time series recorded at nodes of the network within a fixed time window $T_{WIN} = 4000$ on (a) a scale-free network and (b) a random graph. The exact value of the scaling exponent α was obtained using software Origin, which applies Levenberg-Marquardt algorithm [100].

k_i. This property results in clearly separated groups of points, where each group contains only nodes with the same degree. The value of the scaling exponent μ may depend on the traffic conditions[81] such as input rate R or the closeness to jamming [82], and on the acquisition time window T_{WIN} [81, 83, 84]. In our simulations we use input rates R much below the jamming limit R_c[62]. We further investigate the dependence on the time window $\mu(T_{WIN})$ for both our network structures. The results are presented in Fig. 2.3.

In both cases the dependence between scaling exponent μ and the length of the observation time window T_{WIN} is monotonical and μ grows with T_{WIN}. One of the explanations given in [79] states that the fluctuations of node i activity have two sources, external and internal ones. When the external fluctuations are absent the observed scaling exponent μ is is always 0.5. But once the external fluctuations are present we observe continuous change of

the scaling exponent μ from 0.5 to 1.

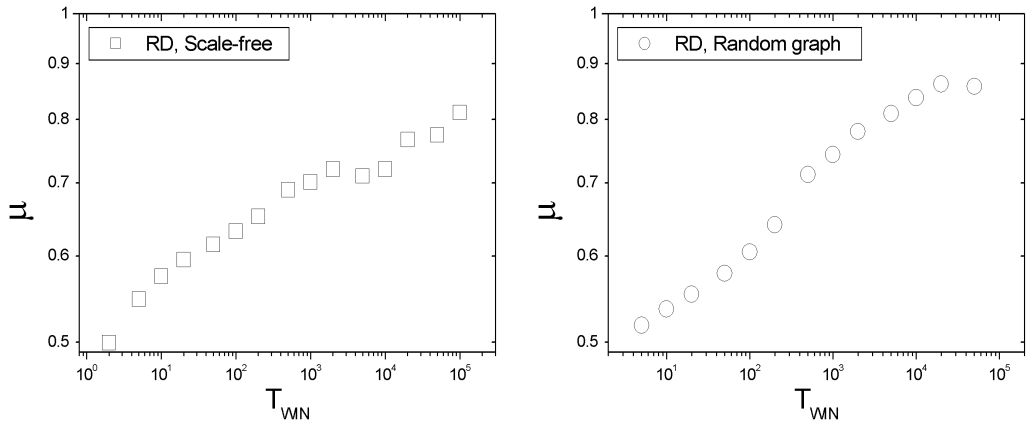
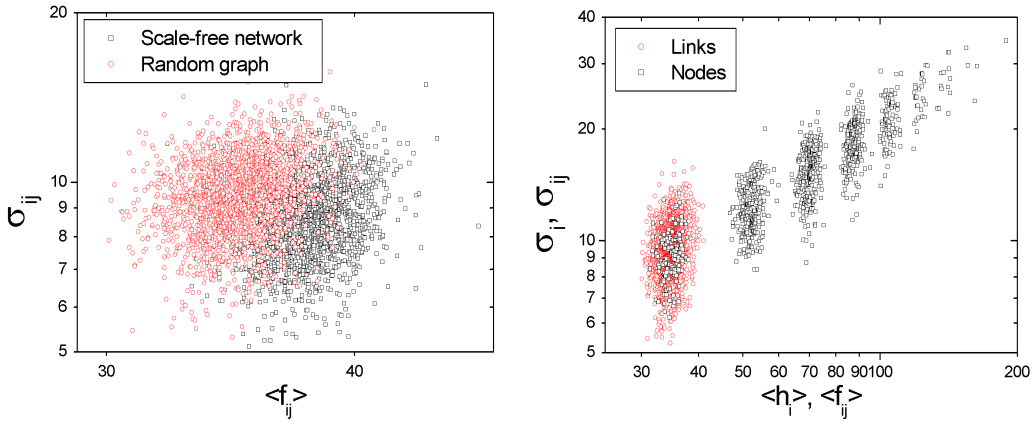


Figure 2.3: Random diffusion: Dependence of the scaling exponent μ for the node activity fluctuations $\sigma_i \sim < h_i >^{\mu}$ on the width of the time window T_{WIN} on a scale-free network (a) and on a random graph (b).

The same technique can be applied to analyse the fluctuations of flow along the links. In this case we measure the *flow*, f_{ij}, which is defined as the number of packets posted from $i \to j$ and from $j \to i$ within a given time window T_{WIN}. The relation between the flow dispersion σ_{ij} and average flow $< f_{ij} >$ is plotted in Fig. 2.4a for the scale-free network and the random graph. In these plots each point represents one edge of the network. In this case, however, no scaling was found for either network structure. This result can be easily understood in view of the Eq.(2.3) if we consider a link as an element with two inputs/outputs, similar to a node with degree $k = 2$. Indeed, all network links form a single group (cf. Fig. 2.4a,b). Moreover, the group of all links overlaps with the group of nodes with the degree 2,

$$< f_{ij} > \approx < h_i >_{k_i=2}, \tag{2.4}$$



(a) (b)

Figure 2.4: Random diffusion: (a) Dispersion σ_{ij} against average flow $< f_{ij} >$ of the time series recorded at links of the network within a fixed time window $T_{WIN} = 4000$ on a scale-free network and a random graph. (b) The comparison between the fluctuations of the node activity and flow along the links on the random graph for the random diffusion process. Several groups of nodes are distinguishable, according to their connectivity, whereas all links fall into a single group.

as shown in Fig. 2.4b in the case of a random graph. In this figure, groups of nodes with increasing connectivity are systematically shifted to the right, in agreement with Eq.(2.3).

The close relationship between the node's connectivity and the differentiation between node groups in the plots, as in Fig. 2.3 and 2.4b, is characteristic for the random diffusion processes. In short, the role that a node plays in the random diffusion is entirely determined by the number of links attached to it. This can be seen even in a simple structure like the regular square lattice with open boundaries shown in Fig. 2.5, where groups of boundary nodes are differentiated from the interior nodes.

We can now conclude that for random diffusion, the scaling of fluctuations in Eq.(2.1) occurs only in systems where the diversification of a property (like

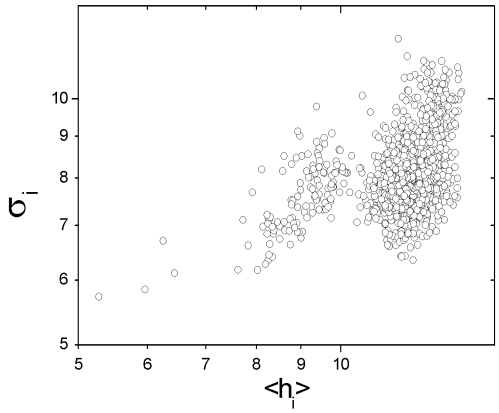


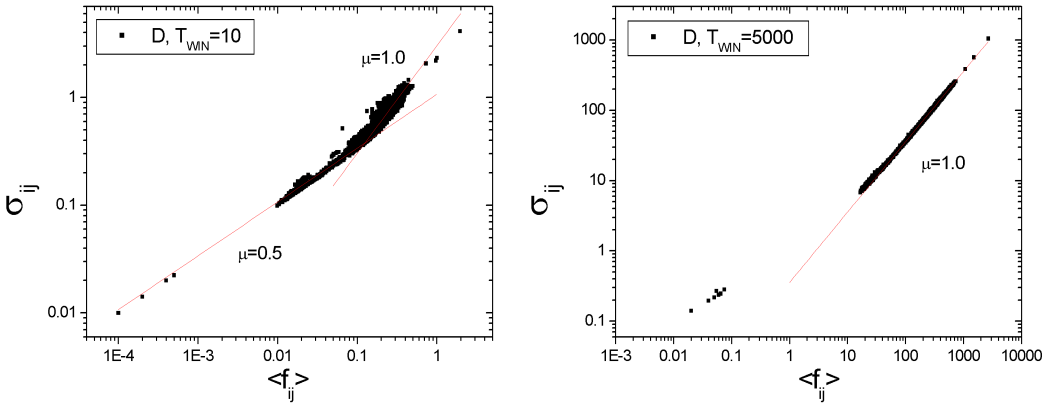
Figure 2.5: Random diffusion on a regular square lattice with 32×32 nodes: Dispersion σ_i against average node activity $< h_i >$. Four corner nodes (left), the groups of boundary nodes (middle) and the interior nodes (right) are distinguishable.

node degree) is present. The degree distribution for scale-free network and even the random graph provides enough diversification in degree values to observe the scaling in node activity fluctuations. In the case of flow fluctuations, however, we do not find any scaling property because in the random diffusion every link transfers almost the same number of packets. In the next section we will show that, although the links on both scale-free and random graph are topologically equally important, the dynamical preference within the navigated diffusion rule (D) will induce the flow differentiation that is necessary for the scaling to appear.

2.4 Scaling of Fluctuations for edge-preferred navigation

We adopt the local navigation rule with the edge-preferred diffusion, defined by in the Eq.(2.2) with the probability p_{ij}^D. As discussed in Section 2, with

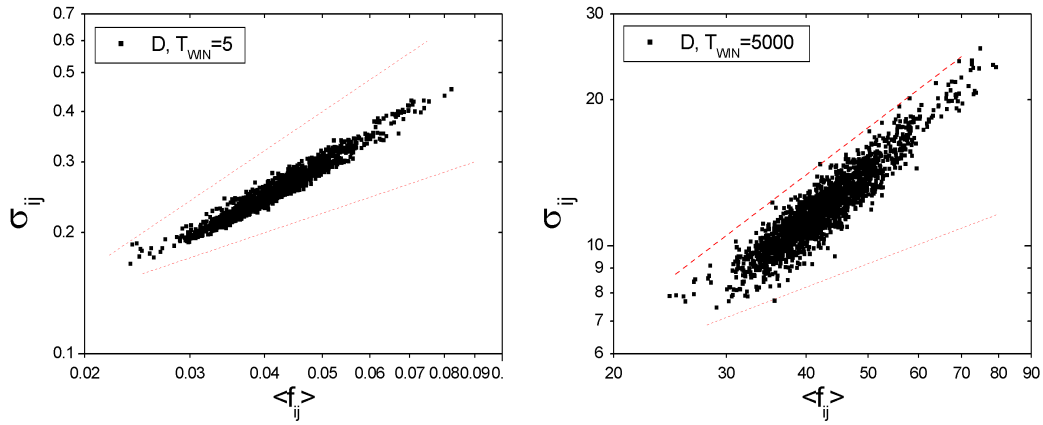
this rule, the packets are preferably posted along the edges pointing towards the neighbour node with smaller degree. The rule is more effective at nodes close to the hub in the scale-free network. Precisely, the packets are avoiding the large-degree nodes, whereas the nodes at the graph boundary and in the random graph, the majority of nodes have similar degree and thus small differences between edges connecting them could only weakly affect the dynamics. Note that we study here the diffusion rule D to demonstrate the origin of scaling at network edges. We do not discuss here the potential effects of the rule on the traffic efficiency (see refs. [72, 42, 71, 62] and Chapter 1).



(a) (b)

Figure 2.6: Edge-preferred D-navigation on scale-free network: Dispersion σ_{ij} of flow along the edges against average flow $< f_{ij} >$ for the time window $T_{WIN} = 10$ (a) and $T_{WIN} = 5000$ (b).

Fig.2.6 and Fig.2.7 show the relation between dispersion σ_{ij} and the averaged flow $< f_{ij} >$ for the scale-free network and the random graph, respectively. In the case of the scale-free network, as a new feature of the navigated diffusion we find the scaling in the flow fluctuations, seen in Fig.



(a) (b)

Figure 2.7: Edge-preferred D-navigation on random graph: Dispersion σ_{ij} of flow along edges against average flow $< f_{ij} >$ for time window $T_{WIN} = 5$ (a) and $T_{WIN} = 5000$ (b). Thin lines indicate slopes $\mu = 1$ and $\mu = 1/2$.

2.6. In particular, we find two scaling regimes with exponent μ either 0.5 or 1.0. Further study has shown that the occurrence of two scaling regimes is rather robust to the variations in the time window T_{WIN}. However, we find that the population of points, representing different links of the graph, migrate from the upper part of the scatter plot with slope $\mu = 1$ to the lower part, where the slope $\mu = 1/2$ was found, when the time window is *decreased*. For the limiting case $T_{WIN} = 1$ almost all points fit to the curve with exponent μ equal to 0.5. Hence, a continuous variation of $\mu(T_{WIN})$ is absent, in contrast to the scaling of the node activity.

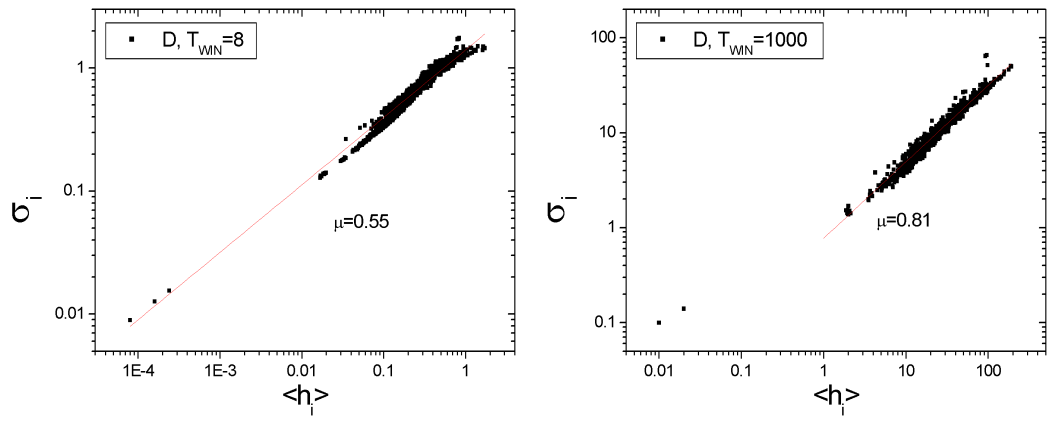
On the contrary, the two scaling regimes for $\sigma_{ij}(< f_{ij} >)$ dependency are not present in the random graph (Fig. 2.7), suggesting that this feature is caused by the structural properties of the scale free network. Indeed, we found that the $\mu = 0.5$ regime is occupied by the links connected to large nodes, in particular the links between hubs. The D navigation algorithm

strongly reduces the usage of such links, so that they are visited very rarely even for very large time windows T_{WIN}. Because of that, the transport of packets through such links is almost uncorrelated with the dynamics in the other parts of the network and resembles rather a random deposition process, for which $\sigma \sim < f >^{0.5}$ [79]. This explains the persistence of $\mu = 0.5$ regime for large T_{WIN}.

In the case of the second regime, only $\mu = 1.0$ was found. However, we show for the random graph that μ for the fluctuations of flow on edges actually scales, but only for very small time windows T_{WIN} (Fig. 2.10b). We were unable to find this behaviour for small T_{WIN} on the scale free network because while we decrease T_{WIN} the points migrate to the $\mu = 0.5$ regime. Hence, the span of the $\mu > 0.5$ part is so small that the real value of μ exponent is highly uncertain.

For the edge-preferred navigation the plots of node activity on the scale-free graph are also shown in Fig. 2.8. We find the qualitatively similar behaviour as in the case of random diffusion, namely, the nonuniversal scaling with the exponent continuously varying with the time window. However, the numerical values of the exponents are different for the two diffusion rules. The dependence $\mu(T_{WIN})$ is plotted in Fig. 2.10a.

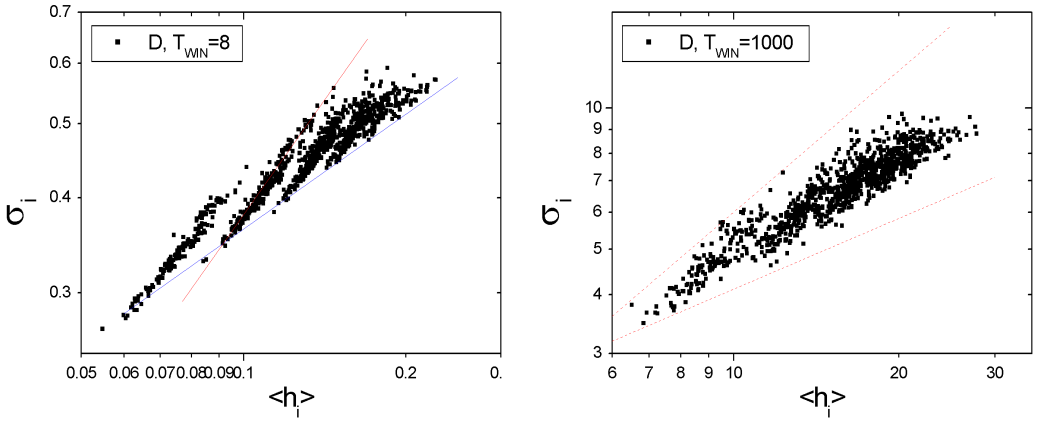
For the edge-preferred navigation on the random graph the scatter plots are shown in Figs.2.7 and 2.9. For the fluctuations of flow on edges we find a qualitatively similar behaviour as for the fluctuations of node activity. In particular, the scaling of Eq.(2.1) occurs with a continuously varying exponent when the time window is changed. The numerical values, however, are different. For the random graph we found a single scaling regime for the fluctuations of flow on edges that is related to the graph structure. There



(a) (b)

Figure 2.8: Edge-preferred D-navigation on scale-free network: Dispersion σ_i of node activity against average activity $< h_i >$ for time window $T_{WIN} = 8$ (a) and $T_{WIN} = 1000$ (b).

are no hubs which might significantly reduce flow on links pointing to them and the flow on links is highly homogenised compared to the flow on the scale free network. As shown in Fig.2.10b for the flow fluctuations, the exponent drops below the value $\mu = 1$ only for very small time windows. In contrast, in the case of node activity, see Fig. 2.10a, lower curve, the characteristic crossover region between $\mu \sim 0.5$ and $\mu \sim 0.8$ was found. Note that for the node activity fluctuations the range of T_{WIN} where exponent μ scales is much larger, but the maximum value of μ found is still lower than 1.0 and in this particular case it is around 0.8. We ran our simulations on different random graphs (constant N=1000) and μ_{max} depends on a graph ensemble used in the simulation, however, for all cases μ_{max} was always smaller than 1.0. Note also that different groups of nodes are also distinguishable in the navigated diffusion on the random graph, Fig. 2.9, whereas such groups cannot be identified in the case of edges, Fig. 2.7. In contrast to the random diffusion, the

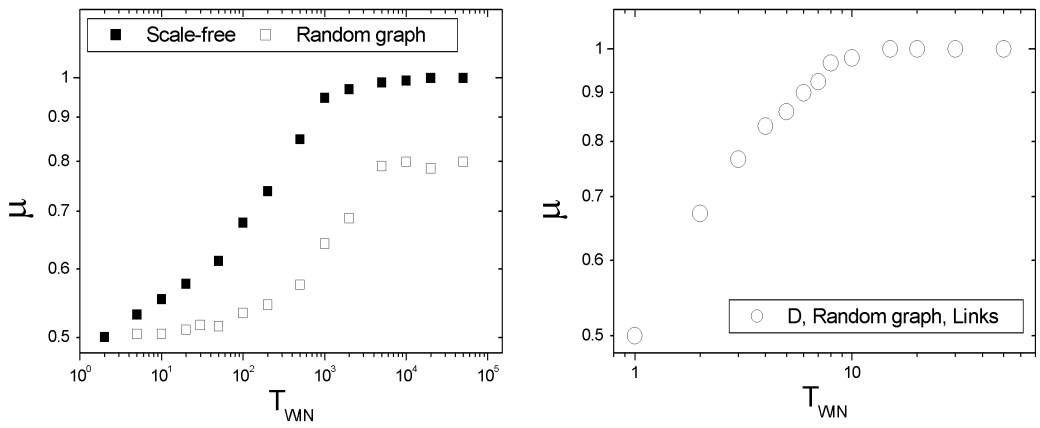


(a) (b)

Figure 2.9: Edge-preferred D-navigation on random graph: Dispersion σ_i of node activity against average activity $< h_i >$ for time window $T_{WIN} = 8$ (a) and $T_{WIN} = 1000$ (b). Thin lines indicate slopes $\mu = 1$ and $\mu = 1/2$.

edge-preferred navigation induces the dynamical difference between edges, that results in the occurrence of scaling in the plots both for scale-free and for random graph, Figs. 2.6 and 2.7. The study of the node activity fluctuations for the D navigation algorithm shown in Fig.2.8 and Fig.2.9 reveals different dependencies of the scaling exponent on the time window, compared to the random diffusion on the same graphs, but also it shows the differences between the traffic on a random graph and on a scale-free network.

The results obtained for the STD algorithm (Figs. 2.11 and 2.12) are similar to those found for the D navigation rule. The two part character of the $\sigma_{ij}(< f_{ij} >)$ dependency is also found here, however the transition from $\mu = 0.5$ to $\mu = 1.0$ region is more complicated (Fig. 2.11a) and disturbed by the additional navigation rules implemented into the STD algorithm than the simple D rule. Due to the same reason, the separation of the groups of nodes with the same degree found for the node fluctuations on the random



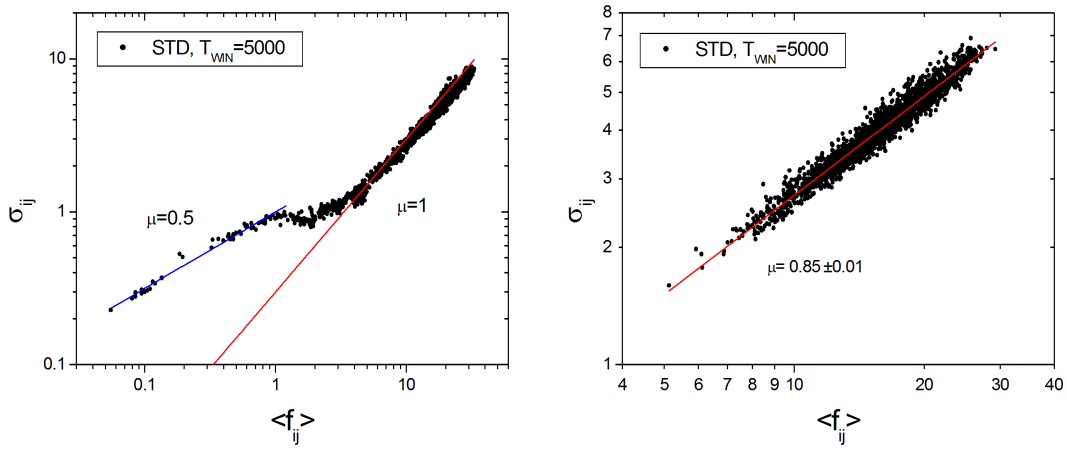
(a) (b)

Figure 2.10: Edge-preferred diffusion: Dependence of the scaling exponent μ on the width of the time window T_{WIN} for: (a) fluctuations of the node activity for the scale-free network (filled squares) and the random graph (empty squares), and (b) fluctuation of flow on edges on the random graph.

graph (Fig. 2.9) is not visible in the STD algorithm case (Fig. 2.12b).

2.5 Waiting times of Nodes and Edges

Another type of dynamic measure collected at individual nodes and edges that depends on the dynamic behaviour of the whole network is the statistics of waiting times ΔT, defined as time intervals between the successive events at a given node or an edge. In collective dynamical systems such as earthquakes [91, 92, 93], critical sandpiles [94, 95], and stock market dynamics [96, 97], a broad distribution of waiting times (sometimes called return times or recurrent times) is always found, with power-law tails suggesting the occurrence of long-range dynamic correlations between the events. In this section we address the question of waiting times to nodes and to edges in

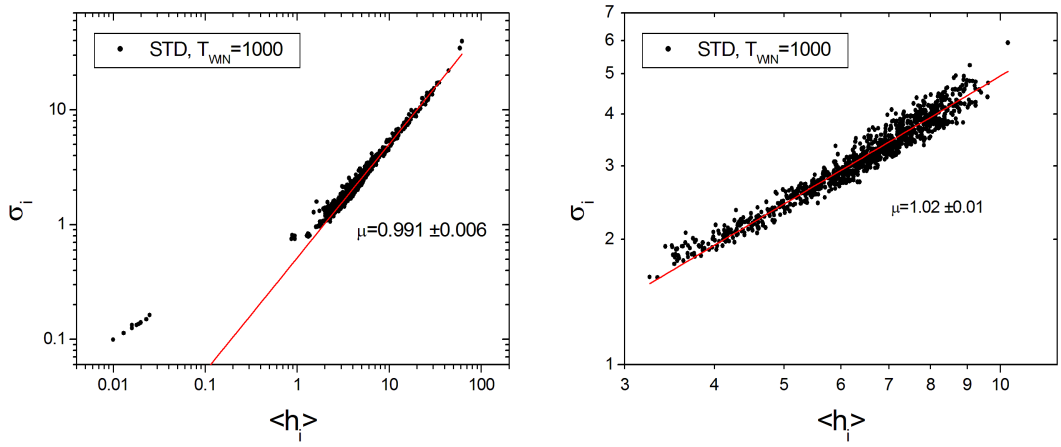


(a) (b)

Figure 2.11: Edge-preferred STD-navigation: Dispersion σ_{ij} of flow along the edges against average flow $< f_{ij} >$ for the time window $T_{WIN} = 5000$ on scale-free network (a) and random graph (b).

order to investigate further the nature of collective dynamic behaviour in our model of diffusion of packets on a scale-free network for the random diffusion and the introduced earlier D navigation rule. Note that the waiting time we study is the time taken for a node (or edge) to receive its next packet, and not the time taken for the same packet to return to that node (or edge).

Waiting times of Nodes. In the case of random diffusion (random walks) on networks the waiting time distribution has been studied in other parts of the theoretical physics literature. In particular, the waiting time of the first return to the origin of a random walker on sparse random graphs, with nodes representing states of a system, was considered by Bray and Rodgers [98] as a model of non-exponential relaxation in spin glasses and other non-ergodic systems. With the help of some heuristic arguments, they arrived at the conclusion that on a random graph the long-time behaviour in the diffusion in the phase space is dominated by the parts of the network with linear chains



(a) (b)

Figure 2.12: Edge-preferred STD-navigation: Dispersion σ_i of node activity against average activity $< h_i >$ for time window $T_{WIN} = 1000$ on scale-free network (a) and random graph (b).

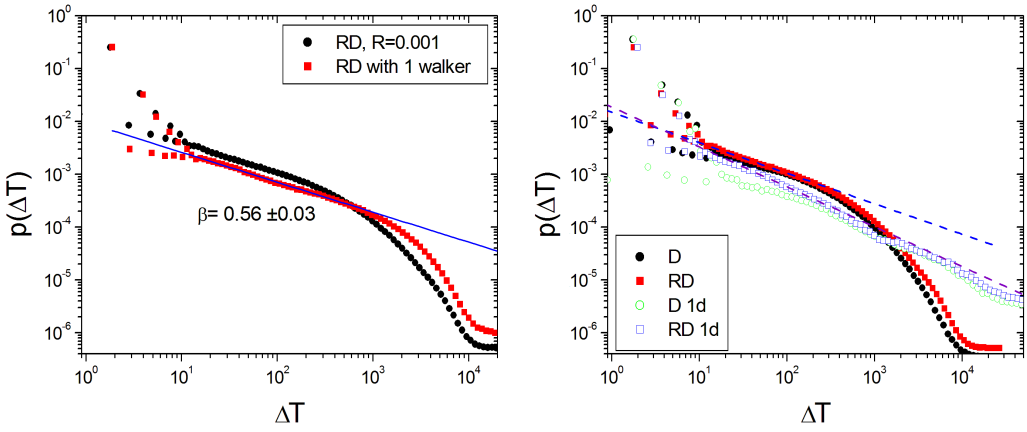
(no loops), leading to the expression $P(\Delta T) \sim \exp(-A(k)(\Delta T)^{1/3})$, where k is the average connectivity of the random graph and $A(k)$ is known.

These arguments can be generalised to introduce a power-law distribution of connectivities. If the distribution of k behaves like $\sim k^{-\tau}$, and using the result in [98] (under the assumption that the system is ramified) that for small k, $P(\Delta T) \sim k \exp(-2\Delta T/k)$, then integrating over k leads to $P(\Delta T) \sim (\Delta T)^{-\tau_\Delta}$ with

$$\tau_\Delta = \tau - 2 \; . \tag{2.5}$$

Thus, the inhomogeneous connectivity creates a power-law distribution in the waiting times distribution for RD. Recently a more rigorous treatment of random walks on scale-free networks was carried out by Noh and Rieger [90], that yielded identical results. The results of our simulations for different diffusion processes are shown in Fig. 2.13.

The waiting time distributions in different cases studied here seem to have



(a) (b)

Figure 2.13: The nodes' waiting time distribution on the scale-free network for non-interacting and interacting walks with (a) the random diffusion algorithm and (b) for interacting walks with different navigation algorithms. For the random diffusion process (the top line) the slope $\tau_\Delta \approx 0.5$ corresponds to theoretical prediction (Eq. 2.5).The character of the distribution of the D navigation rule is very similar to the random diffusion process and the slopes of two other diffusion processes are altered by the 1-depth search rule.

a power-law behaviour before a cut-off. (The cut-off can be related to the network size in the case of single random walker.) Note also a characteristic splitting at small ΔT with an inherent preference for even waiting times, caused by the lack of clustering and the low density of walkers. For instance, in a chain structure with a single walker only even nodes' waiting times are possible (Fig. 2.14). The odd waiting times for nodes and even for links can be found only in a structure with loops or a diffusion process with more than one walker. Thus, the higher clustering or traffic, the smaller split in the waiting time distributions.

In the case of non-interacting random walks, i.e., random diffusion without queuing, the results agree, within error bars, with the above theoretical

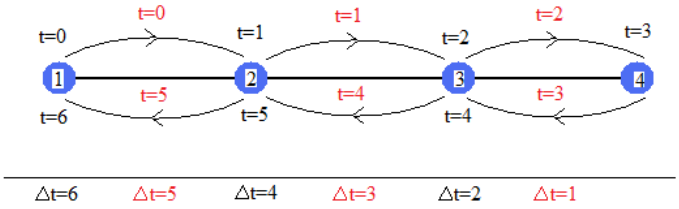


Figure 2.14: The nodes' and edges' waiting times in a chain structure with a single walker. Only even waiting times Δt for nodes and odd for links are possible here. The walker travels from node 1 to 4 and back starting at time $t = 0$. Black numbers are related with nodes and red with links.

prediction. We have the exponent $\tau_\Delta = 0.56 \pm 0.03$, whereas the distribution of the network's connectivity has a power-law exponent $\tau \approx 2.5$ (See sec. 2.2).

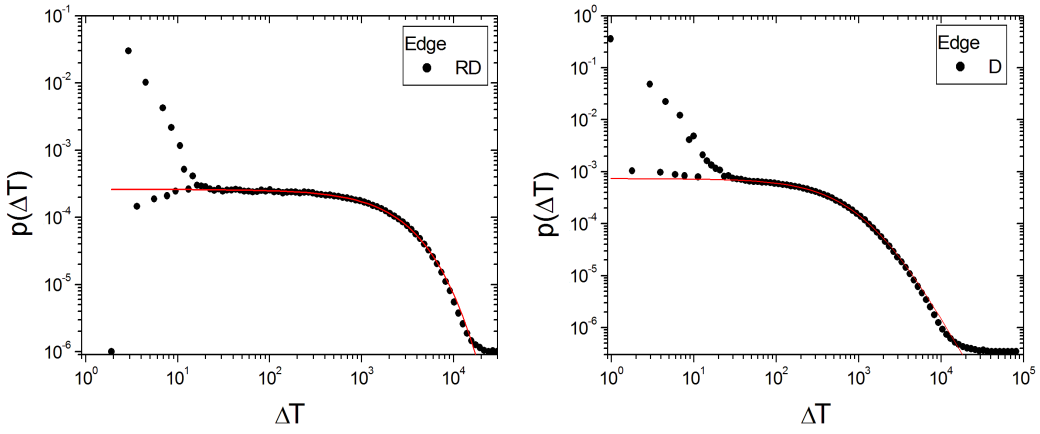
Increasing the traffic density reduces the value of the cut-off, but the slope remains practically unchanged. However, when the navigated diffusion is considered, both the slope and the cut-off of the distribution are changed. In Fig. 2.13b we show the results for the random diffusion and the D navigation algorithm, both with 0 and 1 depth search at low packet density ($R = 0.001$).

Waiting time of Edges. The situation is entirely different from the point of view of edges on the same network. The results are shown in Fig. 2.15. We find a pronounced difference between the random and navigated diffusion in the tails of the distributions. In both cases, however, a unique functional form can be found. For larger ΔT the distributions of the edges waiting times can be fitted with a $q-$exponential form, which is often related to non-ergodic behaviour in dynamical systems [99] :

$$P(\Delta T) = B\left(1 - (1-q)\frac{\Delta T}{\Delta T_0}\right)^{1/(1-q)} . \tag{2.6}$$

In the case of random diffusion, shown in Fig.2.15a, the distribution is very

close to the exponential form, which corresponds to the $q \to 1$ limit of Eq. (2.6). In fact, we find $q = 1.09 \pm 0.05$ in the case of random diffusion, whereas in the case of edge-preferred D navigation $q = 1.34 \pm 0.02$.



(a) (b)

Figure 2.15: Edges waiting time distribution for (a) the random diffusion and (b) D-navigated diffusion algorithms. The fit lines according to Eq. (2.6), as explained in the text.

2.6 Conclusion

Using the model of traffic on networks with packet creation and delivery, local navigation and queuing at nodes, we analysed the fluctuations of time series of node activities and traffic flow along the links. Two types of networks—uncorrelated scale-free network and a random graph and two local diffusion rules—random diffusion and edge-preferred navigation are applied to study the occurrence and universality of scaling defined by Eq.(2.1). Our findings, summarised in sections 2.3 and 2.4, confirm that for the occurrence of the power-law behaviour in the scatter plots in Figs. 2.2, 2.4, 2.6, 2.7, 2.8, 2.9 a

certain preferential behaviour in the diffusion is necessary. Such preference is either induced by the topology, i.e., the dispersion of node connectivity which directly influences the random diffusion process, or by local alteration in the diffusion rules, in our case the edge-preferred diffusion is related to the degree of the node down the link. The results of scaling of fluctuations obtained for the random graph shows that even small differences in nodes connectivity lead to the occurrence of scaling.

Moreover, for the random diffusion process and for the fluctuations of node activity, the span of the scaling region in the plots is directly related to the span of node connectivity resulting from $< h_i > \sim k_i$ property. Knowing the number of points belonging to each group of points shown in Fig. 2.2 we can reconstruct the degree distribution of the underlying structure. These results suggest that the scatter plots can be used for a network structure recognition for the random diffusion process taking place on the network, however without the insight in the exact topology.

We demonstrated systematic dependence of the scaling exponents with the width of the acquisition time window. Generally, the exponents increase with larger time windows, however, the functional dependencies $\mu(T_{WIN})$ are related to the network and to the diffusion rule. Even for the same type of the navigation rule the scaling properties of fluctuations for a scale-free network or a random graph may look very different. For instance, we show that the edge-preferred navigation rule induces the scaling of flow fluctuations. However, the character of these fluctuations depends on the underlying structure, i.e., two scaling regimes for the scale-free network and one for the random graph.

The $\sigma(< f >)$ dependence with two scaling regimes can be viewed as an indicator that there are two types of processes taking place on a network. In

our case the D navigation rule forces packets to avoid links leading to hub nodes. Thus, the fluctuations of flow on these links were almost uncorrelated with the flow of packets in the other parts of the network. As a result we found persistent $\mu = 0.5$ regime even for very long time windows T_{WIN}. The existence of two μ regimes were also found for fluctuations of nodes' activity in [62]. A very efficient 2-depth search navigation algorithm used there brought about that some nodes were receiving packets only when they were the destinations of those packets. Hence, the dynamic in such nodes is reduced to the random deposition, where the average number of visits grows linearly with time $< h > \sim t$, and the σ increases as $\sigma \sim t^{0.5}$, providing $\mu = 0.5$. The second regime with $\mu > 0.5$ is related with many interplaying factors such as the main dynamic process, packets input rate R, time window of observation T_{WIN} or the underlying structure [81]. Moreover, we found the number of links belonging to $\mu = 0.5$ regime changes with T_{WIN}. This implies that the traffic on given parts of a network can be viewed as uncorrelated with other parts of the network only for given observation time window T_{WIN}, which can be important in particular measurements.

We also presented arguments that in structurally inhomogeneous networks, such as the scale-free structures, for the time series measured at network edges (i.e., in the case of edge-preferred navigation) the scaling features are different from those obtained for the node activity fluctuations. Therefore, it is important to make the distinction in the analysis of the empirical data of the traffic on the Internet, where usually the flow along an edge is monitored. We would like to stress that for the purposes of this work the network structures that we studied have low clustering (vanishing in the $N \to \infty$ limit). In more realistic packet traffic models both in-depth search algorithms and correlated network structures may lead to additional features

in the scaling of the noise and flow time series [62, 82].

Chapter 3

Growing Trees in Internet Discussions

We present an empirical study of the networks created by users within internet news groups and forums, and show that they organise themselves into scale-free trees. The structure of these trees depends on the topic under discussion; specialist topics have trees with a short shallow structure, whereas more universal topics are discussed widely and have a deeper tree structure. For news groups we find that the distribution of the time intervals between when a message is posted and when it receives a response exhibits a composite power-law behaviour. From our statistics we can see if the news group or forum is free or is overseen by a moderator. The correlation function of activity, the number of messages posted in a given time, shows long range correlations

connected with the users' daily routines. The distribution of distances between each message and its root is exponential for most news groups and power-law for the forums. For both formats we find that the relation between the supremacy (the total number of nodes that are *under* the node i, including node i) and the degree is linear $s(k) \sim k$, in contrast to the analytical relation for Barabási-Albert network.

3.1 Introduction

One of the most important features of the internet is the opportunity it offers people to exchange opinions with one another. Now anyone can participate in a discussion or debate on-line and the global reach of the internet allows a single person's opinion to be shared with people from all over the world. Thus each of us can now be a source of information, not only for our relatives and friends, but for the whole world. We can offer our opinion to a very wide range of people and receive feedback on this opinion. Thus internet discussions are potentially important in helping to shape people's opinions and behaviour and in the spreading of ideas and information. In this way the internet is a medium which is very different to traditional media such as newspapers, radio and television. The use of the internet has led to an explosion of interest within other academic disciplines in phenomena such as social contagion, viral marketing and stealth marketing. Despite this importance, scientific research into internet discussions has been rather limited.

There have only been a few scientific papers examining internet discussion networks. Makowiec and Bykowska [27] considered the three most popular blog web pages in Poland. They provided an analysis of the network structure of blogs and gave a sociological explanation of the results. In related work, Zhongbao and Changshui [28] examined the network properties of bul-

letin board systems (BBS), which are similar to the news groups examined in this paper. They [28] studied a network in which edges were between users, and were able to identify distinct communities within the network of users. BBS and users' networks were also studied by Goh *et al.* [29], who found intercommunities and intracommunities with different topological properties. The intracommunity was a homogenous system, in which members all knew each other, while intercommunities were characterised by a power-law degree distribution. Capocci *et al.* [30] investigated the largest internet encyclopedia, Wikipedia. A bow-tie-like, scale-free structure with almost neutral mixing was found. Only small and medium nodes exhibited linear preferential attachment. Valverde and Solé [31] focused on technology development communities, such as open source communities, by looking at e-mail exchanges. Non-local growth rules based on a betweenness centrality model were examined and compared with the empirical data. The temporal properties of e-mail exchange groups were studied by Barabási [50].

Internet forums and news groups are similar to BBS networks, but in contrast to previous work [27, 28, 29, 31], here we place an edge between messages and focus on the network of ideas or opinions posted by users, rather than networks between the users themselves. In this way we obtain tree-like networks with a central topic, the root node, and the surrounding threads.

In the last few years there has been much work characterising the topology of real networks [2, 6, 8, 13, 12, 23, 5]. This work has shown that our world is more complex than we had originally imagined and has led to the development of the idea of a complex network. The most significant result arising from these studies is that a power-law degree distribution appears to be very common in real complex networks.

In this chapter we examine empirically a variety of basic structural and temporal properties of the internet discussion networks that are created by internet users. The chapter is organised as follows: in the next section we introduce the different types of internet discussions, and describe the scope of our empirical study; in section 3.3 we describe our results, both topological and temporal; before summarising our findings in the final section.

3.2 Types of internet discussions

Almost all internet discussions take place through the medium of forums. Most internet information portals, on subjects such as politics, accidents, sport, etc..., include forums as part of their website. New topics are introduced to these forums on a daily basis. Some portals give people fixed forums to discuss common topics such as love, work and sport. Users cannot put un-moderated messages into these forums; most forums have a person or computer program - a *moderator* - that acts as a referee for the comments posted, and rejects posts that are deemed unsuitable.

Another type of internet discussion are *news groups*. They contain an enormous number of topics to be discussed. Originally, the news groups were accessed by a computer program - a client, usually built into an e-mail program such as Microsoft Outlook or Mozilla Firefox. Nowadays access is much simpler and it is provided through a web browser, e.g. Google news groups. Usually users of the news groups visit them regularly, take part in a few discussions and from time to time open their own topic. The administrator of a news group's server can block access to the server to people that break its rules.

A third popular medium of internet discussion is a blog. There are a

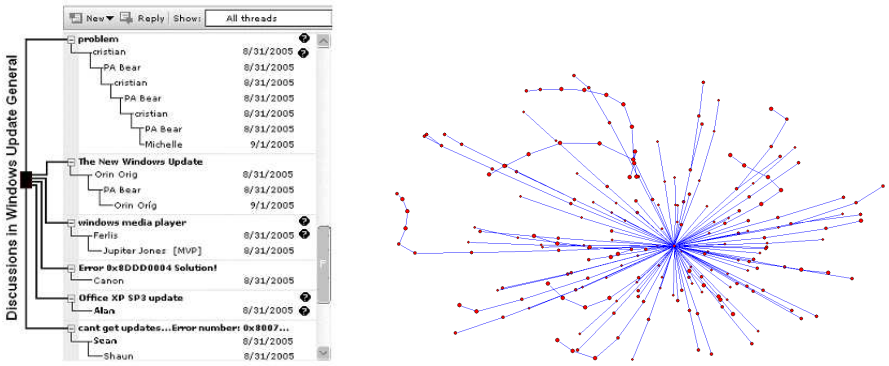
number of websites where people can establish their own blog, which usually takes the form of a diary of their day-to-day life. Other people can discuss the blogs and express their opinions about them to other readers or the owners of the blogs themselves. The bloggers are usually able to place links to other blogs, which are either on a topic related to their blogs or of general interest to them, on their website. These links create a network of blog owners [27].

3.2.1 Typical construction of internet discussions

For a news group and an internet forum the topic of the discussion is a root node. The threads that initiate new discussions are connected directly to the root node. When people contribute to a forum they can either write a commentary on a previous opinion or start a new thread. Every message is indexed by the name of the author, its place in the hierarchy and its time of posting. In this paper we treat each message as a node. We create a link between a message and a responding or answering message. This procedure creates a tree-like structure. Fig. 3.1 shows a typical structure of a small internet discussion.

The organisation of the threads on the main page of the online discussion web page varies depending of the provider of the service. Some of the internet discussions advertise on the main page the threads that received the latest messages. This allow new users to quickly join and comment on the most recent opinions connected to even old topics. Other, simply display the list of threads sorted according to their time of creation, similar to the one shown in Fig. 3.1. We study here the examples of the former (forums) and the latter ones (news groups).

We have investigated the network structure and temporal properties of 3 forums and 15 news groups, whose data was collected from two sources:



(a) (b)

Figure 3.1: (a) The typical structure of an internet discussion. The black lines show links between messages and the responses to them. (b) The tree-like structure of the small news group Physics, $N = 220$ nodes.

- The internet forum on the web site www.onet.pl
- The news groups on the server news.student.pw.edu.pl

In the case of news groups the people who can contribute to a discussion is limited by the fact that only computers inside the university's network are allowed to login. Because of this only students and academic staff have access to these discussions and there are around $30,000$ of them each year. We did not measure the number of active users of these news groups, but we suppose that there are less than $5,000$.

The internet forum on www.onet.pl is part of the largest Polish news portal, which is used by around 50% of all Polish internet users.

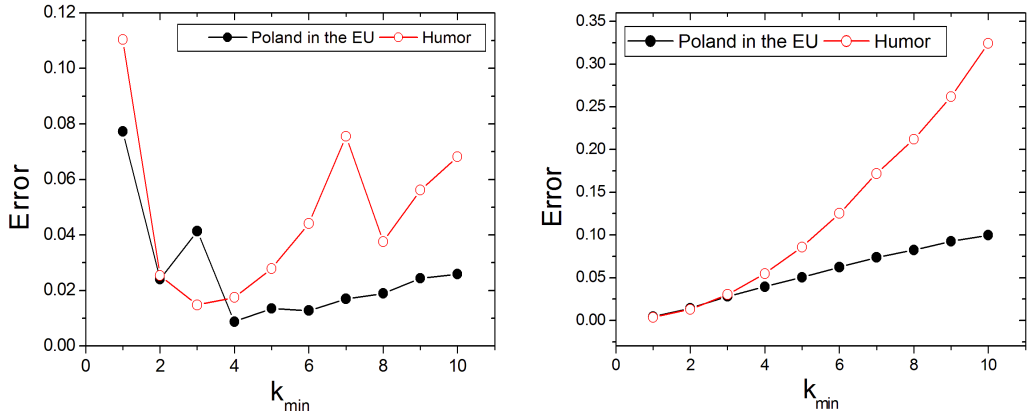
Almost all internet discussions that we have collected, were created at different times. However for internet forums the period of collected data is between $2001 - 2005$ and for news groups the period is $2002 - 2005$.

3.3 Empirical results

We studied empirically a number of properties of real internet discussions. Our networks are trees and consist of messages, not users, so we are unable to study properties such as the clustering coefficient or to define communities. Similarly it would be fruitless to study node mixing or the betweenness centrality, which were studied in [27, 28, 29, 31]. Thus, the structural properties we examine are the degree distribution, the average and maximal distance, the distribution of distances between messages and their root nodes, and the average supremacy [54] of each node as a function of degree. The temporal properties we examine are the distribution of time between a message being posted and there being a response to it, the activity time series and its correlation function. With the temporal properties we distinguish between network time; time in which one message is posted in one time step and message i is added at time i, and real time; the actual time that messages were posted in our experimental data. Where appropriate we present results for both an internet forum and a news group, for the largest and most representative examples.

All fitting procedures were performed in Origin scientific analysis software, which uses Levenberg-Marquardt non-linear fitting curve method [100]. In most real datasets the distributions of given properties do not hold single behaviour in whole fitting range. For instance, the degree distributions shown in Fig. 3.3 display power law character starting from $k = 2$ and have noisy tails, due to the system's finite sizes. Thus, to obtain most reliable results we follow a simple idea to maximise the fitting range and minimise the error bars of the fitted curves. In particular, the exponents of the degree distributions were found for k_{min} for which the error bars were smallest, where k_{min} is the minimum value of k for which we perform calculations.

The dependence of the error bars on the k_{min} found for 3.3 is shown in Fig. 3.2a. Note, that a similar approach would not be possible for the maximum likelihood estimator (MLE) method [58, 59] (Fig. 3.2b), due to a strong error dependence on the number of samples n, $\sigma = \frac{\alpha-1}{\sqrt{n}} + O(1/n)$.



(a) (b)

Figure 3.2: The dependence of the error bars on k_{min} for Levenberg-Marquardt method (a) and the maximum likelihood estimator (MLE) (b).

3.3.1 Degree distribution

All the networks we examined were found to have power law degree distributions

$$p(k) \sim k^{-\gamma}. \tag{3.1}$$

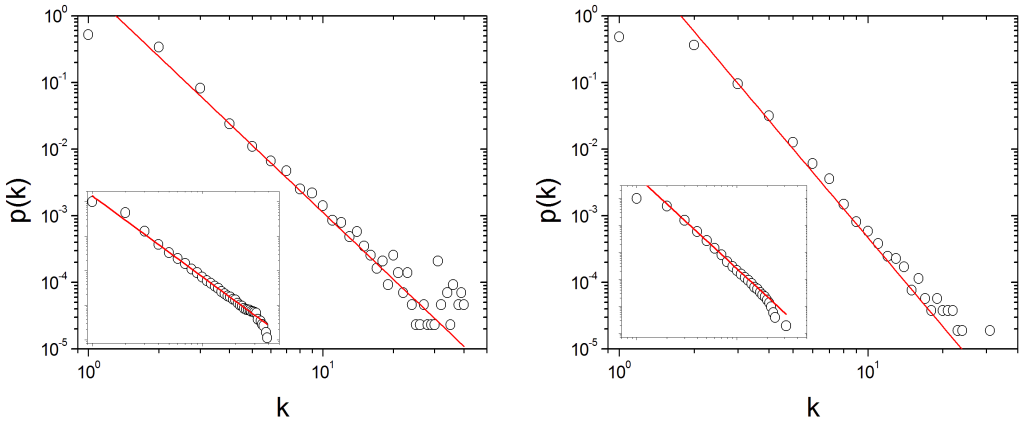
Table 3.1 lists the topics of these discussions, their size N, the exponent γ of their power law degree distribution, the maximal distance R_{max} from the root node, the ratio of the number of threads n_1 over the total number of messages N and finally the average distance of all nodes in the network from the root node $< r >$.

The internet forums generally have a lower exponent γ than the news groups. In particular, the exponents for forums are in the range $3.28 < \gamma < 3.34$ and for news groups $4.36 < \gamma < 5.62$.

Fig. 3.3 shows a typical degree distribution for the forums and the news groups. The networks have few nodes with high degree, even for the larger networks, with only 7 networks having a maximum degree $k_{max} > 30$. For the news groups the largest degree is around 20.

The power law character of a distribution that spans over less that two decades is usually unclear and sometimes questionable. However, in the case of our datasets considering high values of obtained γ exponents the sizes of recorded discussions should be in the hundreds of thousands or even millions of nodes to see scaling region spanning over two and more decades. Moreover, the preference in subjects that users discuss is an important preferential attachment mechanism, which usually leads to a power law distribution of connectivity. Finally, we studied 18 representations of the internet discussions, where number of nodes varied from 11 to 52 thousands. This enabled us to obtain robust degree distributions, for which character could be confirmed in a number of cases.

In all networks we examined, the number of nodes with degree 1 was similar to the number of nodes with degree 2, that is $p(1) \approx p(2)$. This seems to be because people like to argue and preferentially create chain structures in threads, and also because people also sometimes respond to their own messages. This behaviour creates more nodes with degrees $k = 2$, $k = 3$, etc... and shifts the degree distribution towards higher values of k.



(a) (b)

Figure 3.3: The degree distribution for the internet forum Poland in the EU (a) and the news group Humor (b). The exponents γ are $\gamma = 3.37 \pm 0.01$ for forum Poland in the EU (a) and $\gamma = 4.43 \pm 0.03$ for news group Humor (b). The inset figures show the cumulative degree distributions.

3.3.2 Time interval distribution $T(\tau)$

Internet users visit news portals to update themselves on the recent news, and some of them will discuss this news in a forum. In most cases they will only discuss the very latest news, and only very interesting topics will be discussed by users over a long period of time. The same rule applies for messages, only interesting or very controversial opinions are discussed for a long time period. This is why messages age very quickly and are soon forgotten. The influence of aging is the reason for the large exponent γ in these networks and for the lack of nodes with large degree.

There have been a number of attempts to model the effect of aging, see for instance, [16, 51, 52, 53]. The fundamental quantity in these models is $\pi(k, t, \tau)$, the rate of attaching a new node to a node of degree k and age τ at time t. All these models assume that $\pi(k, t, \tau)$ a separable function of

the degree and the age of the node. In particular, Dorogovtsev and Mendes [51, 52] modelled this aging by assuming that incoming nodes are linked to a node with degree k and age τ with rate $\pi(k,t,\tau) = A(\tau)k$, where $A(\tau)$ is some aging function, given by

$$A(\tau) \sim \tau^{-\beta}. \tag{3.2}$$

They found that the degree distribution of this network remained power law, $p(k) \sim k^{-\gamma}$ in the large time limit but with an exponent γ that strongly depends on the exponent β in the aging function [51].

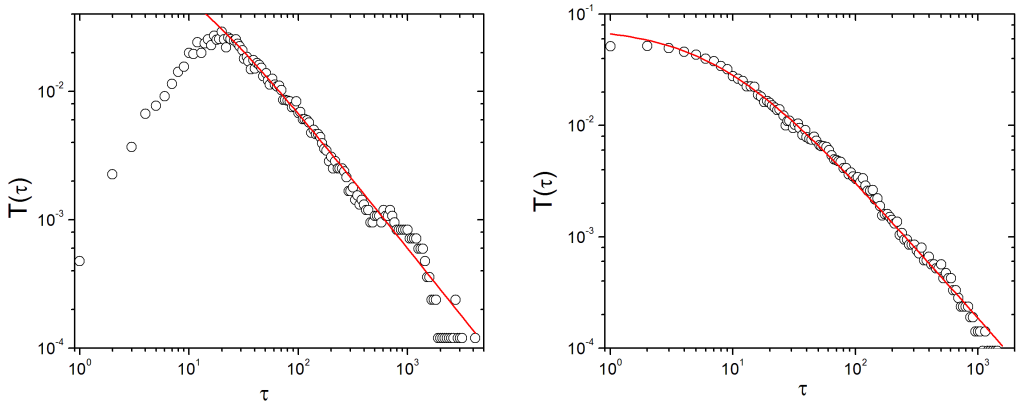
Unfortunately, $A(\tau)$ is not easily measured empirically, as attempts to verify that some real networks were grown by preferential attachment without aging clearly illustrate [55]. Instead, we have measured a related quantity, [56], the time interval distribution. This is the distribution of times between a message and a response, for all the internet discussions. More precisely, where a message j, posted at real time t_j, receives a response i at real time t_i, we have studied both the distribution of the real time interval $\tau = t_i - t_j$, $T(\tau)$ and the distribution of network time interval $i - j$. The distribution $T(\tau)$ is related to the degree distribution at time t, $p(k,t)$ via

$$T(\tau) = \int w(k,t,\tau)p(k,t)dkdt \tag{3.3}$$

where $w(k,t,\tau)$ is the probability that a node of degree k at time t waits another τ time steps before gaining an edge. This latter function contains, implicitly, two temporal processes, the natural waiting time for a new edge which exists in all growing network models, plus the effect of the aging identified and modelled in [16, 51, 52, 53]. However, for $1 << \tau << t$, we expect that the effect of the former will be exponential in τ on $T(\tau)$ whereas if there is appreciable aging, this will manifest itself as a fat tail in $T(\tau)$ for large τ.

In fact our results for real time show that in an internet news group messages age and have a power law $T(\tau)$. In Fig. 3.4 we show the time interval distribution

$$T(\tau) \sim [\tau + \tau_0]^{-\delta}. \tag{3.4}$$

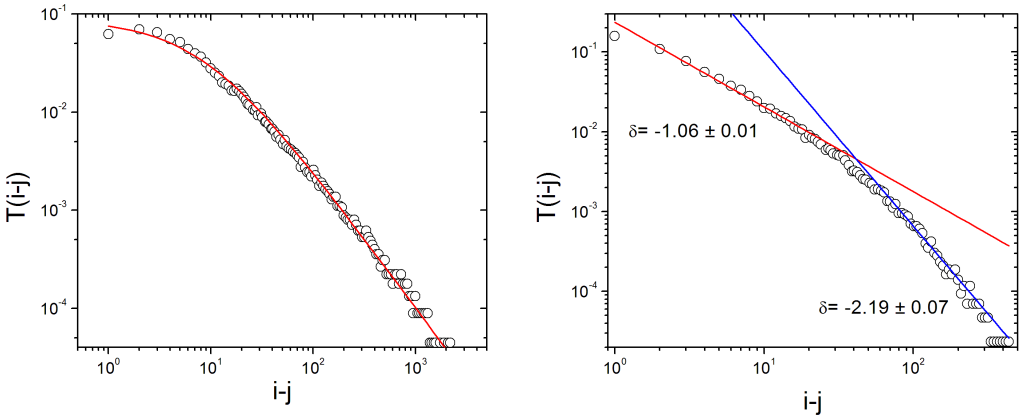


(a) (b)

Figure 3.4: The time interval distribution in real time for (a) the forum Poland in the EU and (b) the news group Humor. The exponent $\delta = 1.07 \pm 0.05$ for (a) and $\delta = 1.26 \pm 0.04$ for (b). The real time unit is 1 minute.

The positive slope of the curve in Fig. 3.4a for small time intervals results from the presence of the *moderator* in the forum www.onet.pl. The moderator has to check each message and this takes some time. Fig. 3.4b shows that Eq. (3.4) gives a good approximation to the empirical measurements.

In Fig. 3.5 we show the time interval distribution in network time and this merits two observations. Firstly, there is a change in the time interval distribution. For all news groups (but not for the forums) we obtained time interval distributions with two regimes of aging. For each news group there is a characteristic, cross-over time interval t_c after which messages start aging



(a) (b)

Figure 3.5: The time interval distribution in network time for (a) the forum Poland in the EU and (b) the news group Humor. The shape of Fig. (a) follows Eq. (3.4) with $\delta = 1.32 \pm 0.03$. Fig. (b) is described by composite power laws with exponents $\delta = 1.06 \pm 0.01$ and $\delta = 2.19 \pm 0.07$.

faster. This characteristic time is different for each network, however in almost all cases $20 < t_c < 60$.

Secondly, the shape of time interval distribution for internet forum is not affected by a *moderator* and exactly follows Eq. (3.4). This means that for small time intervals messages age slower and for large intervals faster but the change is smooth and without the critical point observed in news groups.

Two regimes of scaling found for news groups are probably related with the organisation of users' discussions. Usually, users discuss one thread during a single session (an evening for instance) and they rarely come back to the previous subjects. Thus, the first part of the network time interval distribution is related with the time intervals within sessions, when threads are mainly discussed. The second part of the distribution is built by time intervals of responses that came much later, posted by late users, somebody who wanted to add something at the end or by users posted off-peak. The

second mechanism that can be responsible for two regimes of scaling is the organisation of the threads in a browser. In the case of the news groups, new threads occur on the top of the main page screen, pushing down older ones. This establishes a list of threads sorted according to their time of creation. Hence, the new coming users that see a list of recently started threads are more likely to join one of the recently originated discussions.

For forums the mechanism of announcing is entirely different, because there the thread that received the latest *message* is placed on the top of the main page and the older ones are pushed down. Thus, the new coming users see on the top recently active threads that were possibly originated a long time ago. Hence, a user is more likely to join an older discussion and respond to an old message. These two different mechanisms of announcement of recently active threads could be the reason why there are two regimes of scaling for the news groups and single one for forums. These two mechanisms reflect also two different characters and approaches to the internet discussions.

Finally, the cross-over time t_c can be viewed as a value that separates time intervals related with users' responses posted within sessions from the time intervals that come from messages posted late or off-peak.

The power law behaviour of the time interval distribution was studied by Barabási [50] for an e-mail exchange group. By simulating the types of activity of internet users, it was shown that only the *burst* activity results in power law distributions, $A(\tau) \sim \tau^{-\delta}$, where $\delta = 1$. Fig. 3.5b shows that for small network time intervals the index δ is close to 1. For all news groups $\delta \in (1.0, 1.5)$. Because of the *moderator* the results for internet forums are disturbed, however the value of $\delta = 1.32$ is still close to 1 (Fig. 3.5a).

We also studied the relationship between the network time interval and the real time interval. Of course these are related by the fact that the activity

$n(t_i)$, which is the number of messages that were posted in time t satisfying $t_i < t < t_{i+1}$, can be approximated by $n(t_i) \approx \frac{i-j}{t_i - t_j} = \frac{\Delta M_{ij}}{\Delta t_{ij}}$, where ΔM_{ij} stands for the number of messages posted within the time interval Δt_{ij}. Our empirical results show that, as would be expected, on average the relation is linear with

$$n(t_i)(t_i - t_j) \sim \epsilon(i - j) \tag{3.5}$$

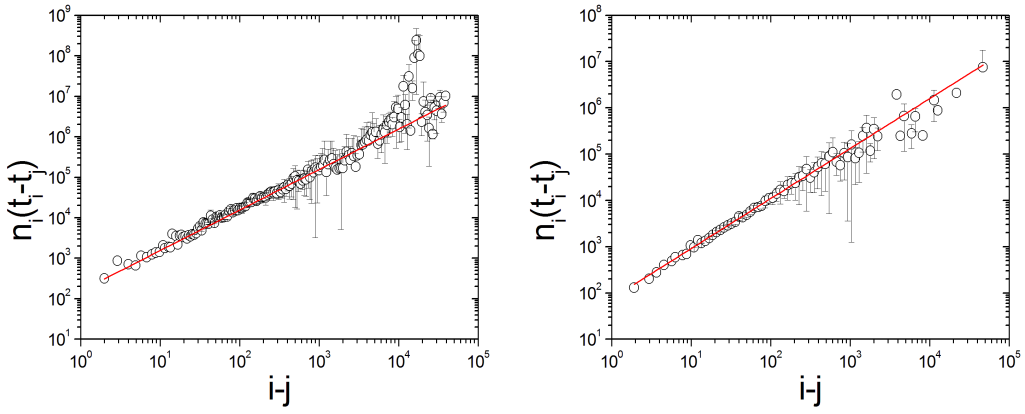
with $\epsilon = 1.07 \pm 0.03$ for the internet forum Poland in the EU Fig.3.6a and $\epsilon = 1.03 \pm 0.02$ for the news group Humor Fig.3.6b.

For small $i - j$ intervals the number of messages posted in time $n(t_i)$ is closely related with the temporal activity of the discussion and follows linear relation well. However, due to the high volatility in the users activity (Fig. 3.7a) the large deviations from the linear behaviour are observed for large $i - j$ intervals.

3.3.3 Activity

We define the activity of a news group as the number of messages posted in a given time interval. In Fig. 3.7 we show the activity time series and the distribution of activity for the discussion forum Poland in the EU. Here we have measured the number of messages posted in one hour. As one can see, there is a variation of activity over a wide range of scales.

The peak, in which 898 messages were posted in a single hour, corresponds to the time when Poland was voting in the accession referendum to the EU. This hiatus can be seen in the activity distribution, corresponding to the points to the right in Fig. 3.7b, away from the main curve. We examined the distribution of activity for all our news groups, and found that all the



(a) (b)

Figure 3.6: The average value of the real time interval multiplied by the activity as a function of the network time interval for (a) the forum Poland in the EU and (b) the news group Humor. The data was logarithmically binned and the error bars express the standard deviations from the averages in bins.

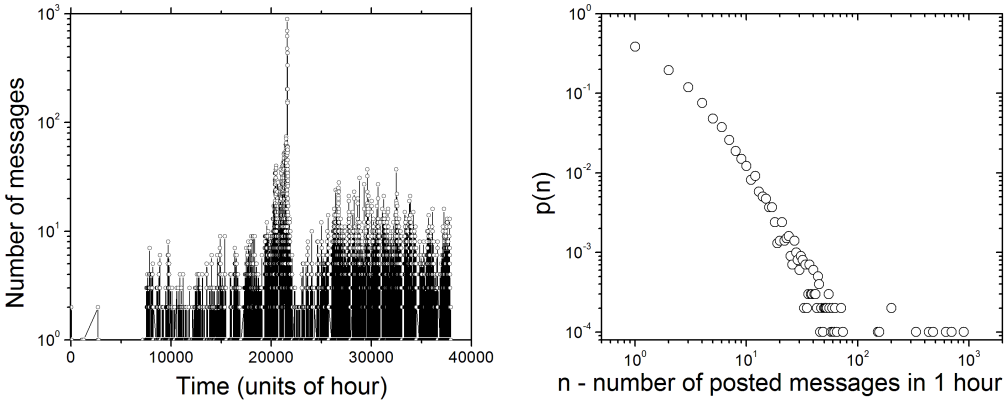
distributions were fat-tailed, with distributions that ranged from power law to Kohlrausch, $\sim exp(-\tau^a)$, with $0 < a < 1$.

We have measured the correlation function $C(\tau^*)$ of the activity time series, $n(t)$, defined by

$$C(\tau^*) = \frac{1}{i_{max}} \sum_{i=0}^{i_{max}} [n(t_i) - < n >][n(t_i + \tau^*) - < n >] \qquad (3.6)$$

where $t_i = t_0 i$ and $< n >$ is the mean number of messages posted per time t_0 over the whole time series. We studied $t_0 = 1hour$ and $t_0 = 1day$.

All the internet discussions indicate a correlation for $\tau^* = 24$ hours, which shows the daily routine of the internet discussion users (see for instance Fig. 3.8b). We also found a weak correlation for news groups on the time scale of one week, which is probably connected to the higher activity over a weekend. This is somewhat less pronounced, as Fig. 3.8a illustrates. Some news groups



(a) (b)

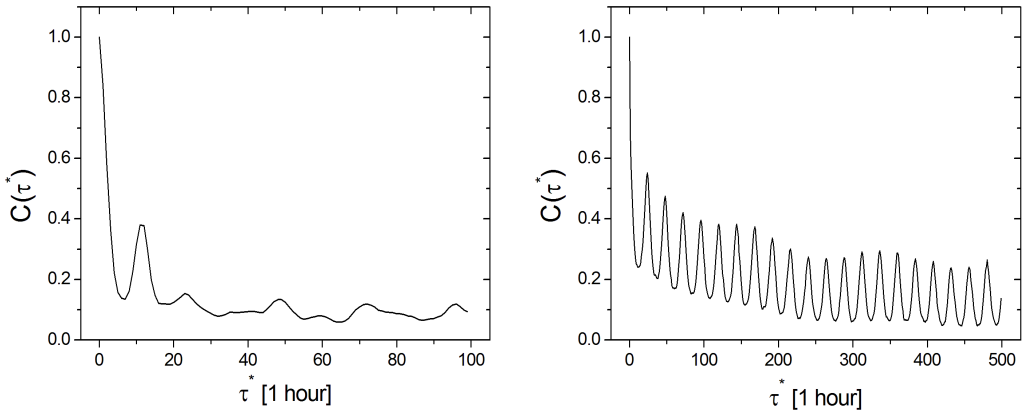
Figure 3.7: The activity time series (a) and the activity distribution function (b) for the forum Poland in the EU.

also show correlations for very long times, for instance for τ^* equal to 180, 270 and 365 days. These were seen in news groups that are only used by students and these long correlations are connected the academic holiday and semester structure. There is an interesting correlation for $\tau^* = 12$ hours in the forum Poland in the EU. This correlation is generated by the before-after work activity of the discussion users.

3.3.4 The distance distribution $D(r)$

$D(r)$ is the distribution of the number of edges between each node in the network and its root node (the central topic of the forum). For all the networks the maximum distances are small. Almost all the news groups exhibit an exponential $D(r)$, such as that illustrated for the news group Electronics in Fig. 3.9b. Of the news groups, only Humor has a distance distribution close to a power law.

The distance distributions for forums are modified by the software used



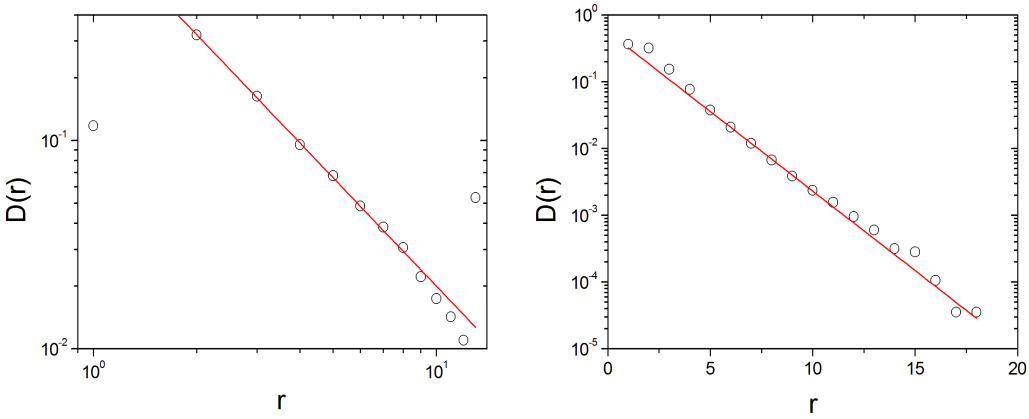
(a) (b)

Figure 3.8: The correlation function $C(\tau^*)$ for (a) the forum Poland in the EU and (b) the news group Humor, with a time step $t_0 = 1$ hour.

to manage the forum, which only allows a maximum distance of $r = 13$. A message that somebody wants to post to a message with $r = 13$ is added to previous the message with $r = 12$. This results in the large value of $D(13)$ seen in Fig 3.9a. Nevertheless, the distance distributions seams show power law character with an exponential cut-off resulted from the maximum allowed distance, as Fig. 3.9a illustrates.

Due to the maximum distance permission we are unable to definitely confirm this character of the distance distribution. We cannot support also this claim through the findings for the news groups. The difference in the behaviour of the distance distributions for forum and news groups is probably related to the main page threads announcement mechanism described in sec 3.3.2. For the forum, the thread with the latest *message* is placed on the top of the main page pushing down the older, which allows even old threads to acquire new comments that could possibly increase the maximum distance if the blockade was not in operation. In the case of news groups, the most

recently started *thread* is in the top of the main page pushing down the older ones, which definitely shortens their life time. However, in both cases the rate of inserting new threads speeds up the aging process of the older threads, which has impact on the length of a discussion. Thus, the discussion's length should be proportional to the rate of inserting new threads. Indeed, Fig. 3.10 shows the exponential relation $n_1/N \sim e^{-<r>/<r_0>}$ between the relative number of threads n_1/N and the average length of the discussion $< r >$.



(a) (b)

Figure 3.9: The distance distribution $D(r)$ for (a) the forum Poland in the EU and (b) the news group Electronics. The distribution for the forum has a power law behaviour $D(r) \sim r^{-\upsilon}$ with exponent $\upsilon = 1.73 \pm 0.02$ and the distribution for news group has an exponential behaviour. The value of υ exponent was obtained by the same method as discussed in section 3.1.

The ratio n_1/N (Table 3.1) shows how many threads are created as a fraction of all posted messages. A small value indicates that internet users are focused on the existing threads and they are prone to continuing the previous discussions. Large values show that there is almost no discussion, users place an offer or question and expect only answers to them. A related parameter that describes a discussion is the average distance from the root

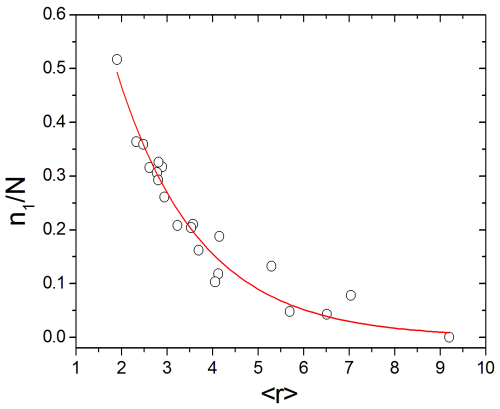


Figure 3.10: The ratio of the number of threads n_1 to the total number of messages N as a function of the average distance from the root $< r >$. The curve is fitted an exponential function $f(< r >) \sim e^{-<r>/<r_0>}$, where $< r_0 > \approx 1.82 \pm 0.13$.

node $< r >$ (Table 3.1). For small value of $< r >$ the discussion is not engaging and users probably just exchange information. For large $< r >$ vigorous discussions are taking place. The ratio n_1/N describes the behaviour of the internet users and the average distance $< r >$ describes the topological consequences of this behaviour. There is a functional dependence between them and Fig. 3.10 demonstrates this. The values of n_1/N and $< r >$ show the kind of discussion we examined, technical, where people are interested only in exchanging goods, information and looking for help, or theoretical, where people introduce ideas, share opinions and argue with others. Good examples are two news groups Games and Games.CS. The Games news group is a general discussion about games, where $< r >$ is rather small. The news group Games.CS is dedicated to only one game's fans, *Counter Strike* and its value of $< r >$ is much higher than for Games news group, which suggests that the fans are more strongly engaging within the discussion.

3.3.5 The supremacy function $s(k)$

The supremacy s_i of node i is defined as the total number of all nodes that are not older than i and can be linked to it by a directed path (including the node i). For tree-like networks this means that the supremacy s_i of node i is the total number of nodes that are *under* the node i, including node i. In other words the supremacy s_i is the total number of nodes in the sub-tree started by node i. The supremacy function $s(k)$ is the average supremacy of all nodes of degree k. In [54] it was shown that for the Barabási - Albert model [14],

$$s(k) = \frac{m}{m+1}\left(\frac{k}{m}\right)^{m+1} + \frac{1}{m+1} \tag{3.7}$$

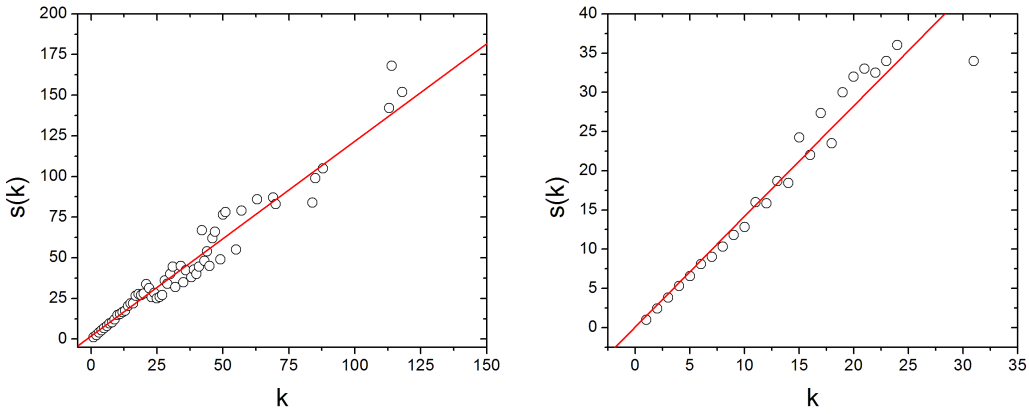
where m is a number of links created by an incoming node, and for trees, when $m = 1$

$$s(k) = \frac{1}{2}k^2 + \frac{1}{2} \tag{3.8}$$

For each network we measured the average value $s(k)$ for a particular degree k. Fig. 3.11 shows that for the internet discussions relation $s(k)$ is not $s \sim k^2$, but relation is linear $s \sim k$. The result $s \sim k^2$, obtained for Barabási - Albert model, which does not include aging of nodes. This suggests that the linear dependence between supremacy s and degree k could be triggered by the aging of nodes.

3.4 Conclusions

Internet discussions are tree-like networks, whose degree distributions are described by a power law function. The networks are growing in time and because the posted messages become out of date naturally, the nodes are aging. For news groups the distribution of the network time interval between



(a) (b)

Figure 3.11: Average supremacy $s(k)$ against degree k for (a) the forum Poland in the EU and (b) the news group Humor. (a) and (b) both follow linear functions with slopes 1.19 ± 0.02 and 1.41 ± 0.07 respectively.

a message and a response has two scaling regimes. The small time interval regime probably corresponds to responses within one session of the discussion, from people currently on-line, which corresponds with *the burst activity* studied in [50], and the behaviour for large time intervals is generated by messages posted later or off-peak by new users arriving on-line. For the internet forums the time interval distribution is described by $T(\tau) \sim [\tau + \tau_0]^{-\delta}$ and shows a smooth behaviour.

The time correlations within the activity time series show that the activity of internet discussion users is integrated with users‘ daily routines on both 12 and 24 hour scales (Fig. 3.8). The most characteristic type of correlations is different for forums and news groups. We suppose that strong $\tau^* = 12h$ correlations found for forums (Fig.3.8a) are related with *before - after work* type of activity. Much weaker $\tau^* = 24h$ correlations can indicate lack of stable community of users gathered around the forum. On the other hand

the most characteristic correlations for $\tau^* = 24h$ found for the news groups (Fig.3.8b) may let us conclude that these students' online discussions have a stable group of active users that every day devote their time to discussing together. The stability and regularity of these groups are probably strengthened through the specific character of student communities and similarities in students' daily routines. On publicly open forum heterogeneity of users is higher and their availability is much more diverse. Moreover, the forum was held by the news portal and 12 hours correlations might be related to news presented there. An important piece of information from the perspective of users might drive hectic *before - after work* exchange of messages, whilst others might induce barely noticable activity, which finally may lead to the strong correlations for $\tau = 12h$ and much weaker for $\tau = 24h$. These measurements could help us to define an optimal place and time of operation for people interested in marketing goods (viral marketing) or services to internet users.

The distance distribution exhibits exponential character for most news groups, which means that discussions are not deeply embedded within larger tree structures. The results for internet forums on www.onet.pl show the intervention of the software employed, which only allows a maximum distance $r = 13$ in its forums. However the distance distributions for these groups exhibit a power law behaviour. In section 3.3.4 we discussed how the different main page announcement mechanism can trigger these results. While the list of threads for the news groups is static and they are organised according to their time of creation, the list of threads for the forum is dynamic and the thread that received the latest message is placed on the top of the main page list. Hence, threads in the news groups are inevitably aging but threads in the forum can be "refreshed" and attract new messages.

Secondly, these results can be also understood by considering the topics of these discussions. The news groups mostly contain closely defined, themed, discussions which are often very technical and frequented by experienced users. Consequently answers are very short and directly address the problem. Thus, the average distance $< r >$ is small. In contrast, internet forums have a wide range of users, who usually want to discuss and argue with others. This attitude towards discussion creates large and deep tree structures.

The length of the discussion can be considered as an indicator of its quality. In Fig. 3.10 we show how the average length of the discussion is related to the dynamic of inserting new threads and in sections 3.3.4 we discuss how the different hierarchy of threads on the main page can influence this average length. These studies may deliver very useful information to the providers of the online discussions on how to shape the character of the service they administrate.

Internet discussions are an important source of data within social sciences. They allow the study of the topology of social connections and their temporal statistics [27, 28, 29, 30, 31]. Our study focused on the growing trees of messages, whose structure and temporal statistics, as we have shown, are related to the subject of the discussion and the day-to-day activities of users. Investigating the emerging, aging and dying of topics in discussion networks should yield data on people's interests - what people like reading or commenting on. This should give insight into the real dynamics of people's opinion change and exchange.

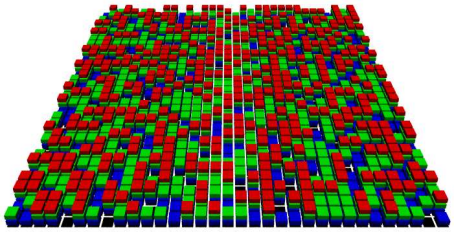
Table 1

No.	*Topic of discussion*	N	γ	r_{max}	n_1/N	$< r >$
	Onet Forums					
1	Poland in the EU	43027	3.34 ± 0.01	13	0.118	4.127
2	Opinions of Poles	36479	3.28 ± 0.03	13	0.103	4.062
3	Situation in Middle East	47075	3.29 ± 0.01	13	0.048	5.701
	News groups					
1	Trade	44266	5.62 ± 0.07	24	0.517	1.905
2	Politics	11706	5.44 ± 0.04	46	0.078	7.041
3	Humor	52525	4.43 ± 0.02	76	0.204	3.534
4	Off-topics	21940	4.53 ± 0.04	51	0.188	4.153
5	Linux	11049	5.43 ± 0.07	25	0.208	3.234
6	Pillory	40495	4.58 ± 0.02	62	0.132	5.299
7	Games	34080	5.42 ± 0.03	30	0.293	2.811
8	Games.CS	18976	4.36 ± 0.04	25	0.162	3.698
9	Programming	14560	5.44 ± 0.04	25	0.261	2.948
10	Music	12461	5.41 ± 0.09	20	0.359	2.481
11	Campus.Riviera	15431	5.03 ± 0.05	33	0.326	2.821
12	Campus.Ustronie	31170	5.14 ± 0.03	26	0.317	2.897
13	Electronics	28199	5.32 ± 0.05	18	0.364	2.329
14	Windows	13684	5.18 ± 0.08	32	0.210	3.575
15	Film	32923	5.04 ± 0.02	20	0.306	2.783

Table 3.1: We measured 19 internet discussions, 4 from the internet forum www.onet.pl and 15 news groups from the server news.student.pw.edu.pl. The columns contain the name of the discussion, the number of nodes N, the exponent γ of the power law degree distribution and the maximum distance R_{max} from the root node. Next column contains number of threads n_1 over all messages N and the last the average distance from the root node $< r >$.

Chapter 4

Self-Organised Criticality at the A&E Department



We present an analysis of one year's worth of empirical data on the arrival and discharge times at an UK Accident and Emergency (A&E) department. We find that discharge rates vary with the workload and that the distribution of the length of stay has a fat tail. A sand pile model is considered to show that the $A\&E$ department is a driven self-organised system, where the department staff manage their work time to cope with the department's occupancy. We use in our model a variable input space to mimic the queuing discipline related to different cases of accidents found in the department. The input space is defined by two parameters; its size $s \times s$ and the distance m from two nearest edges. We study the length of stay distribution for the sand pile model for both s and m parameters. We show that while s or m are increased the character of the tail of the distribution changes from power law to exponential one.

4.1 Introduction

In this paper, we present and analyse empirical observations of the arrival and discharge of patients at an UK Accident and Emergency (A&E) department. Our methodological approach is in keeping with the increasing trend to apply methods and perspectives from statistical physics outside the traditional boundaries of natural science, and in particular to social and economic systems. Statistical physics provides methods for moving from microscopic or individual elements to macroscopic or collective phenomena and can yield important insights into our understanding of social systems [101]. Much of the physics-inspired empirically based modelling has taken place in finance, because of the availability of large quantities of high resolution data in this field. Attempts have also been made to apply a similar approach to other problems where humans are the microscopic elements and a broad range of topics have been studied, including opinion dynamics [102], decision making [103], terrorism [104], pedestrian flow [105], correspondence patterns [106], and airline disasters [107].

In health care science, empirical results demonstrating power law signatures in data on hospital waiting lists has opened up the possibility of using physics-inspired models whose main focus is to give qualitative understandings of the systems [108, 109, 110, 111, 112]. In this paper we use a sand pile model [113] to mimic the service of patients in the A&E department, which provides us with some insight into the dynamic process.

4.2 Empirical observations

The dataset analysed here consists of the arrival and discharge times for 92,965 patients from an UK A&E department over the period April 1 2003

to May 31 2004. We focus on two observations that deviate from what would be expected if the system were working as a simple queue, or in a random manner. When the term *discharge* is used in this paper it refers to the event of a patient leaving the A&E department by being admitted to the hospital, transferred to another hospital, or being released. The term *completion time* or *length of stay* then refers to the time interval between the registered arrival time and the registered discharge time.

Arrival/Discharge rate. Two important metrics of an A&E department are the rates of arrival and discharge. The mean values of these rates can be measured by simply counting the total numbers of arrivals and discharges and dividing by the total time. Of course, in reality these values show huge variability with the time of day and the day of the week. To get around this we look at the relative discharge rate $\Gamma(n)$, i.e. *the number of discharges divided by the number of arrivals*, and how it varies with n - the number of patients in the department. In Fig. 4.1a we show the dependence of the relative discharge rate on the departmental workload. The values of the number of arrivals, discharges and patients in the department were obtained within the time window 1 hour. We note that the relative discharge rate is somewhat higher when there is a large patient occupancy in the system and lower when the workload is low.

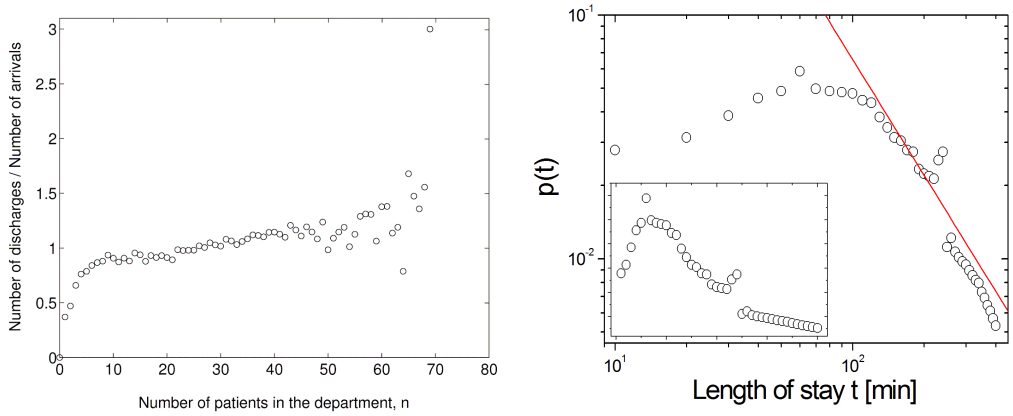
If we would consider an A& E department as a simple FIFO (first in first out) queue with a negligible service time, the number of discharges would be equal to the number of arrivals and thus, independent from the queue load. Furthermore, if the service capacities were not enough above some certain load level, we would observe a monotonic decrease in the discharges / arrivals rate from that point. However, none of these were observed. This means that the system adjusts itself to the occupancy of the department. One explana-

tion is that, as the department becomes busy, the staff work harder and some limited resources are devoted to ensure that patients' waiting times do not become unacceptably high, and hence the relative discharge rate increases. As the department becomes empty, doctors and nurses catch up on their administrative work and on accidents with lower priorities or time consuming ones. Thus, the relative discharge rate decreases.

These indicate that the A& E department exhibits a sort of self - adjusting properties. Patients arrive and are discharged. Each member of the staff manages their work to match themselves to the demands of the system so that the system just copes and spends the majority of the time in the centre of Fig. 4.1. Thus it seems natural to compare our empirical data with a model of self-organised criticality such as a sand pile model [113].

Length of stay distribution. In the UK, a national target for emergency care has been set. It states that after 2004, no patient should spend more than four hours in an A&E department [115]. In September 2002, the completion rate at four hours was 77% [116]. We have not compared our data with this target, because we are interested in the dynamics of this system, and focus on the distribution of the completion times. The length of stay distribution is shown in Fig. 4.1b. As physicists have come to expect for human systems of this kind, the tail of the distribution is fat and appears to fit a power law function $f(x) = ax^{-\gamma}$. Of course we can never know if this is a real power law since that would require A&E departments thousands of times greater than their real size and large numbers of patients spending years between arrival and discharge! Nevertheless, to allow us to pursue our analogy with a sand pile model in the next section, we fit the curve to a power law and find an exponent $\gamma = 1.58$. The distribution is also disturbed by the aming of staff to meet the completion time target. Hence the discontinuity in the distribution

when the length of stay equals $240min$, as staff redistribute resources to increase the number of patients that meet the completion time target.



(a) (b)

Figure 4.1: (a) Empirical number of discharges divided by number of arrivals, as a function of the number of patients in the A&E department, n. An increasing trend is clearly visible, which indicates that the system responds to a high occupancy by increasing the discharge rate. The higher variability for large values of n is an effect of the lower number of observations here. (b) The length of stay distribution for the patients in the A&E department. The fitted curve is a power law function $f(x) = ax^{-\gamma}$, where $\gamma = 1.58$. The inset figure presents the same data on a linear scale. The discontinuity in the tail of the distribution is exactly at 4 hours (240 minutes).

4.3 The sand pile model

The adjusting property of the A&E department discussed above resembles the self-organised behaviour of the sand pile model. Moreover, as we show it later, this model displays a similar response of the discharges / arrivals rate to the load of the system. One of the characteristics of the A&E department is that it is the first place where all patients come or are delivered. Because

the arriving patients have different illnesses and are in different conditions, the waiting list cannot follow a simple FIFO queue (First In First Out) and the positions of the patients waiting for the services are effectively reshuffled. Additionally, the length of stay at the department involves not only the waiting time but also the service, which varies from case to case. The data obtained from the A&E department does not distinguish between the waiting and service time, thus the dynamics of the sand pile model studied here necessarily represents both the behaviour of the virtual complex queue and service time at the A&E department. The relation between the studied model and the A&E department is rather virtual and the specific choice of the 2D model is the simplest possible version, which exhibits desired behaviour.

In our approach the grains represent the patients and we use a square $N \times N$ lattice, where $N = 75$. This particular size of the system is not related with any physical properties of the A& E department (i.e. the number of beds or seats) and its choice was driven by the trade-off between minimising the finite size effects (large system size) and the computational efficiency of our algorithm. In each time step of the simulation one sand grain is added to the system. The time step is defined on a macroscopic scale, which means that any reshuffling of the sand grains within the lattice, which we describe shortly, happens during a single time step. The simplest possible driving of the system (one grain per time step) was chosen to minimise the number of possible factors that might influence the dynamic of our model. While our idea is only to show the link between the self-adjusting property of the A&E department and the self-organised criticality, the influence of the faster/slower driving is negligible here.

We call square of the lattice a *bin* and it represents the position of patients in a complex treatment queue (waiting + service). Each bin is identified by

two coordinates i and j and it contains grains. The number of grains in bin (i, j) is given by $H(i, j)$, which is called *height* of the bin. If the height of the bin $H(i, j)$ where a grain was added or moved to is greater than or equal to 4, 4 grains from this bin are moved to the bin's 4 adjacent neighbours. This represents the progression of the patients within the complex queue. Thus,

$$H(i \pm 1, j) \rightarrow H(i \pm 1, j) + 1 \tag{4.1}$$

$$H(i, j \pm 1) \rightarrow H(i, j \pm 1) + 1 \tag{4.2}$$

$$H(i, j) \rightarrow H(i, j) - 4 \tag{4.3}$$

If a bin that releases sand grains is on the border, one or two grains fall outside the lattice. This represents the discharge of patients. When the number of grains is larger than the average, at some point a series of avalanches will pass through the system removing grains from the lattice. The system rests for some time, gaining new incoming grains and once again at some point a wave of avalanches goes through the system. This behaviour is repeated infinitely and the system balances all the time around an average number of grains on the lattice (Fig. 4.2b). Dynamics of this type are called *punctuated equilibria* [114].

Input space $s \times s$. In the usual model of a sand pile, a new sand grain is placed in the middle of the lattice, or is placed in a bin chosen at random. However, in this particular problem it would mean that each patient comes to the hospital with an average illness and needs an average length of treatment.

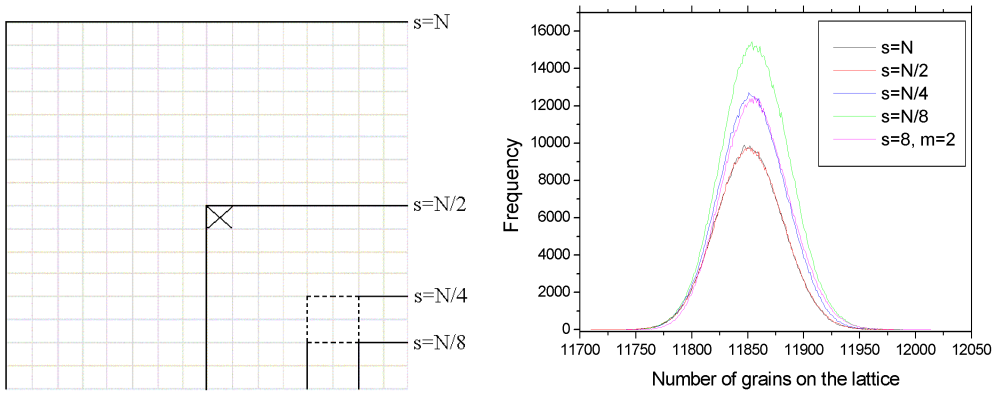
We modify these driving rules and use the input space explained in Fig. 4.2. In this way we are able to represent the diversity of patients presenting at the department, most of whom have minor illnesses but others have life threatening conditions. Thus, when we move the input space closer to the

border we increase the spread of the distribution of treatment times. In other words, the distance m from the edge of the lattice is responsible for setting up the virtual rate *quick/time demanding* treatment time. A few grains introduced at the edge of the lattice move inside the lattice and take a long time to be discharged but most are discharged after a few time steps.

In our simulations we use a square $s \times s$ space. We link the value of s to the size of the lattice, such as $s = N/k$, where $k = 1, 2, \ldots$. Increasing k we make the input space smaller and concentrated in a corner (Fig. 4.2a). For $s = 1$ the input space is a single bin. We allow the input space to "move" (Fig. 4.2a dash line square) by introducing a parameter $m = 0, 1 \ldots N$, which translates the input space m lattice sites away from the *two* nearest edges of the lattice. As the system is symmetric, parameters s and m completely define the input space. In the following, we take $m = 0$ unless stated otherwise. The different input spaces do not affect the average number of grains on the lattice (Fig. 4.2b).

Results. Our numerical simulations were made over $t = 5 \times 10^6$ time steps. Fig. 4.3a shows the results for the discharge divided by arrival rate against the number of grains on the lattice ,$\Gamma(n)$, for several different input spaces. For $s = N$ and $s = N/2$ the shape of the curve is exponential, but for smaller input spaces it becomes linear, similar to Fig. 4.1a for the A&E department. The linear growth shown in the inset figure is for $s = 8$ and $m = 2$. Despite the large number of time steps t, extreme values of the number of grains in the system n, are very rare. Thus, the large fluctuations on both edges are observed.

We calculate the length of stay distribution by collecting the input and output times for each grain. First we study the case when $m = 0$. We find for $s = N$ the distribution is an exponentially decaying function, but for



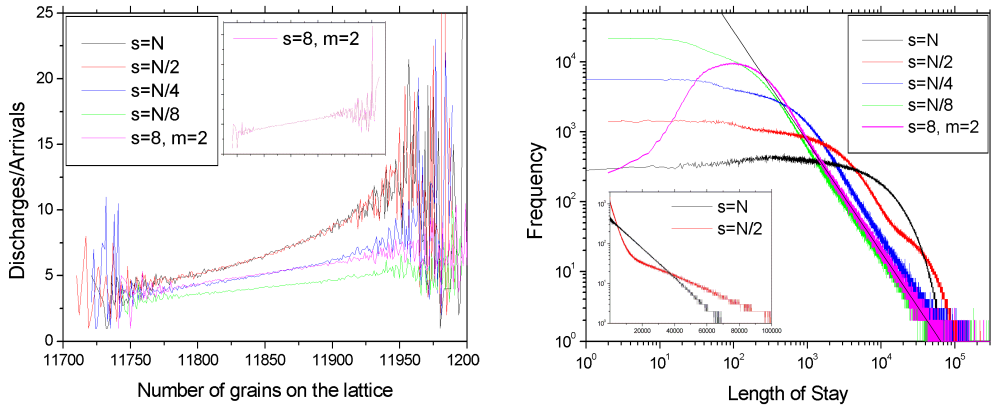
(a) (b)

Figure 4.2: (a) The definition of the input space $s \times s$. The square (dash line) shows the situation when the input space has no contact with borders. The cross shows the situation when the input space is limited to the single element (deterministic case). (b) The frequency distribution of sand pile grains for different input spaces.

$s = N/4$ and $s = N/8$, the tail of the distribution has a power law character, $f(x) = ax^{-\gamma}$ (Fig. 4.3b). For smaller values of s the tail of the distribution starts earlier. The threshold value of s between exponential and power law tail seems to be for $s = N/2$ (Fig. 4.3b inset figure). As we might expect, when the input space is moved from the borders ($m > 0$), a maximum value in the distribution emerges (Fig. 4.4b). The limit case is for $m = N/2$ (middle) and as we show for $s = 1$ the distribution is Poissonian (Fig. 4.4 inset).

We find γ to be slightly growing while parameters m and s are increased. For constant $m = 2$ and for $s = 1$, $s = 4$ and $s = 8$ the exponents γ of the power law tails of the length of stay distribution are $\gamma = 1.50$, $\gamma = 1.54$ and $\gamma = 1.58$ respectively. However, if we increase m and s too much the power law tail of the length of stay distribution almost disappears and it is impossible to obtain a reliable value of γ (Fig. 4.4a and b). The value of

$\gamma = 1.58$ found for $m = 2$ and $s = 8$ was used to fit the real length of stay distribution (Fig. 4.1b).

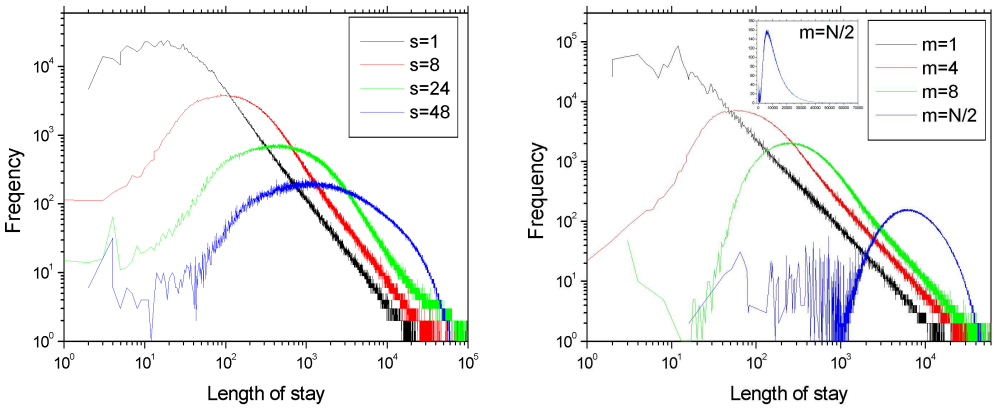


(a) (b)

Figure 4.3: (a) The discharge/arrival rate over the number of grains on the 75×75 lattice for different input spaces $s \times s$. The inset shows a linear $\Gamma(n)$ relation for input space $s = 8$ and $m = 2$. (b) The distributions of the length of stay for a sand grain on the lattice for different input spaces $s \times s$. The fitted curve for $s = 8$ and $m = 2$ is a power law function $f(x) = ax^{-\gamma}$, where $\gamma = 1.58$. The inset figure shows distributions for $s = N$ and $s = N/2$ in single logarithm scale.

4.4 Conclusions

In this chapter we have provided empirical evidence that for the UK A&E department the ratio between the discharges and arrivals is dependent on the number of patients in the department. While there are limited possibilities to move doctors and nurses from their current duties to support the A&E department, we believe the staff work harder during rush hours and catch up their administrative duties during quieter periods. We have shown that this behaviour exhibits many similarities to the sand pile model, where quieter



(a) (b)

Figure 4.4: (a) The distributions of the length of stay for a sand grain on the lattice for $m = 2$ and input spaces 1×1, 8×8, 24×24 and 48×48. (b) The distributions of the length of stay for a sand grain on the lattice for input space 1×1 and $m = 1$, $m = 4$, $m = 8$ and $m = N/2$ (middle).

periods when the system gains new grains are interrupted by busy periods with avalanches of activity when grains move out from the system. Each member of the staff makes his/her own "to do" list where certain priorities are given for each accidents. The process of prioritization is responsible for the burst activity displayed by human systems [50].

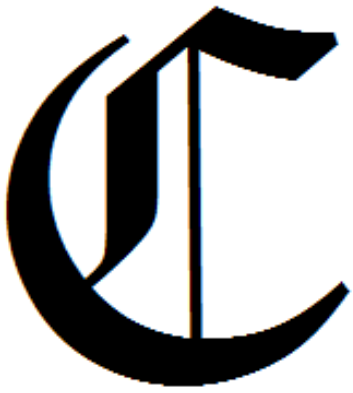
The length of stay distribution for the A&E department (Fig. 4.1b) is a Poissonian-like curve for short waiting times with a fat tail for long ones. Two facts can be responsible for that: the conditions of the presenting patients and the queuing discipline. We put both together by considering the input space close to edges of the lattice. In most cases a single grain leaves the system quite quickly, with average time related to parameter m. However, some grains move inside the lattice where they stay much longer and their waiting time contributes to the fat tail of the length of stay distribution. This mechanism mimics the range of conditions with which patients present

at the A&E department: simple and complex.

We studied here a small 75×75 lattice just to show self-organised behaviour of the UK A&E department. However, it would be strongly recommended in further studies to use a larger one and define s and m parameters as a ratio of the system size; $\tilde{m} = m/N$ and $\tilde{s} = s/N$. It would be interesting to find out how the exponent γ depends on the $\tilde{s}$ and $\tilde{m}$ to determine the boundary values of γ.

Chapter 5

Conclusions



The multidisciplinary character of this work is a reflecion of several placements and collaborations I have been involved in during the time of my PhD studies. The four projects which I studied here span several fields such as networks, navigation algorithms, the fluctuations of packets' traffic, the analysis of social interactions via web pages and the analysis and modeling of the part of the UK national health service. Thus, in this concluding chapter I will sum up each of the above chapters separately, rather that comment on the whole thesis.

5.1 Navigation on networks

The goal of the first chapter was to investigate the possibility of using the average shortest delivery time property to navigate packets in a complex network. Moreover, my intention was to apply a local navigation algorithm,

which could benefit from a simple architecture and nearest neighbourhood knowledge. The latter points were motivated by my interest in designing an efficient algorithm, which could operate in a network which shape is unknown and global navigation is impossible. The conditions described above exclude application of widely used shortest path algorithm, where the optimal shortest path between nodes i and j is obtained based on full information of the network connectivity. Hence, I introduced a three variations of the average shortest path algorithm ST, STD and CDT (see Chapter 1). The performance of two latter ones was compared with deterministic version of the algorithm based only on the degree property (CD, Chapter 1).

The analysis of the ST algorithm showed two intrinsic problems related with the average shortest time approach. Firstly, the initially average shortest paths were not necessarly the best possible ones and the navigation algorithm lacks an update mechanism, which would inform nodes successively about them. Secondly, the jamming of hubs occurred even for very small input rate R due to long delivery times and the back and forth phenomenon described in Chapter 1. The introduction of $1/\Delta T_i$ and degree k_i properties in the STD and CDT algorithms mostly solved both problems, however not without costs in potential efficiency. This is because the degree property is responsible for the avoiding of large nodes, which actually leads to losing benefits of the scale free structure and hub nodes. On the other hand, the $1/\Delta T_i$ property helps nodes to explore new paths and make the STD and CDT algorithms sensitive to changes in the network structure. However, the rate of changes must be much slower that the updating procedure.

The degree property included in both STD and CDT algorithm makes it sensible to compare their properties with an algorithm which based only on the degree in the local navigation of packets in the network. Thus, I intro-

duced the CD algorithm, a deterministic version of the algorithm described in [46] and found the STD algorithm has 14% lower load and 14% shorter mean delivery time compared to the CD navigation algorithm (see Tab. 1.1 and 1.2 in Chapter 1). The practical implementation of the STD and CDT algorithms needs the feedback packets to update nodes with the delivery times, what would double the flow in the network. However, the feedback packets would know the return paths and in reality, they are widely used in the most popular internet protocol TCP as the confirmations of the correct transmitions.

Finally, the STD algorithm is a step forward compared with the CD algorithm, but still greatly outperformed by the shortest path algorithm. It would be interesting to replace the 1 depth neighbourhood search with a 2 or 3 depth search [62], but it seams that the crucial thing for further implementation of the average shortest time algorithm is solving the back and forth sending of packets between two localy most efficient nodes. It would require probably an implementation in a node a memory of trafficked packets, what would however heavily complicate the potential algorithm.

5.2 Scaling of fluctuations

In chapter 2 I used a model of a network traffic to study the scaling of fluctuations of time series of nodes and traffic flow along the links. The analysis was performed for two types of navigation rules and using two different network structures: a scale-free network and a random graph. While the occurrence of scaling of fluctuations on nodes attracted recently several scientists [79, 80, 81, 82, 83, 84, 85, 86], the scaling of fluctuations of flow along the links were not studied. This could be related with the fact that in

all cases only the random diffusion of packets was considered, which as we showed in Fig. 2.4a does not exhibit any scaling properties of fluctuations of flow along the links.

I compared the results for two navigation rules: a random diffusion of packets and the degree navigation algorithm (D), which is a probabilistic version of the CD algorithm studied in Chapter 1. For the random diffusion the scaling of fluctuations of traffic flow along the links is not present. Moreover, the relations $\sigma_{ij} \sim < f_{ij} >$ and $\sigma_i \sim h_i(k = 2)$ are on average identical (Fig. 2.4b), what suggests that dynamical properties of nodes with $k = 2$ and links are the same. This is due to both constituents having two inputs/outputs and the same probability of receiving a packet in the network structure.

In the case of the D navigation rule, the probability of posting a packet along a link is proportional to the degree of the node at the end of the link (Eq. 2.2). Thus, the links in a network are not equal any longer and the probability of packets' flow along given link depends on the local network structure and the network degree distribution. In this way, the D algorithm introduces necessary preference among links, which finally leads to the scaling of fluctuations of traffic flow along the links (Fig. 2.6). The necessity of preference among network constituents for obtaining scaling of fractionations was highlighted by the result obtained for the lattice structure and the random diffusion in Fig. 2.5.

The peculiar property of the D navigation rule is the emergence of two scaling regions found for the $\sigma_{ij}(< f_{ij} >)$ relation on the scale free graph. I found this outcome results from the sharp differences between links' probabilities of transporting packets in a scale-free network for the D algorithm. The links that direct to very large nodes (hubs) have a very small chance

to transport packets and it happens very rarely even for the large observation time windows. Hence, the dynamics on those links resembles a random deposition process for which $\sigma_{ij} \sim < f_{ij} >^{0.5}$ [79]. This indicates that the dynamics on a network is dynamically split into the dynamics related with the main navigation rule and random deposition. The scenario described here can be used to understand the emergence of two regimes of scalings for fluctuations of time series of nodes for a very efficient navigation rule [62], for which some nodes do not take part in traffic of packets and are only receivers of randomly addressed packets, following the random deposition.

5.3 Internet discussions

In chapter 3 I analysed two types of internet discussions: forums and news groups. While recent work on on-line discussions focused on the communities of users, I studied the properties of networks of messages, which are the outcomes of human on-line interactions. The messages posted to the discussions by users create scale-free, tree-like structures with power law degree distributions. The subject of the forum or news group is a root of the network and it is surrounded by branches representing one of the threads of the discussion. The nodes of these networks, which represent messages, are aging, i.e. the older a message, the smaller chance it has to receive an answer. The power law character of the degree distribution indicates that while most of the messages do not attract much attention, some of them are the sources of large activity, attracting many responses from other on-line users. The degree distributions have large γ exponents, ranging from $3.28 < \gamma < 5.62$, what is probably related with the aging character of nodes.

The study of correlations of user activity time series revealed significant

correlations for two time periods: 12 and 24 hours. Both of these were present for forums and news groups, but 24h correlations were much stronger for news groups and 12h for forums. The very strong 24h correlations for news groups indicate the existence of a stable user community coming back every day at the same time. The regularity of these groups are probably amplified by the specific character of student communities and similarities in students' daily routines. On the contrary, the publicly open forum attracted a much more diverse audience.

The forums and news groups are accessible through an internet browser and the layout of the threads displayed on the web page may influence observed properties. In particular, two types of internet discussions studied here have a different approach in displaying recent activities. In the case of the news groups, the list of presented threads is organised according to the time of creation. Only the creation of a new thread pushes down the older ones. On the other hand, on the main page of the forum the list of threads is dynamic and the thread which received the last message is placed on the top of the list. This mechanism allows even older threads to jump to the top of the main page and attract new responses. The difference in the web page layout is responsible for several differences between the forums and news groups. Firstly, the exponents of the degree distributions are lower for forums than for news groups, because the messages on the forum are aging slower. Secondly, the distance distribution $D(r)$ for the new groups has an exponential character. This shows that long discussions are not expected, because old threads are pushed down on the main page, what inevitably decrease their chance of receiving a new message. On the other hand, the distance distribution $D(r)$ for forums display power law character, disturbed however by the maximum distanced $r = 12$ allowed on the web page.

Finally, I showed that the rate of inserting new threads is related with the average length of the thread (Fig. 3.10). The faster users start new threads the shorter on average is the thread, what probably has an impact on the quality of the discussion. Thus, it is not surprising that on the far ends of the spectrum we have trading news group, where users are not interested in chatting and political news groups, where the long exchange of arguments is a norm.

5.4 Self-organised criticality at A&E Department

In chapter 4 I studied the behaviour of the A&E Department by analysing two metrics: the ratio between the discharges and arrivals in function of the department workload and the patitnts' length of stay distribution. I discussed that the growing character of the discharges / arrivals ratio in function of number of patients in the department is an evidence of a self-adjustment of the staff to the current department's workload. This-self adjustment property prompted me to discuss the application of a self-organised model to simulate the behaviour of the A&E department. In particular, I focused on the sand-pile model, which mirrors the behaviour of the department on the macro level of analysis.

A hospital in general and an A&E department in particular are the systems where a simple first in first out (FIFO) queue cannot be applied. The cases of patients admitted at the A&E department vary from a simple sore throat to life threatening conditions. Thus, the effective treatment queue there changes all the time, reflecting the patients conditions.

In my model the dynamics of changes of patients' position in the treat-

ment queue is mirrored by the reshufling of grains on the lattice, when a new grain enters the system. The position and size of the input space, described by two parameters m and s respectivelly, sets the proportion of short and long treatment times, what captures the proportion of simple and complex cases of patients arriving to the $A\&E$ departemnt. The model is a macro level analogy to the A&E department, however the characters of obtained discharges / arrivals ratio in function of number of grains on the lattice and the length of stay distribution are in a good agreement with the ones found for the $A\&E$ departemnt.

Appendices

Much of my work is based on computational methods and the algorithms I use in my simulations. They are vital products of my work, but never shown in any publication.

The subjects of the Appendix A is:

Obtaining the exponent from a power law distribution. A set of Matlab functions for calculation and visualisation.

Appendix A
Obtaining the exponent from a power law distribution

Our main goal was to develop a set of tools to obtain the exponent of a power law function or distribution. This kind of distribution is commonly observed in many real data sets. However obtaining the proper value of the exponent of such distributions is usually problematic.

The most important part of the power law distribution is the tail, which usually consists of a small number of samples. Other problems complicate the analysis, for example in some cases it is difficult to find data that spreads over a suitable number of orders of magnitude while other projects, for example a health study, have a limited number of observations. These effects make it more difficult to reliably obtain the exponent of the distribution, also called the *scaling exponent*.

In this appendix we do not discuss wheather a given distribution is a valid power law function and we assume the user of our Matlab functions is confident about the character of the data he analyses. However, if the user does not feel particularly bold about the true character of the distribution we suggest calculating goodnes-of-fit between the data and the power law using for instance Kolmogorov-Smirnov (KS) test [59, 60].

APPENDIX A

General Ideas

The power law function is given by an equation:

$$P(x) = Ax^{\alpha} \tag{A-1}$$

where α is the exponent and A is a constant. Some of the methods discussed here are related only to the case where $\alpha < 0$. This is the case for cumulative and rank distributions as well as the Newman method [58]. In this appendix we will mostly talk about distributions, however linear regression methods can be also applied to other power law relations.

We divided our techniques into two parts:

- Data preparation
- Analysing methods

We begin by creating histograms or probability distributions of the raw data, this might be the number of web pages based in various countries for example. We call raw data set of samples obtained from a measurement. The histogram/probability distributions are not usually the best sources for regression techniques - linear or nonlinear, because they are usually too noisy. However we can convert these histograms into both cumulative and rank distributions so as to make them smoother. For a power law probability distribution the relation between its exponent α and the exponent α' of the cumulative or rank distribution is $\alpha = \alpha' - 1$. Figure A-1 shows the reduction in the noise of a data set with both a cumulative or rank distribution applied. The data becomes very smooth so that regression methods will work much better than with the raw data. The Newman method was designed to work only with raw data, we do not need to build a distribution. However quite often the only data we have is a probability distribution or histogram, so we adapted the Newman method also for this situation.

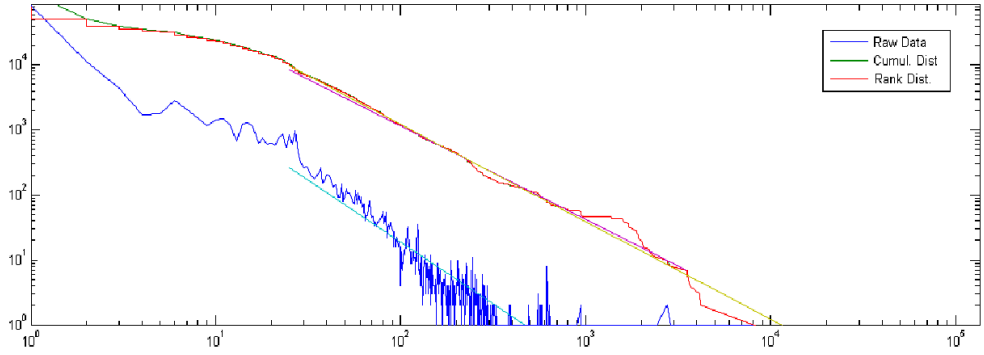


Figure A-1: A comparison between raw data ,cumulative and rank distributions.

Working with data

Obtaining the exponent of a power law distribution is one part of the process of analysing data, the shape and quality of the distribution affects the research methods that can be applied. Thus the decision of which method to use is mostly data dependent. Fig. A-3 explains the flow of processes from the raw data to the output (the scaling exponent α). The regression methods also give us the constant parameter A, so we can obtain complete information about the data. Fig. A-3 shows which methods are available at each step of the analysis. It was prepared with distributions in mind for for other power law relations (involving growing functions) the only possible flow of data is shown in Fig. A-2.

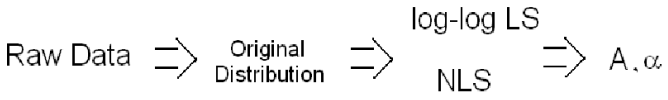


Figure A-2: The flow of data for growing functions.

According to Fig.A-3 there are three ways to proceed with the raw data.

The fastest and most straightforward way is the Newman method. Alternatively we can prepare the original distribution for example the degree distribution. This gives us an access to all of the available methods of analysing the power law distribution. Finally we can create a rank distribution, which can be used by the regression based methods. The i symbol in Fig.A-3 indicates that this process is not a standard procedure. If we have discreet original distributions, we can create its "raw data" and use the Newman method or make the rank distribution. However for real data (y has real values) linear regression methods, based on original or cumulative distributions, are also possible.

In the next sections we will describe the methodology to create cumulative and rank distributions. We will also discuss the regression methods in detail. The maximum likelihood estimator (Newman) method is well described in recent works [58, 59] and is not discussed in details here.

Data preparation methods

The raw data is a set of measurements or elements that are characterised by a given property. The good example is a network, where each node has its degree - or number of connections. In this example our set $X \subset (x_1, x_2 \dots x_N)$ consists of nodes degrees x_i and is the number of nodes N in the network.

From raw data we can make a histogram or probability distribution of a given property $p(x)$, for example the degree distribution. This distribution gives the probability of property x_i or how frequent a given property is.

The rank distribution is created from raw data, for each element or measurement we give it a rank. Rank 1 is given to largest element, rank 2 to the next one and so on. Then we plot $rank(x)$ to obtain the rank distribution. The rank distribution can be made from the original (frequency) distribution

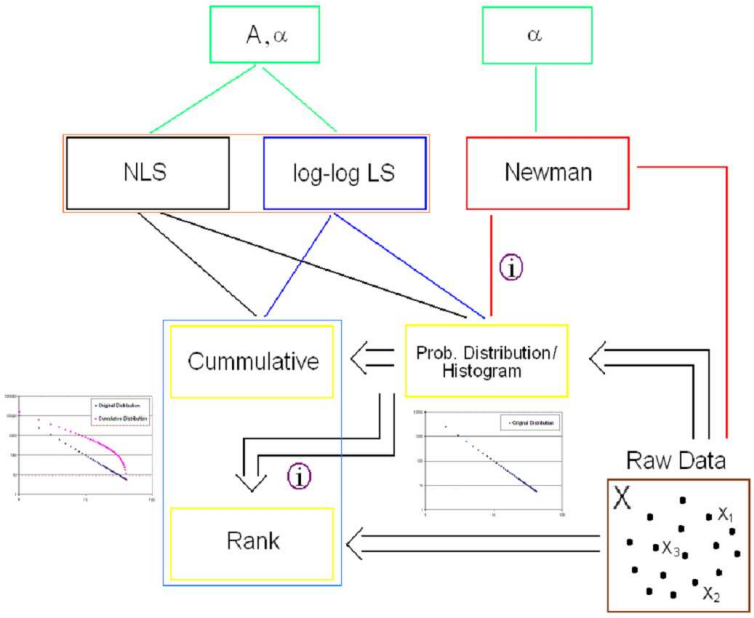


Figure A-3: Flow of data preparation methods, analysing techniques and possible outputs. The rank is connected with the cumulative one (blue box). The NLS and Log-Log LS are regression methods, the first is nonlinear and the second is linear operating on double logarithmic data. The light green boxes show the possible output from given methods. The i symbol indicates non-standard data processing.

if we have no access to the raw data. Because we know how many elements with a given property should be in the system we can create the 'raw data' and then create the rank distribution.

You can also extract this information from the probability distribution by multiplying each probability by $1/p_{min}$ and rounding it to the natural value. p_{min} is the smallest probability found in the distribution.

The cumulative distribution is given by:

$$P(x) = p(x' > x) = \int_x^\infty p(x')dx' \tag{A-2}$$

or in discrete form:

$$P(x) = p(x_i > x) = \sum_{i=j}^{N} p(x_j) \tag{A-3}$$

Therefore it is a sum of all probabilities (x') larger than x. The exponent of a cumulative distribution is larger by 1 than the exponent of the probability distribution ($\alpha' = \alpha + 1$). If we insert $p(x') = A(x')^{\alpha}$ and $\alpha \geq -1$, into the cumulative distribution the value of the integral will diverge.

However, the relation $\alpha' = \alpha + 1$ between the exponents of the probability distribution and the cumulative one holds only if the analysing probability distribution is a purely power law function. Especially, the measurements performed on small systems often lead to the finite-size effect in particular distributions obtained from these systems. In Fig. A-4 we show the probability and cumulative distributions of the nodes waiting times discussed in Chapter 2. In this particular case we observe an exponential cutoff related with the number of nodes in the network $N = 1000$. The finite-size effect affects the cumulative distribution function (CDF) severely and one should not perform any analysis of the scaling exponent based on it. In such a case, one should identify the source of the finite-size effect and related with it maximum value of x and then discard all data larger that x_{max} from the raw data set before obtaining of the scaling exponent α.

Analysis methods

Our two regression techniques NLS and Log-Log LS work with the least square method. The goal in the method is to minimise following equation:

$$S = \sum_{i=1}^{N} (y_i - f(x_i, \vec{a}))^2 \tag{A-4}$$

Or to minimise the square of the difference between our data $Y \subset (y_i)$

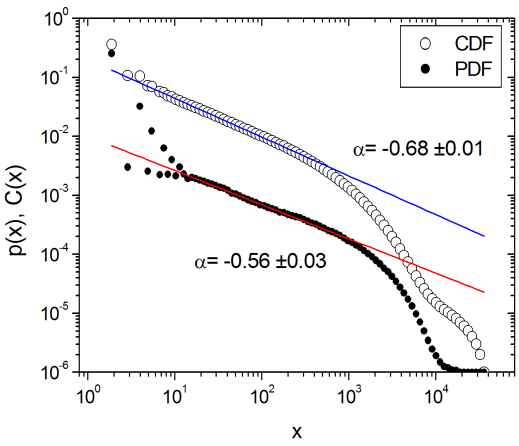


Figure A-4: The impact of the finite-size effect in the probability distribution function (PDF) on the cumulative distribution function (CDF). The exponents of PDF and CDF do not hold $\alpha' = \alpha + 1$ relation.

and the function f. Let's consider situation we have two parameters a_0 and a_1. To find the minimal value of S we calculate the derivate over a_0 and a_1:

$$\frac{\partial S}{\partial a_0} = 0 \tag{A-5}$$

and

$$\frac{\partial S}{\partial a_1} = 0 \tag{A-6}$$

As a result we have two equations from which we can calculate a_0 and a_1. This method works well with linear regressions where parameters a_0 and a_1 are described by the linear relationship $y(x) = a_0 + a_1 x$, if we have a set of (x, y) data. This method is also used in Log-Log LS regressions, where a double logarithmic transformation is used. If we consider a power law relation $y = Ax^{\alpha}$, this can be written as a linear function if we calculate the logarithm of both sides of this equation:

$$log(y) = log(Ax^{\alpha}) = log(A) + \alpha log(x) \tag{A-7}$$

which can be written as:

$$y' = A' + \alpha x' \tag{A-8}$$

where $y' = log(y)$, $A' = log(A)$ and $x' = log(x)$

The NLS method also uses the least square method but for nonlinear functions [61]. It is not straightforward, because the sum S is a form of averaging, which works well for linear functions. However, for nonlinear functions different parts of the curve should be calculated with different weights.

If we consider power law functions, most of the elements contribute to the left side of the curve, with small values of x. The tail of the distribution is neglected by the least square method. To deal with this problem for nonlinear functions the nonlinear least square technique was introduced. We do not calculate the parameters a_0 and a_1 directly, but in each step j we find the best approximation to them. We run this method till parameters converge to constant values or the level of improvement from one step to another drops to a sufficient level ε. Provided that our nonlinear function f has continuous second partial derivatives, we can write for it Taylor expansion for a given step j:

$$f(x_i)_{j+1} = f(x_i)_j + \frac{\partial f(x_i)_j}{\partial a_0}\Delta a_0 + \frac{\partial f(x_i)_j}{\partial a_1}\Delta a_1 + O(\Delta {a_0}^2) + O(\Delta {a_1}^2). \tag{A-9}$$

where $\Delta a_0 = a_{0,j+1} - a_{0,j}$ and $\Delta a_1 = a_{1,j+1} - a_{1,j}$. Then, based on a linear approximation to the components of f (a linear model of f, for instance the Gauss-Newton method) in the neighbourhood of x_i, for small Δa_0 and Δa_1 we see from the Taylor expansion A-9 that

$$f(x_i)_{j+1} = f(x_i)_j + \frac{\partial f(x_i)_j}{\partial a_0}\Delta a_0 + \frac{\partial f(x_i)_j}{\partial a_1}\Delta a_1 \tag{A-10}$$

As a result we can write the linear formula:

$$y_i - f(x_i)_j = \frac{\partial f(x_i)_j}{\partial a_0}\Delta a_0 + \frac{\partial f(x_i)_j}{\partial a_1}\Delta a_1 \tag{A-11}$$

or in matrix notation:

$$\{D_j\} = \{Z_j\}\{\Delta A\} \tag{A-12}$$

From this equation we can use the classic least square method to obtain a_0 and a_1. The crucial point of this method is to use suitable initial values of a_0 and a_1. The constant parameter a_0 should be of the order of the largest value in the data set (x or y) and the exponent a_1 should be positive for growing and negative for decaying functions. This method can be unstable and only for some datasets it only works only if the initial parameters are close to their solution. It is recommended that the Log-Log LS method is used first and then the NLS method is run with the results.

The Newman method [58, 59] is very simple method to obtain the exponent, without calculating the distribution. This is very useful in many cases, however there are a few problems with its accuracy. This method is sensitive to $xmin$ and $xmax$, which are the lower and upper bound of the scaling region. It is very important that $xmax$ is large so that the data range is over a few orders of magnitude. For data where between 1 to 1.5 orders of magnitude is available the Newman method calculates the exponent with a large error. The analysis of a theoretical distribution is shown in Fig. A-5. For the same dataset the whole data and only first 50 results were analysed. It shows that when a higher maximum value is used the calculation of the exponent is largely insensitive to the minimum of the data set. However as the range of the data set narrows the calculation of the exponent diverges.

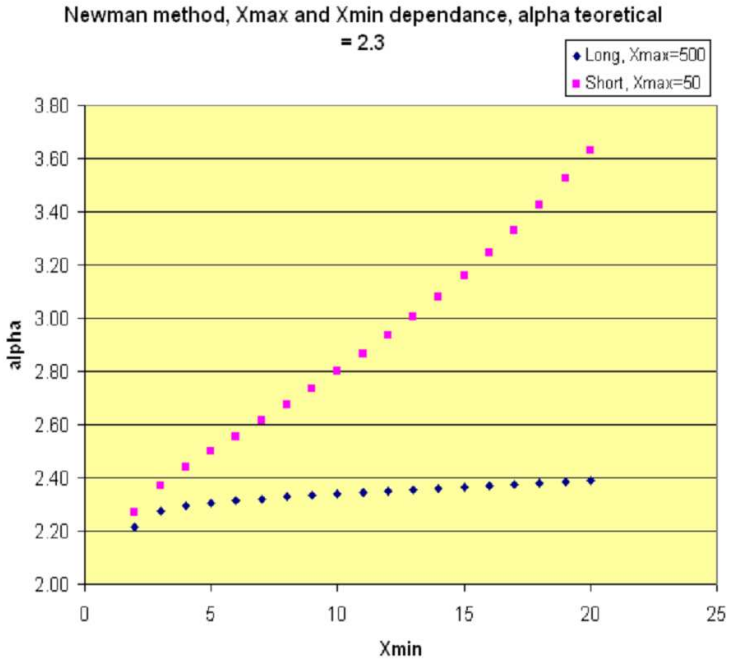


Figure A-5: The accuracy of the Newman method compared for long and short part of the same distribution.

Matlab functions

Data vector B and low level methods We wrote a set of Matlab functions that use all of the methods described above. Fig. A-6 explains the flow of data. On the lowest level we have the raw data - a vector $B = (x_1, x_2 \ldots x_N)$, where x_i is the measured value and N is the number of samples. Because Matlab treats a vector as a matrix, we need to make sure that the data is in a row or column order. The row is the standard in all functions the $B = B'$ operation can be used to transpose the data. This vector can be analysed using the Newman method using:

- a=newman(B,xmin);

or transformed into:

- rank distribution - R=RankDistNew(B);
- histogram or probability distribution - D=histogram(B);

The Newman method, rank distribution and probability distribution all work with real numbers. A histogram can be regarded as not normalised probability distribution. The output of the newman(B,xmin) function is 1×2 matrix, where $a(1)$ is the exponent and $a(2)$ is the error. The rank (R) and probability (D) distributions are analysed with other tools.

There are several parameters that are common to many functions:

- xmin - minimum value of x to include in the analysis;
- xmax - the maximum value, similar to xmin. In all functions xmax can be equal to 0, in this case functions takes largest value in vector B;
- a1,b1,e - the parameters for the NLS method. This method needs to start with some parameters. $a1$ is equivalent to the constant A, $b1$ is the exponent and e is the difference below which the iterations stop. The value of a1 should be around the largest value (or larger) in the data and $b1$ should be at least having the correct sign (growing, decaying). However sometimes quite precise values are needed. The better the initial parametrisation the faster the result is obtained. For unrealistic parameters the function will diverge;

Regression methods starting from a given distribution If we have a distribution created from the raw data or we have some given distribution we can use:

- both regression methods for the probability, cumulative and rank distributions;

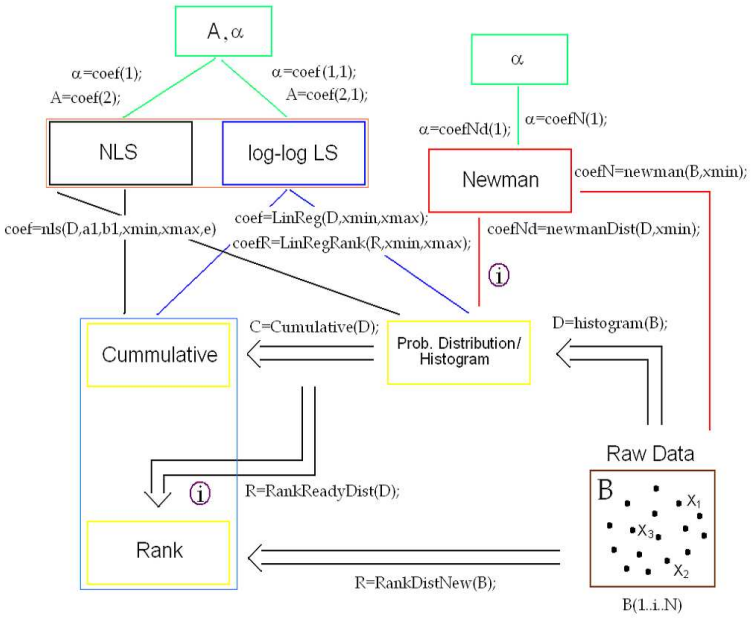


Figure A-6: The data flow with Matlab functions.

- the Newman method for the histogram;

The Newman method (a=newmanDist(D,xmin)) has been adapted to work with histograms. The D parameter is the input histogram. The output parameter is 1×2 matrix, where $a(1)$ is the exponent and $a(2)$ is the error.

The first linear regression function coef=LinReg(D,xmin,xmax) works with probability and cumulative distributions. It works with real numbers. The output parameter is a 2×2 matrix, where the coef(1,1) value is the exponent α, coef(1,2) the error and coef(2,1) the constant parameter A. The function has problems with rank distribution so a second function was written: LinRegRank(R,xmin,xmax).

The last nonlinear regression function (NLS method is run through the a=nls(D,a1,b1,xmin,xmax,e) function and works with probability, cumulative and rank distributions. It works with real numbers. The input parame-

ters are described in previous sections. This function works fine, however it gives poor results if the initial parameters $a1$ and $b1$ are not very well set. a1 is particularly crucial as the solution will diverge if it is too small. The output contains $a(1)$ as the exponent and $a(2)$ as the constant parameter A. This method does not give an error.

If we have an (x, y) relation instead of a histogram or we have some growing function, we should only use the regression methods. Quite often (x, y) relations do not have a functional form; there are the same x values with different y values. This of course will create problems, however the functions will work. For better accuracy we should calculate an average for each y value or at least sort our data in ascending/descending order.

Useful Functions

We created a set of useful functions to calculate the exponent using several methods in just one run. We also prepared functions for visualising the results.

- These functions go through all the necessary steps from raw data (vector B) to the output (the exponent and its error). detect(B,xmin,xmax) function compares the Newman method with Log-Log LsSmethod for histogram, cumulative and rank distributions. It gives four outputs with their errors (the param matrix). detect_nls(B,a1,b1,xmin,xmax) function calculates the same results using a NLS method. Because this method is more sensitive to the input parameters and does not give any errors, we separated it from the detect(B,xmin,xmax) function. largest(B) and largest2(D) find the largest value in $1 \times N$ and $2 \times N$ matrixes respectively. They are useful in estimating the value of $a1$. The find_exp_B(B,xmin,xmax) function calculates exactly the

same results as the detect(B,xmin,xmax) function but also draws the results: firstly the approximated power laws compared to the histogram and histogram compared to the cumulative and rank distributions.

  - param=detect(B,xmin,xmax);
  - param=detect_nls(B,a1,b1,xmin,xmax);
  - param=find_exp_B(B,xmin,xmax);
  - a1=largest(B);
  - a1=largest2(D);

- If we want to start from a given distribution/histogram or relation we can use the param=detect_dist(D,xmin,xmax) function. It uses the same methods as detect(B,xmin,xmax) except for the two marked on Fig. A-6 with the symbol *i*. The limitations of these methods are described in previous sections. We also created the B=dist_B(D) function, which converts a histogram into raw data (vector B).

  - param=detect_dist(D,xmin,xmax);
  - B=Dist_B(D);

- These functions draw one or two power laws with a given exponent and compares it with the data D. The output is a figure and the value of R^2. This value of course depends on the input parameter *xmin*. The last function draws 2 datasets with given α and *xmin* for each dataset.

  - draw(D,α,xmin);
  - draw2(D,α1,α2,xmin);
  - draw22(D1, α1,D2,α2,xmin1,xmin2);

- One of the initial goals of this project was to obtain the exponent of the degree distribution of a network. We built the find_exp(G,xmin,xmax) function, which works with an adjency matrix G. The output is a figure with approximated degree distributions (one for each method) and the matrix with the parameters: exponents and errors.

  – param=find_exp(G,xmin,xmax);

The functions shown here can be widely used to calculate a range of results. However, there are a lot of cases where we do not need to go through all the steps or run all the methods. In the next section we will show how to use the functions.

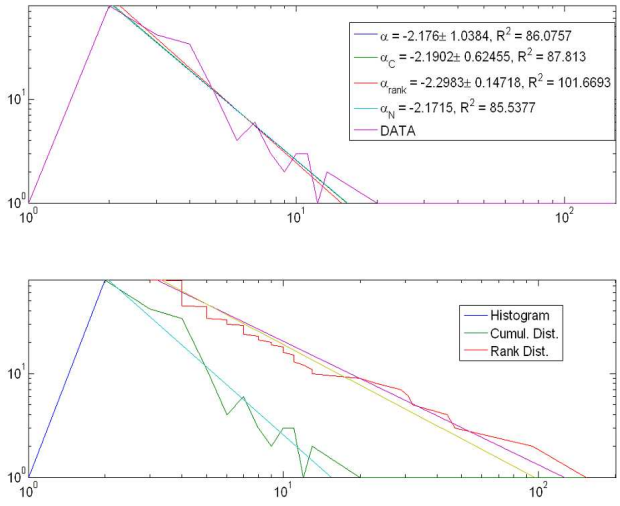


Figure A-7: The output figure from find_exp(),find_exp_B() and find_exp_D() functions.

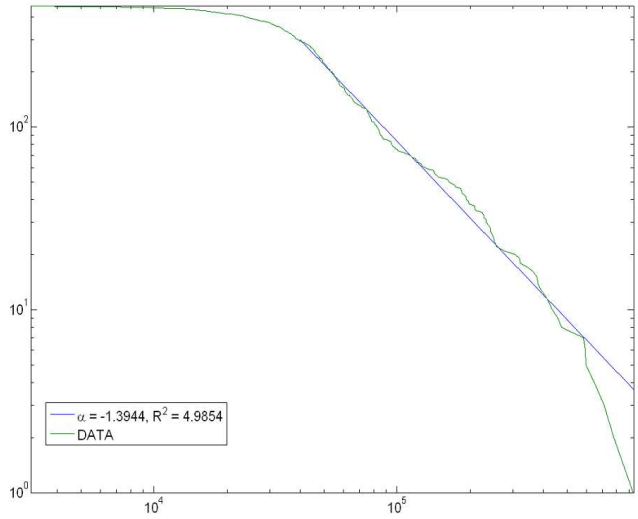


Figure A-8: The output figures from draw() function.

Examples

The functions were written to be used step by step and to allow new functions to be easily built. Here we give some examples of how to build them.

We start from reading the raw data from a file

- load 'data.txt';

if our data is a column it is transposed into a row

- data=data';

usually we need to estimate the *xmin* for our functions, for small and medium sized datasets we can run the rank distribution analysis very quickly

- R=RankDistNew(data);

For very large datasets running the histogram function is recommended

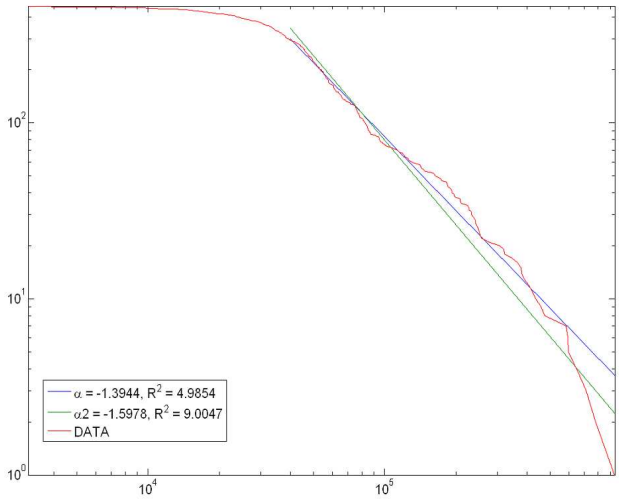


Figure A-9: The output figures from draw2() function.

- D=histogram(data);

We can plot this R or D matrix and estimate the *xmin* from which power law relation starts.

Having *xmin* we can run the detect(data,xmin,xmax) function. For most cases a zero value for *xmax* is recommended, because the function will use then use the largest value in the data.

- param=detect(data,xmin,0);

If we already have the histogram or distribution D, we can easily find *xmin* and run

- param=detect_dist(D,xmin,0);

This function works quite slowly (because of the internal use of the R=RankReadyDist(D) function) . You can create vector B (raw data) by using

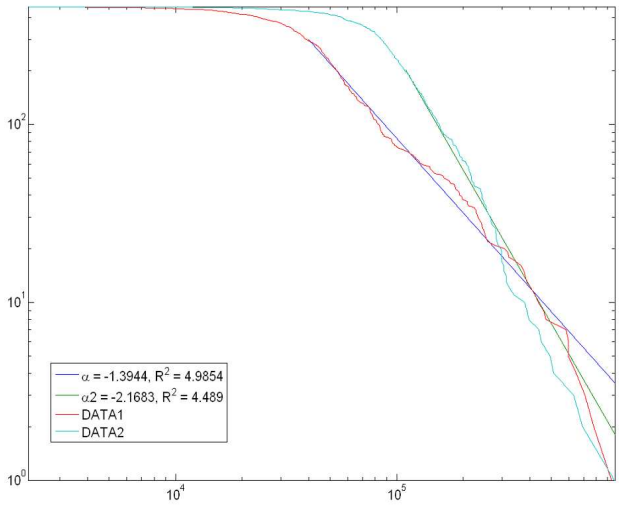


Figure A-10: The output figures from draw22() function.

- B=Dist_B(D);

When we have a histogram, where large y values exist creating a rank distribution or a vector B can take a long time. If this histogram is quite noisy we want to make it smoother. In this case it is much faster to create a cumulative distribution from a histogram and then run regression method. Below is the list of steps from histogram D to the graphical output.

- C=Cumulative(D);
- alfa=LinReg(C,xmin,xmax);
- alfa2=nls(C,largest2(C),alfa(1,1),xmin,xmax,e);

(the value alfa(1,1) from LinReg was used as an input parameter, function largest2 finds largest value in $2 \times N$ matrix)

- draw2(C,alfa(1,1),alfa2(1),xmin);

These steps are the most typical, however the user can look at the Fig. A-6 (the data flow diagram) and create their own functions.

List of functions

Below is a list with all of the functions written for this project. Some of them are just internal functions, not for end-users. First we describe some standard notation:

- B is the vector of raw data;
- D is a histogram;
- C is a cumulative distribution;
- R is a rank distribution
- *xmin* is the minimum value a function uses in a regression;
- *xmax* is the maximum value a function uses in a regression (0 is used for automatically obtaining the maximum value from the data);
- α is the exponent;
- param, a, anls, alfa, b, L, p are the outputs from some functions where the most important are described in Tab. A-1.

The outputs param and par come from higher-level functions, it means they use basic functions inside to calculate the exponents. The alfa, a and anls outputs come from basic low-level functions.

- Higher-level functions:
  - param=detect(B,xmin,xmax);

Method	**Exponent α**	**Error**	**Constant A**
Newman	param(1,1)	param(1,2)	-
Log-Log LS, histogram	param(2,1)	param(2,2)	-
Log-Log LS, cumulative	param(3,1)	param(3,2)	-
Log-Log LS, rank	param(4,1)	param(4,2)	-
NLS histogram	par(1,1)	-	par(1,2)
NLS cumulative	par(2,1)	-	par(2,2)
NLS rank	par(3,1)	-	par(3,2)
Log-Log LS (depending on input)	alfa(1,1)	alfa(1,2)	alfa(2,1)
Newman	a(1)	a(2)	-
NLS (depending on input)	anls(1)	-	anls(2)

Table A-1: The list of all output parameters and what they contain inside themselves.

  - param=detect_dist(D,xmin,xmax);
  - param=find_exp(G,xmin,xmax);
  - param=find_exp_B(B,xmin,xmax);
  - param=find_exp_D(D,xmin,xmax);
  - par=detect_nls(B,a1,b1,xmin,xmax,e);

- Low-level functions:

  - alfa=LinReg(D,xmin,xmax);
  - alfa=LinRegRank(R,xmin,xmax);
  - a=newman(B,xmin);
  - a=newmanDist(D,xmin);
  - anls=nls(D,a1,b1,xmin,xmax,e);

APPENDIX A

- Data preparation functions:
  - C=Cumulative(D);
  - D=histogram(B);
  - B=Dist_B(D);
  - R=RankDistNew(B);
  - R=RankReadydist(D);
- Drawing functions:
  - draw(D,α,xmin);
  - draw2(D,α1,α2,xmin);
  - draw22(D1,α1,D2,α2,xmin1,xmin2);
- Internal functions:
  - b=findB(D,xmin,xmax,α);
  - p=nls_core(D,a1,b1);
  - draw_slopes(coef,coef2,coefR,alfaN,D,C,R,xmin);
- Other functions:
  - D=PowerLaw(N, ,a);
  - G=scalefreeBernard(N,m);
  - G=scalefreeKertesz(N,m);
  - L=largest(B);
  - L=largest2(D);

To obtain a socecode pleas emial berni.kujawski@gmail.com

Bibliography

[1] P. Erdős and A. Rényi, *"On Random Graphs. I."*, Publ. Math. (Debrecen) **6**, 290 (1959).

[2] R. Albert, H. Jeong, and A.-L. Barabási, *Diameter of the World-Wide Web*, Nature **401**, 130 (1999).

[3] R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tompkins, E. Upfal, *The Web as a graph*, *Proceedings of the nineteenth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, 1 (2000).

[4] J. M. Kleinberg, R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins, *The Web as a Graph: Measurements, Models, and Methods*, *Proceedings of the 5th Annual International Conference, COCOON' 99, Tokyo, July 1999* (Springer-Verlag, Berlin), 1 (1999).

[5] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajalopagan, R. Stata, A. Tomkins, and J. Wiener, *Graph structure in the Web*, Comput. Netw. **33**, 309 (2000).

[6] L. A. Adamic and B. A. Huberman, *Power-Law Distribution of the World Wide Web*, Science **287**, 2115 (2000).

[7] L. A. Adamic, *The Small World Web*, *Proceedings of the Third European Conference, ECDL'99* (Springer-Verlag, Berlin), 443 (1999).

[8] M. Faloutsos, P. Faloutsos, and C. Faloutsos, *On power-law relationships of the Internet topology*, ACM SIGCOMM Comp. Comm. Rev. **29**, 251 (1999).

[9] R. Govindan and H. Tangmunarunkit, *Heuristics for Internet map discovery*, in *Proceedings of IEEE INFOCOM 2000*, Tel Aviv, Israel (IEEE, Piscataway, N.J.), **3**, 1371 (2000).

[10] S. Yook, H. Jeong, and A.-L. Barabási, *Modeling the Internet's large-scale topology*, Proc. Natl. Acad. Sci. **99**, 13382 (2002).

[11] R. Pastor-Satorras, A. Vázquez, and A. Vespignani, *Dynamical and Correlation Properties of the Internet*, Phys. Rev. Lett. **87**, 258701 (2001).

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

[13] M. E. J. Newman, S. H. Strogatz, and D. J. Watts, *Random graphs with arbitrary degree distributions and their applications*, Phys. Rev. E **64**, 026118 (2001).

[14] A.-L. Barabási and R. Albert, *Emergence of scaling in random networks*, Science **286**, 509 (1999).

[15] R. Albert and A.-L. Barabási, *Topology of Evolving Networks: Local Events and Universality*, Phys. Rev. Lett. **85**, 5234 (2000).

[16] L. A. N. Amaral, A. Scala, M. Barthelemy, and H. E. Stanley, *Classes of small-world networks*, Proc. Natl. Acad. Sci. **97**, 11149 (2000).

[17] M. E. J. Newman, *From the Cover: The structure of scientific collaboration networks*, Proc. Natl. Acad. Sci. **98**, 404 (2001).

[18] M. E. J. Newman, *Scientific collaboration networks. I. Network construction and fundamental results*, Phys. Rev. E **64**, 016131 (2001).

[19] M. E. J. Newman, *Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality*, Phys. Rev. E **64**, 016132 (2001).

[20] M. E. J. Newman, *Assortative Mixing in Networks*, Phys. Rev. Lett. **89**, 208701 (2002).

[21] M. E. J. Newman and J. Park, *Why social networks are different from other types of networks*, Phys. Rev. E **68**, 036122 (2003).

[22] A.-L. Barabási, H. Jeong, E. Ravasz, Z. Néda, A. Schubert, and T. Vicsek, *Evolution of the social network of scientific collaborations*, Physica A **311**, 590 (2002).

[23] F. Liljeros, C. R. Edling, L. A. N. Amaral, H. E. Stanley, and Y. Aberg, *The web of human sexual contacts*, Nature **411**, 907 (2001).

[24] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and A.-L. Barabási, *The large-scale organization of metabolic networks*, Nature (London) **407**, 651 (2000).

[25] H. Jeong, S. P. Mason, A.-L. Barabási and Z. N. Oltvai, *Lethality and centrality in protein networks*, Nature (London) **411**, 41 (2001).

[26] D. A. Fell and A. Wagner, *The small world of metabolism*, Nat. Biotechnol. **18**, 1121 (2000).

[27] W. Bachnik, S. Szymczyk, S. Leszczynski, R. Podsiadlo, E. Rymszewicz, L. Kurylo, D. Makowiec, and B. Bykowska, *Quantitive and Sociological Analysis of Blog Networks*, Acta Physica Polonica B **36**, 3179 (2005).

[28] K. Zhongbao and Z. Changshui, *Reply networks on a bulletin board system*, Phys. Rev. E **67**, 036117 (2003).

[29] K.-I. Goh, Y.-H. Eom, H. Jeong, B. Kahng and D. Kim, *Structure and evolution of online social relationships: Heterogeneity in unrestricted discussions*, Phys. Rev. E **73**, 066123 (2006).

[30] A. Capocci, V. D. P. Servedio, F. Colaiori, L. S. Buriol, D. Donato, S. Leonardi and G. Caldarelli, *Preferential attachment in the growth of social networks: The internet encyclopedia Wikipedia*, Phys. Rev. E **74**, 036116 (2006).

[31] S. Valverde, G. Theraulaz, J. Gautrais, V. Fourcassié, and R. V. Solé, *Self-Organization Patterns in Wasp and Open Source Communities*, IEEE Intelligent Systems **21**, 36 (2006).

[32] J. Abello, P. M. Pardalos, and M. G. C. Resende, *On maximum clique problems in very large graphs*, in *External Memory Algorithms*, edited by J. Abello and J. Vitter, DIMACS Series in Discrete Mathematics Theoretical Computer Science (American Mathematical Society), 119 (1999).

[33] W. Aiello, F. Chung, and L. Lu, *A random graph model for massive graphs* , *Proceedings of the 32nd ACM Symposium on the Theory of Computing* (ACM, New York), 171 (2000).

[34] J.-P. Onnela, J. Saramäki, J. Hyvönen, G. Szabó, D. Lazer, K. Kaski, J. Kertész, and A.-L. Barabási, *Structure and tie strengths in mobile communication networks* , Proc. Natl. Acad. Sci. **104**, 7332 (2007).

[35] J. Živković, B. Tadić, N. Wick, and S. Thurner, *Statistical Indicators of Collective Behavior and Functional Clusters in Gene Networks of Yeast*, Physical Journal B **50**, 255 (2006).

[36] M. Ludwig and P. Abell, *An Evolutionary Model of Social Networks*, Phys. Jurnal B **58**, 97 (2007).

[37] J. Travers and S. Milgram, Sociometry **32**, 425 (1969).

[38] H. D. Rozenfeld, L. K. Gallos, C. Song, H. A. Makse, *Fractal and Transfractal Scale-Free Networks*, arXiv:0808.2206v1 [physics.soc-ph], (2008).

[39] B. A. Huberman and L. A. Adamic, *Growth dynamics of the World-Wide Web*, Nature **401**, 131 (1999).

[40] L. A. Adamic, R. M. Lukose, A. R. Puniyani, and B. A. Huberman, *Search in power-law networks*, Phys. Rev. E **64**, 046135 (2001).

[41] B. Tadić and S. Thurner, *Information super-diffusion on structured networks*, Physica A **332**, 566 (2004).

[42] B. Tadić, S. Thurner, and G. J. Rodgers, *Traffic on complex networks: Towards understanding global statistical properties from microscopic density fluctuations*, Phys. Rev. E **69**, 036102 (2004).

[43] A. Arenas, A. Díaz-Guilera, and R. Guimerà, *Communication in Networks with Hierarchical Branching*, Phys. Rev. Lett. **86**, 3196 (2001).

[44] R. V. Sole and S. Valverde, *Information transfer and phase transitions in a model of internet traffic*, Physica A **289**, 595 (2001).

[45] V. Jacobson, M. J. Karels, *Congestion Avoidance and Control*, *Proceedings of SIGCOMM '88* (ACM, Standford, CA, 1988).

[46] G. Yan, T. Zhuo, B. Hu, Z.-Q. Fu, and B.-H. Wang, *Efficient routing on complex networks*, Phys. Rev. E **73**, 046108 (2006).

[47] C.-Y. Yin, B.-H. Wang, W.-X. Wang, T. Zhou, and H.-J. Yang, *Efficient routing on scale-free networks based on local information*, Phys. Lett. A **351**, 220 (2006).

[48] E. W. Dijkstra, *A note on two problems in connexion with graphs*, Numerische Mathematik, **1**, 269 (1959).

[49] R. Albert and A.-L. Barabási, *Statistical mechanics of complex networks*, Rev. Mod. Phys. **74**, 47 (2002).

[50] A.-L. Barabási, *The origin of bursts and heavy tails in human dynamics*, Nature **435**, 207 (2005).

[51] S. N. Dorogovtsev and J. F. F. Mendes, *Evolution of networks with aging of sites*, Phys. Rev. E **62**, 1842 (2000).

[52] S. N. Dorogovtsev and J. F. F. Mendes, *Scaling properties of scale-free evolving networks: Continuous approach*, Phys. Rev. E **63**, 056125 (2001).

[53] H. Zhu, X. Wang, and J.-Y. Zhu, *Effect of aging on network structure*, Phys. Rev. E **68**, 056121 (2003).

[54] J. Hołyst, A. Fronczak, and P. Fronczak, *Supremacy distribution in evolving networks*, Phys. Rev. E **70**, 046119 (2004).

[55] H. Jeong, Z. Néda and A.-L. Barabási, *Measuring preferential attachment in evolving networks*, Europhys. Lett. **61**, 567 (2003).

[56] K. B. Hajra and P. Sen, *Aging in citation networks*, Physica. A **346**, 44 (2005).

[57] H. Bauke and D. Sherrington, *Local attachment in networks under churn*, arXiv:0706.0018v1 [cond-mat.stat-mech] (2007).

[58] M. E. J. Newman, *Power laws, Pareto distributions and Zipf's law*, Contemporary Physics **46**, 323 (2005).

[59] A. Clauset, C. Rohilla Shalizi, and M. E. J. Newman, *Power law distributions in empirical data*, arXiv:0706.1062v1 [physics.data-an], (2007).

[60] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing* Cambridge University Press, Cambridge, England, 2nd edition, (1992).

[61] K. Madsen, H. B. Nielsen and O. Tingleff, *Methods For Non-Linear Least Squares Problems*, Informatics and Mathematical Modelling, Technical University of Denmark, 2nd Edition (2004).

[62] B. Tadić, G. J. Rodgers, and S. Thurner, *Treansport on Complex Networks: Flow, Jamming and Optimization*, Int. J. Bifurcation and Chaos **17**, 2363 (2007).

[63] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, *Complex networks: Structure and dynamics*, Phys. Rep. **424**, 175 (2006).

[64] M. Takayasu, K. Fukuda, and H. Takayasu, *Application of statistical physics to the Internet traffics*, Physica A **274**, 140 (1999).

[65] J. M. Kleinberg, *Navigation in a small world*, Nature **406**, 845 (2000).

[66] B. Tadić and G. J. Rodgers, *Packet Transport on Scale Free Networks*, Advances in Complex Systems **5**, 445 (2002).

[67] R. Guimerà, A. Arenas, A. Díaz-Guilera, and F. Giralt, *Dynamical properties of model communication networks*, Phys. Rev. E **66**, 026704 (2002).

[68] M. Rosvall and K. Sneppen, *Modeling Dynamics of Information Networks*, Phys. Rev. Lett. **91**, 178701 (2003).

[69] B. A. Huberman and L. A. Adamic, *Information Dynamics in the Networked World*, LNP: Complex Networks **650**, 371 (2004).

[70] P. Echenique, J. Gómez-Gardeñes, and Y. Moreno, *Improved routing strategies for Internet traffic delivery*, Phys. Rev. E **70**, 056105 2004.

[71] W.-X. Wang, B.-H. Wang, C.-Y. Yin, Y.-B. Xie, and T.Zhou, *Traffic dynamics based on local routing protocol on a scale-free network*, Phys. Rev. E **73**, 026111 (2006).

[72] R. Guimerà, A. Díaz-Guilera, F. Vega-Redondo, A. Cabrales, and A. Arenas, *Optimal Network Topologies for Local Search with Congestion*, Phys. Rev. Lett. **89**, 248701 (2002).

[73] M. Šuvakov and B. Tadić, *Transport processes on homogeneous planar graphs with scale-free loops*, Physica A **372**, 354 (2006).

[74] P. Holme, *Congestion and centrality in traffic flow on complex networks*, Advances in Complex Systems **6**, 163 (2003).

[75] P. Echenique, J. Gómez-Gardeñes, and Y. Moreno, *Dynamics of jamming transitions in complex networks*, Europhysics Letters **71**, 325 (2005).

[76] G. Mukherjee and S. S. Manna, *Phase transition in a directed traffic flow network*, Phys. Rev. E **71**, 066108 (2006).

[77] S. Valverde and R. V. Sole, *Internet's Critical Path Horizon*, Eur. Phys. J. B **38**, 245 (2004).

[78] W.-X. Wang, C.-Y. Yin, G. Yan, and B.-H. Wang, *Integrating local static and dynamic information for routing traffic*, Phys. Rev. E **74**, 016101 (2006).

[79] M. Argollo de Menezes and A.-L. Barabási, *Fluctuations in Network Dynamics*, Phys. Rev. Lett. **92**, 028701 (2004).

[80] M. Argollo de Menezes and A.-L. Barabási, *Separating Internal and External Dynamics of Complex Systems*, Phys. Rev. Lett. **93**, 068701 (2004).

[81] J. Duch and A. Arenas, *Scaling of Fluctuations in Traffic on Complex Networks*, Phys. Rev. Lett. **96**, 218702 (2006).

[82] B. Tadić, *Structure of Flow and Noise on Functional Scale-Free Networks*, Prog. Theor. Phys. Suppl. **162**, 112 (2006).

[83] Z. Eisler, J. Kertész, S.-H. Yook, and A.-L. Barabási, *Multiscaling and non-universality in fluctuations of driven complex systems*, Europhys. Lett. **69**, 664 (2005).

[84] Z. Eisler and J. Kertész, *Scaling theory of temporal correlations and size-dependent fluctuations in the traded value of stocks*, Phys. Rev. E **73**, 046109 (2006).

[85] Z. Eisler and J. Kertész, *Random walks on complex networks with inhomogeneous impact*, Phys. Rev. E, **71**, 057104 (2005).

[86] Z. Eisler, I. Bartos and J. Kertész, *Fluctuation scaling in complex systems: Taylor's law and beyond*, arXiv:0708.2053 (2007).

[87] S. N. Dorogovtsev and J. F. Mendes, *Evolution of Networks: From Biology to the Internet and WWW*, Oxford University Press (2003).

[88] P. Fronczak, A. Fronczak, and J. A. Hołyst, *Self-organized criticality and coevolution of network structure and dynamics*, Phys. Rev. E **73**, 046117 (2006).

[89] B. Tadić, *Adaptive Random Walks on the Class of Web Graphs*, European Physical Journal B **23**, 221 (2001).

[90] J. D. Noh and H. Rieger, *RandomWalks on Complex Networks*, Phys. Rev. Lett. **92**, 118701 (2004).

[91] A. Corral, *Long-Term Clustering, Scaling, and Universality in the Temporal Occurrence of Earthquakes*, Phys. Rev. Lett. **92**, 108501 (2004).

[92] A. Corral, *Mixing of rescaled data and Bayesian inference for earthquake recurrence times*, Nonlinear processes in Geophysics **12**, 89 (2005).

[93] A. Corral, *Universal Earthquake-Occurrence Jumps, Correlations with Time, and Anomalous Diffusion*, Phys. Rev. Lett, **97**, 178501 (2006).

[94] M. Boguna and A. Corral, *Long-Tailed Trapping Times and Lévy Flights in a Self-Organized Critical Granular System*, Phys. Rev. Lett. **78**, 4950 (1997).

[95] R. Sánchez, D. E. Newman, and B. A. Carreras, *Waiting-Time Statistics of Self-Organized-Criticality Systems*, Phys. Rev. Lett. **88**, 068302 (2002).

[96] L. Sabatelli, S. Keating, J. Dudley, and P. Richmond, *Waiting Time Distributions in Financial Markets*, European Physical Journal B **27**, 273 (2002).

[97] J. W. Lee, E. K. Lee, and P. A. Rikvold, *Waiting-Time Distribution for Korean Stock-Market Index KOSPI*, Journal of Korean Physical Society **48**, s123 2006.

[98] A. J. Bray and G. J. Rodgers, *Diffusion in a Sparsely Connected Space: a Model of Glassy Relaxation*, Phys. Rev. B **38**, 11461 (1988).

[99] C. Tsallis, *Possible Generalization of Boltzmann-Gibbs Statistics*, J. Stat. Phys. **52**, 479 (1988).

[100] K. Levenberg, *A Method for the Solution of Certain Non-Linear Problems in Least Squares*, The Quarterly of Applied Mathematics **2**, 164 (1944).

[101] S. N. Durlauf, *How can statistical mechanics contribute to social science?*, Proc. Natl. Acad. Sci. **96**, 10582 (1999).

[102] S. Galam, *Application of statistical physics to politics*, Physics A **274**, 132(1999).

[103] S. Battiston, E. Bonabeau, and G. Weisbuch, *Decision making dynamics in corporate boards*, Physics A **322**, 567 (2003).

[104] L. Telesca and M. Lovallo, *Are global terrorist attacks time-correlated?*, Physics A **362**, 480 (2006).

[105] D. Helbing and P. Molnár, *Social force model for pedestrian dynamics*, Phys. Rev. E **51**, 4282 (1995).

[106] J. G. Oliveira and A. L. Barabási, *Human dynamics: Darwin and Einstein correspondence patterns*, Nature **437**, 1251 (2005).

[107] M. Ausloos and R. Lambiotte, *Time-evolving distribution of time lags between commercial airline disasters*, Physics A **362**, 513 (2006).

[108] A. Hellervik and G. J. Rodgers, *A power law distribution in patients' lengths of stay in hospital*, Physics A **379**, 235 (2007).

[109] D. P. Smethurst and H. C. Williams, *Power laws: Are hospital waiting lists self-regulating?*, Nature **410**, 652 (2001).

[110] R. P. Freckleton and W. J. Sutherland, *Hospital waiting-lists (Communication arising): Do power laws imply self-regulation?*, Nature **413**, 382 (2001).

[111] D. Sornette, *Mechanism for Powerlaws witout Self-Organization*, International Journal of Modern Physics C **13**, 133 (2002).

[112] G. J. Rodgers, Y. J. Yap, and T. P. Young, *Simple Models of Waiting Lists*, Advances in Complex Systems **6**, 215 (2003).

[113] P. Bak, C. Tang, and K. Wiesenfeld, *Self-organized criticality*, Phys. Rev. A **38**, 364 (1988).

[114] P. Bak, K. Sneppen, *Punctuated equilibrium and criticality in a simple model of evolution*, Phys. Rev. Lett. **71** 4083, (1993).

[115] UK Department of Health, *The NHS Plan*, London, HMSO (2000).

[116] UK Department of Health, *The NHS Plan - a progress report*, London, HMSO (2003).