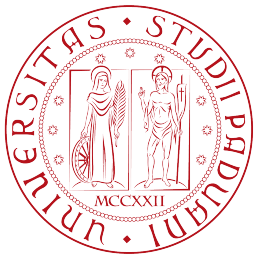


UNIVERSITÁ DEGLI STUDI DI PADOVA

Dipartimento di Fisica e Astronomia "Galileo Galilei"
Corso di Laurea Magistrale in Fisica

Tesi di Laurea

Controllability in Complex Brain Networks

Relatore
Prof. Samir Suweis

Laureanda
Federica Lorenzi

Anno Accademico 2018/2019

Abstract

Complex functional brain networks are large networks of brain regions and functional brain connections. Statistical characterizations of these networks aim to quantify global and local properties of brain activity with a small number of network measures. Recently it has been proposed to characterize brain networks in terms of their "controllability", drawing on concepts and methods of control theory. The thesis will review the control theory for networks and its application in neuroscience. In particular, the study will highlight important limitations and some warning and caveats in the brain controllability framework.

Contents

CHAPTER 1

Introduction

1.1 Networks Science

Every place we look, we see a world of amazing complexity [1]. The world offers a wide variety of complex instances in a number of areas: microbial communities, gene regulatory networks, bird flocks, ecosystems, cognition, political systems, stock markets, language, cities, economy, just to name a few. All these phenomena belong to the wide class of *"complex systems"*, characterized by the fact that their behavior cannot be understood simply from the knowledge of its constitutive parts. In order words, in complex systems the whole is more than the sum of its parts. [2] Even though these kinds of systems are very different from one another, the aim of physicists in Complexity Science is to find elegant and simple laws that can be used to describe and predict their emergent properties and collective behaviours.
The common feature of all complex systems is that they are generally composed of many elements that interact with each other in often complicated and nonlinear ways. Therefore, it is essential to find the right concepts that we can use to represent and analyze them: the use of "networks" and network theory has provided exceptionally powerful tools in this sense.

Practical examples of networks abound in our everyday life: the Internet (the network of computers physically connected between each other through optical fiber cables or telephone lines) and the World Wide Web (the network of information stored in web pages and linked by "hyperlinks") constitute

some of the most famous cases.

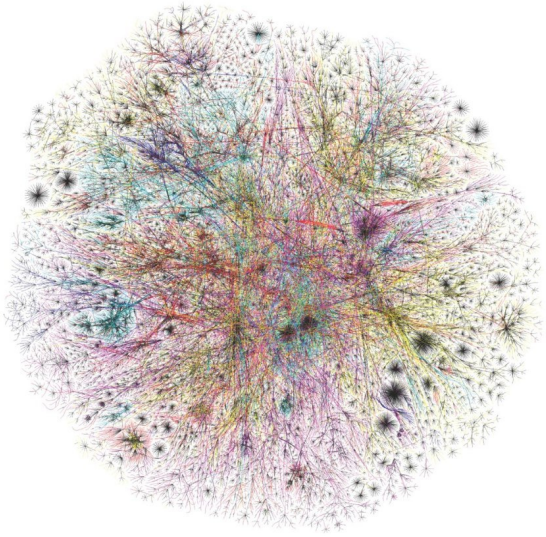
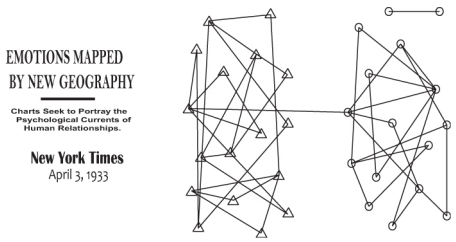


FIGURE 1.1 – A pictorical representation of the Internet network [3].

We can find instances of networks also in biological situations: ″food webs″, for example, are the network representation of species in an ecosystem linked by predator-prey interactions. Other relevant examples of biological networks are metabolic networks, protein-protein interaction networks and genetic regulatory networks. Networks find applications also in the scenario of social sciences: social, friendship and collaboration networks are often used to represent complex social systems in order to model interaction among people or groups.


EMOTIONS MAPPED
BY NEW GEOGRAPHY

Charts Seek to Portray the Psychological Currents of Human Relationships.

New York Times
April 3, 1933

FIGURE 1.2 – An example of one of the first social network analysis: the beginnings of social network analysis is credited to Jacob Moreno with the graphical social network of interactions between students. Here two groups of girls and boys are respectively represented. [4].

Since networks are a very general and abstract mathematical tool, they

can provide a language which can describe extremely different systems with a certain universality: the structure and the evolution of the networks behind each system is driven by a common set of fundamental laws and principles. Therefore, despite the amazing differences in form, size, nature, properties, and purpose of real networks, most of them are driven by common organizing principles. [2]

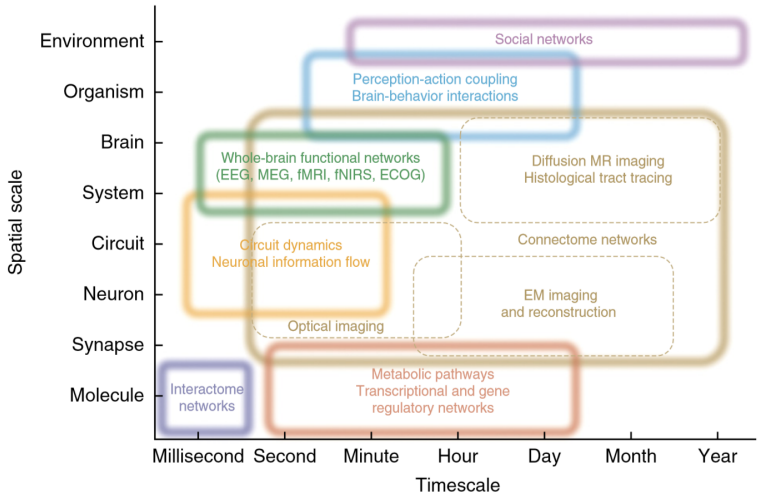


FIGURE 1.3 – Networks describe very different systems across many spatial and temporal scales [5].

Surely, this universality has contributed to the foundation of the new discipline of Network Science, but another force that has helped the growing interest in networks analysis is the recent access to large databases, which offer accurate maps of networks merging from different disciplines. All these factors have boosted the interest in the development of this interdisciplinary science.

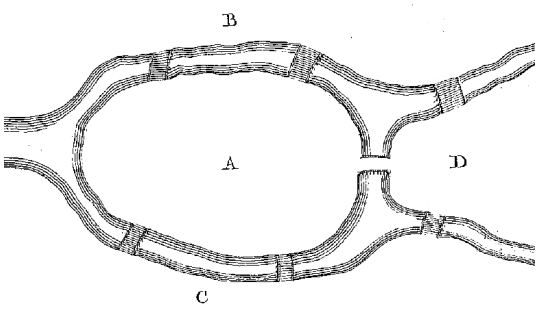
Even if Networks Science can be considered as a new science, it sinks its roots into preexisted disciplines. In fact, the formalism of Network Science is rooted in Graph Theory, and has borrowed conceptual frameworks to deal with randomness and seek universal organizing principles from Statistical Physics and Statistics, which can help extracting information from incomplete and noisy datasets.

In order to address the basic concepts related to networks, it is therefore useful to investigate the main topics developed from these disciplines, starting from Graph Theory.

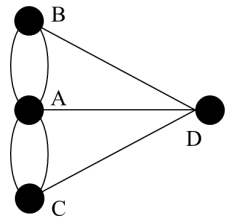
1.2 BASIC INTRODUCTION TO GRAPH THEORY

1.2.1 *Historical overview*

In 1736, Euler stated the basis of Graph Theory by solving a puzzle known as the *Königsberg seven-bridge problem*: Königsberg (now known as Kaliningrad) was an old city in Eastern Prussia lying on the Pregel River, which forms an island and then separates into two branches.



In this way, four land areas are created: the island A, two river banks B and C, and the land D between two branches. Seven bridges connect the four land areas of the city. The puzzle consists in finding a walk around the city which crosses each of the seven bridges just once. Begging pardon for spoiling, there is no solution to the puzzle, but there was no proof until 1736, when one of the leading mathematicians of that time, Leonhard Euler, published a proof that no such walk is possible. He not only dealt with this particular problem, but also gave a general method for other problems of the same type. Euler constructed a mathematical model for the problem in which each of the four lands is represented by a point and each of the seven bridges is represented by a line as follows:



Thus, the problem can now be reformulated by: *beginning at one of the points A, B, C and D, is it possible to draw the figure without traversing the same edge twice?* [6]
This mathematical model is known as a graph model of the problem. The points A, B, C and D are called *vertices*, the line segments are called *edges*, and

the whole diagram is called *graph*.

1.2.2 *Preliminaries and basic definitions*

As seen, a graph is essentially a way to code a relation between the elements of a system. A **graph** G is a tuple (V, E), where the N elements of the system identify the set of vertices, namely V, and the M relations among those the set E, i.e. the set of edges. Furthermore, the graph is said to have order N and size M respectively.
Each edge is an unordered pair of vertices. The two vertices associated with an edge e are called the *end-vertices* of e.
Whenever a real number can be attached to an existing edge we have that the edge is characterized by a weight w. In this case, the graph is said to be a *weighted graph*.

Hence, we also denote the set of vertices of a graph G by $V(G)$ and the set of edges of G by $E(G)$.
A **subgraph** of G is a graph H such that $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$, with the assignment of endpoints to edges in H which is the same as in G. It is used to write $H \subseteq G$.
A **spanning subgraph** of a graph G is a subgraph obtained by edge deletions only, in other words, a subgraph whose vertex set is the entire vertex set of G.

Denoting by $e = (u, v)$ an edge between two vertices u and v, then the two vertices u and v are said to be *adjacent* or *neighbour* in G and the edge e is said to be *incident* to the vertices u and v. Similarly, two edges e and f are said *adjacent* if they have an end in common.

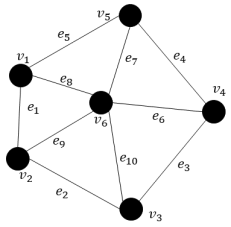
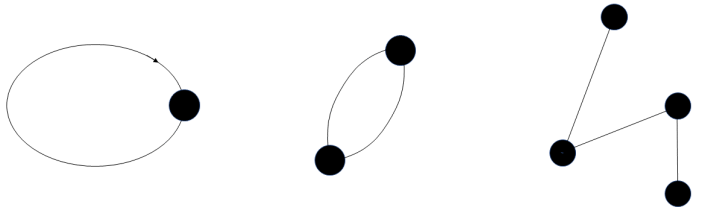


FIGURE 1.4 – This graph has six vertices ($v_1, v_2, v_3, v_4, v_5 and v_6$), and ten edges. Vertices v_1 and v_2 are the end vertices of edge e_1, thus they are adjacent. Vertices v_2, v_5 and v_6 are the neighbors of the vertex v_1.

A **loop** is an edge whose endpoints are coincident, i.e. $e = (u, u)$. **Multiple edges** are the ones having the same pair of endpoints, such as $e_1 = (u, v)$ and $e_2 = (u, v)$.

When in a graph, there are loops neither multiple edges, the graph is said to be **simple**. Conversely, a **multigraph** is characterized by the occurrence of multiple edges.



a) a loop b) a multiedge c) a simple graph

Ordered sequences of unique edges and intermediate nodes are called **paths**. Less formally we can say that a series of consecutive edges forms a path. The number of edges in a path is called the *length* of the path.
Whilst sequences of non unique edges are called **walks**. A graph is **connected** if each pair of vertices in G belongs to a walk.

If all the vertices of G are pairwise adjacent, then G is **complete**. A complete graph on N vertices is denoted by K_N. It is easy to note that a complete graph has $M = N(N-1)/2$ edges.

In contrast, a set of vertices is **independent** if each pair of its elements is not constituted of adjacent nodes. A graph G is called a **bipartite graph** if the vertex set V of G can be partitioned into two disjoint nonempty sets V_1 and V_2 , both of which are independent. If for each vertex $u \in V_1$ and each vertex $v \in V_2$, there is an edge (u, v) in G, one has a **complete bipartite graph**, and it is denoted by $K_{m,n}$ (whit the two partitioned sets containing m and n vertices, respectively).

In general, considering vertex degree distributions is an important technique in graph analysis. The **degree of a vertex** v, denoted by $k(v)$, is the number of edges incident to v in G. Thus, the degree distribution shows how many vertices have degree $0, 1, 2, ...$ and so on.
If all the vertices of a graph G have equal degrees, then we call G a **k-regular graph**, where k is the common degree.

There is an important further classification of graph, concerning the direction of the edges.
If the lines of a graph have a direction assigned to them, we have what is known as a **directed graph** or **digraph**.
A digraph consists of a set of points $V(D)$ and a set of ordered pairs of points $E(D)$, plus two functions $I(E \to V)$ and $F(E \to V)$. The first one assigns to

every edge e an initial vertex $I(e)$. The second one assigns to every edge e a final vertex $F(e)$.

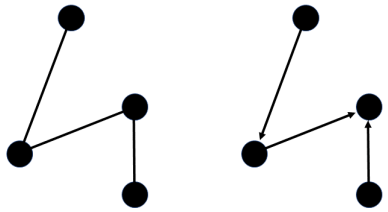


FIGURE 1.5 – An undirected graph and a digraph having the same vertices set.

More simply, every edge e has assigned a direction from one vertex $I(e)$ to another $F(e)$. Sometimes $I(e)$ and $F(e)$ coincide: in this case e forms a *loop*. Given a digraph D, its **underlying graph** $G(D)$ is obtained by replacing each directed edge with its undirected counterpart. Conversely, we can transform any undirected graph G into a directed one, $D(G)$, by associating a direction with each edge. Such a digraph is also known as an **orientation**.

The number of edges having v as their second point is called the indegree of v and is denoted by $k_{-}(v)$ or $k_{in}(v)$. Similarly, the outdegree of point is the number of lines having v has their first point and is written $k_{+}(v)$ or $k_{out}(v)$.

THEOREM 1.1 – For any directed graph D the sum of indegrees as well as the sum of outdegrees is equal to the total number of edges:

$$\sum_{v \in V(D)} k_{in}(v) = \sum_{v \in V(D)} k_{out}(v) = |E(D)|$$

The concept of indegree and outdegree can sometimes play a surprisingly important role when devising or analyzing real-world networks. To give an example, suppose we are devising a communication network in which we model the case that node u can send a message directly to node v by means of an edge $u \to v$. The indegree of node v may then indicate how many messages v can expect per time unit, also known as the rate of incoming messages. [7]

1.2.2.1 Tools for the characterization of networks

It is useful to investigate which tools have been developed to describe and represent graphs and their connectivity.

Let G be a graph with the vertex set $V = \{v_1, v_2, ..., v_n\}$ and the edge set $E = \{e_1, e_2, ..., e_m\}$.
The distance between two nodes is the length of the shortest path linking the nodes and is often of particular interest. All pairwise distances in a graph may be represented in the distance matrix: the *distance matrix* $\mathbf{D}(G)$ contains the element d_{ij} which represents the distance between vertices v_i and v_j, i.e. the number of edges belonging to the path.

The *incidence matrix* $\mathbf{I}(G)$ of G is an $n \times m$ matrix whose entry m_{ij} is 1 if the vertex v_i is adjacent to the edge e_j , and 0 otherwise.

The *adjacency matrix* $\mathbf{A}(G)$ of G is an $n \times n$ symmetric matrix, whose entry a_{ij} is the weights of the edge between the two vertices v_i and v_j .
The adjacency matrix can have even binary entries:

$$a_{ij} = \begin{cases} 1 & \text{if the edge } j \to i \text{ occurs} \\ 0 & \text{otherwise} \end{cases}$$

In other words, the adjacency matrix defines the topology of the graph by representing nodes as matrix rows and columns and representing edges as binary or weighted matrix entries.
For simple graph, all the diagonal elements, representing the loop edges, are zeroes. Furthermore, the adjacency matrix is symmetric for undirected graph.

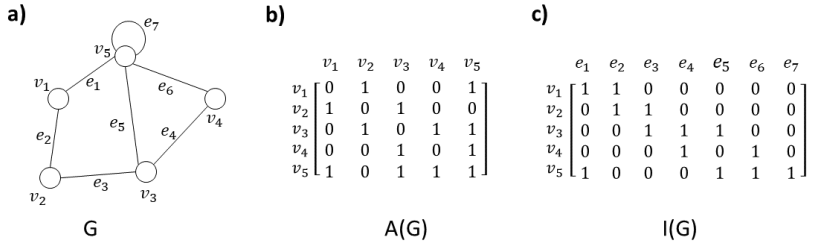


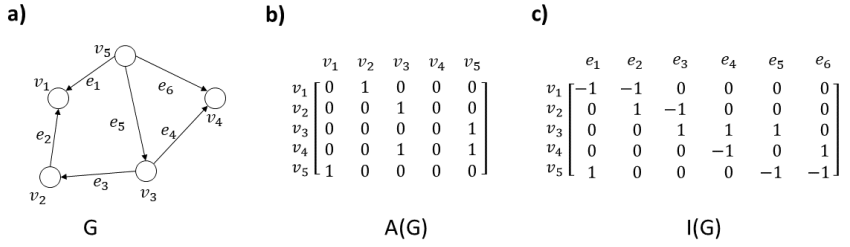
FIGURE 1.6 – (a) A graph G, (b) adjacency matrix representation of G, and (c) incidence matrix representation of G.

The adjacency matrix allows the derivation of the degree: the in-degree can be obtained by summing the adjacency matrix over rows, while the out-degree over columns.
In other words, the sum of values in row i is equal to the indegree of vertex v_i, that is, $\sum_j a_{ij} = k_{in}(v_i)$ and similarly for the outdegree $\sum_j a_{ji} = k_{out}(v_i)$.

In order to generalize these tools for directed graphs, the incidence matrix can be redefined as:

$$m_{ij} = \begin{cases} 1 & \text{if the edge } e_j \text{ starts from vertex } v_i \\ -1 & \text{if the edge } e_j \text{ ends from vertex } v_i \\ 0 & \text{otherwise} \end{cases}$$

Obviously, the adjacency matrix is still not symmetric for digraphs.



$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{ccccc} v_1 & v_2 & v_3 & v_4 & v_5 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array} \qquad \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{cccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{array} \\ \begin{bmatrix} -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \\ 1 & 0 & 0 & 0 & -1 & -1 \end{bmatrix} \end{array}$$

FIGURE 1.7 – (a) A digraph G, (b) adjacency matrix representation of G, and (c) incidence matrix representation of G.

1.3 CONTROLLABILITY FRAMEWORK

Recently, network science has been enhanced using concepts borrowed from engineering, like control and information theory, allowing us to understand the control principles of networks.
Indeed, there is a growing interest in understanding whether and how the design principles of many complex systems, even characterized by self organization, are determined by the need to control their behaviour.
Thus, the aim is to characterize the networks in terms of their *controllability* and to understand which is the impact of topology on the capacity of control the system.
First of all, it is essential to understand what we mean by ˝controllability˝.
In very general terms, controllability deals with the possibility to control a dynamical system via external inputs. In other words, it concerns with the capacity to influence the behaviour of a dynamical system with appropriately chosen inputs so that the system's output follows a desired trajectory of final state.

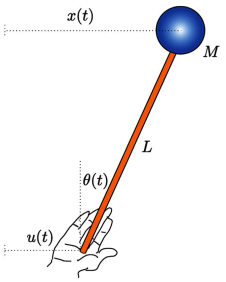
Hence, given a system with N elements, any state of the system is described at time t by a state vector $\mathbf{x}(t) \in \mathbb{R}^N$, whose entries $x_i(t)$ are called *state variable*. For a nonlinear system, the equations tuning the dynamics are the *state equation* and the *ouput equation* respectively:

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t); \Theta) \tag{1.1}$$

$$\mathbf{y}(t) = \mathbf{h}(t, \mathbf{x}(t), \mathbf{u}(t); \Theta) \tag{1.2}$$

where $\mathbf{u}(t) \in \mathbb{R}^M$ represent the M external inputs, the output vector $\mathbf{y}(t) \in \mathbb{R}^k$ describe the experimental measurable variable and Θ involves all the system parameters.

However, many nonlinear systems can be linearized around an equilibrium state. A simple mechanical instance is the *inverted pendulum*: our goal is to balance, and therefore to control, the stick in the upright position using the horizontal position of the hand as the control input $u(t)$.
Consider a stick of length L whose mass M is concentrated at the top. Denote the angle between the stick and the vertical direction with $\theta(t)$. The hand and the top of the stick have horizontal displacement $u(t)$ and $x(t)$,



respectively. [8]

The nonlinear equation of motion for this system is

$$L\ddot{\theta}(t) = gsin\theta(t) - \ddot{u}(t)cos\theta(t)$$

where g is the gravitational and

$$x(t) = u(t) + Lsin\theta(t)$$

When the stick is nearly at rest in the upright vertical position ($\theta = 0$, which is an equilibrium point), θ is small; hence we can linearize the equation of motion

$$\ddot{x}(t) = \frac{L}{g}(x(t) - u(t))$$

According to the notation used in 1.1, the state vector is $\mathbf{x} = (x(t), v(t))^T$ with velocity $v(t) = \dot{x}(t)$, and assuming $y(t) = x(t)$, it is possible to rewrite the state and output equations as the system:

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 1 \\ g/L & 0 \end{bmatrix}\mathbf{x}(t) + \begin{bmatrix} 0 \\ -g/L \end{bmatrix}u(t)$$

$$y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix}\mathbf{x}(t)$$

Generalizing this example, the class of time invariant linear (LTI) systems represents an interesting case. An LTI system describes systems with linear, or linearized, dynamics, and for them the 1.1 and 1.2 can be rewritten as:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \tag{1.3}$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \tag{1.4}$$

where $\mathbf{A} \in \mathbb{R}^{N\times N}$ is the *state matrix* which captures the interaction among the components of the system, $\mathbf{B} \in \mathbb{R}^{N\times M}$ is the *input matrix* which tunes the effect of the external inputs, $\mathbf{C} \in \mathbb{R}^{k\times N}$ is the *output matrix* and finally, $\mathbf{D} \in \mathbb{R}^{k\times M}$ is the *feedforward matrix*.

One of the most critical issue, previously cited, with the complete knowledge of the dynamical behaviour deals with the non linearity which is typical in a number of real systems. Most of the developed tools of control theory on networks concern with LTI systems.
Even if investigating LTI systems is surely a limitation, the linear approach is a fundamental first approximation: while many complex systems are characterized by nonlinear interactions between their components, the first step in

any control challenge is to establish the locally controllability of linearized system. Furthermore, the nontrivial network topology of real-world complex systems brings a new layer of complexity to controllability. Before we can explore the fully nonlinear dynamical setting, which is mathematically much harder, it is useful to investigate the impact of the topological characteristics on linear controllability, serving as a prerequisite for the nonlinear one. [9]

Dealing with brain networks, which are characterized by deep non linearity, we will show that LTI is compatible with linearization around the quiescent state, i.e. very low activity.
Hence, the differential equation, describing the evolution of the state is the

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \tag{1.5}$$

where the state variable $x_i(t)$ describes the neural activity; $\mathbf{A}$ is a matrix depending on the network anatomical connectivity and incorporating the effective interactions in the dynamics between the states, whose entries a_{ij} provide whether the neural element j affects the element i; while the matrix $\mathbf{B}$ provides which brain regions are affected by external inputs and the strength of that stimulation.
According to Kalman definition, a system is *controllable* if it can be driven from any initial state to any desired state in finite time via a suitable choice of $\mathbf{u}(t)$.
In mathematical terms, there exists a useful test to predict whether an LTI is controllable: the *Kalman's criterion of controllability*. [10]
Defined the $N \times NM$ *control matrix* C as:

$$C = [B, AB, A^2B, ..., A^{N-1}B] \tag{1.6}$$

the system is controllable if C has full rank, i.e. $rankC = N$
Dealing with the inverted pendulum

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ g/L & 0 \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} 0 \\ -g/L \end{bmatrix}$$

$$C = [B, AB] = \begin{bmatrix} 0 & -g/L \\ -g/L & 0 \end{bmatrix}$$

Hence, C has full rank and the inverted pendulum is controllable, according to our experience that we can balance a stick on our palm.

To understand the origin of 1.6, the formal solution of the state equation is:

$$\mathbf{x}(t) \int_0^t e^{[\mathbf{A}(t-\tau)]}\mathbf{B}\mathbf{u}(t)d\tau$$

If $e^{[\mathbf{A}(t-\tau)]}$ is expanded in series, it is easy to see that $\mathbf{x}(t)$ is actually a linear combination of the columns in the matrices $B, AB, A^2B, \ldots$. Note that for any $N^* \geq N$, we have $rank[B, AB, A^2B, \ldots, A^{N^*-1}B] = rankC$. So if $rankC < N$, then even the infinite series of $B, AB, A^2B, \ldots$ will not contain a full basis to span the entire N-dimensional state space. In other words, we cannot fully explore the state space, regardless of $\mathbf{u}(t)$, indicating that given our inputs the system is stuck in a particular subspace, unable to reach an arbitrary point in the state space. If, however, $rankC = N$, then we can find an appropriate input vector $\mathbf{u}(t)$ to steer the system from $\mathbf{x}(0)$ to an arbitrary $\mathbf{x}(t)$. Hence, the system is controllable.

1.3.0.1 Structural Controllability

Notwithstanding treating LTI systems is mathematically easier, trying to infer dynamical predictions from the state and output equations is a hard problem: it is nearly impossible to be aware of all the system's parameters, not to mention that it is needed an accurate wiring diagram and description of the system.

Fortunately, many control problems can be addressed even without the simultaneous knowledge of these features thanks to the emerging field of network science, as it will be shown.

Kalman's criterion is useful when we are treating small networks, but, it is impracticable for many networks.
First, because it can be actually hard to know all the entries of the system matrix and, on the other hand, computing the rank of C can be extremely laborious.
On the other hand, knowing only whether there is a link or not is a weaker information that is often available.
Exploiting this kind of information, *structural control* offers a theoretical framework to avoid issues related to Kalman's approach.
An LTI system $(\mathbf{A}, \mathbf{B})$ is a *structured system* if $\mathbf{A}$ and $\mathbf{B}$ are either fixed zeros or independent free parameters.
Thus, the system is said to be *structurally controllable* if we can set the nonzero elements in $\mathbf{A}$ and $\mathbf{B}$ such that the resulting system is controllable for almost all possible parameters realization.
In fact, it may occur that combinations of nonzero parameters are such that the system is not controllable when these parameters have values in some proper algebraic variety in the parameter space.
On the other hand, a system is *strongly structurally controllable* if it remains controllable for any value of the nonzero elements.
In other words, according to strong structural controllability, a system is

(strongly) controllable if there is no combination of nonzero link weights that violates Kalman's criterion.

Structural control provides a tool to infer if a system is controllable even if the values of the interactions are not known but only the map of the wiring diagram, i.e. the only required information is the knowledge of which elements are interacting with each other.

1.3.0.2 Graphical Interpretation of structural controllability

As previously said, according to structural controllability, the topology of the network, determined by the non zero entries of the matrices, is crucial.
Given an LTI system $(\mathbf{A}, \mathbf{B})$, it can be represented as a *directed graph*, or digraph, $G(\mathbf{A}, \mathbf{B}) = (V, E)$ where $V = V_S \cup V_I$ is composed both by the state vertices $V_S = (x_1 ... x_n) = (v_1 ... v_N)$, corresponding to the N nodes in the network, and the input vertices $V_I = (u_1 ... u_M) = (v_{N+1} ... v_{N+M})$, representing the M inputs; while E involves all the non zero element of the structure matrices $\mathbf{A}$ and $\mathbf{B}$: more precisely $E = E_S \cup E_I$ with $\{E_S = (x_j, x_i) | a_{ij} \neq 0\}$ and $\{E_I = (u_j, x_i) | b_{ij} \neq 0\}$. Thus, the condition $a_{ij} \neq 0$ implies that exists a link $j \rightarrow i$ in the network, which means that the node-j affects the node-i.
The M input vertices are called **origins** of the digraph, while the state vertices directly connected to the origins are known as **controlled nodes** or **actuators**. It may occur that the number of the controlled nodes is greater than the origins, since a origin can be linked with more state vertices, i.e. the cardinality of the actuators is determined by the number of non zero elements of $\mathbf{B}$. Therefore, the controlled vertices which don't share the origin are called **driver nodes**, whose number is N_D.
Obviously, any system is fully controllable if each node is individually controlled, i.e. $M = N$, however the aim is to identify the minimum number of input, and therefore driver nodes, to control the whole system.

In Lin's work [11], a graphical interpretation of structural controllability is provided. In particular, it is shown that *a system is not structurally controllable if and only if occur dilations either isolated nodes*.
A state vertex is **inaccessible** if there are no directed paths reaching it from an input vertices.
Given a digraph, it contains a **dilation** if there is a subset of state nodes $S \subset V$ such that the neighborhood set $T(S)$ of S, i.e. the set of vertices v_i for which there is a directed edge from v_i to some other vertex in S, has fewer nodes than S itself.
Roughly speaking, dilations are subgraphs in which a small subset of nodes attempts to rule a larger subset of nodes. In other words, there are more ″subordinates″than ″superiors″. While an inaccessible node cannot be in-

fluenced by input signals applied to the driver nodes. Consequently, the occurrence of one of the two conditions makes the whole network uncontrollable.

Furthermore, Lin provided an alternative graph-theoretical formulation. Preliminarely, a general graph is covered or spanned by a subgraph if the subgraph and the graph have the same vertex set. For a digraph, a sequence of oriented edges $v_1 \to v_2 \to v_{k-1} \to v_k$, where the vertices $v_1, v_2..v_k$ are distinct, is called an elementary path. While, when v_k coincides with v_1, the sequence of edges is called an elementary cycle.

Thus, the following subgraphs are defined:[9]

- A **stem** C is an elementary path originating from an input vertex;
- a **bud** O is an elementary cycle C with an additional edge e that ends, but does not begin, in a vertex of the cycle;
- a **cactus** is defined recursively: a stem is a cactus. Let C, O, and e be, respectively, a cactus, an elementary cycle that is disjoint with C, and a directed edge that connects C to O in $G(\mathbf{A};\mathbf{B})$. Then $C \cup \{e\} \cup O$ is also a cactus.

Dealing with structural controllability, a cactus is a minimal structure that contains neither inaccessible nodes nor dilations. Thus, an LTI system $(\mathbf{A};\mathbf{B})$ is structurally controllable if and only if $G(\mathbf{A};\mathbf{B})$ is spanned by cacti, i.e. if there exists a set of disjoint cacti that cover all state vertices.

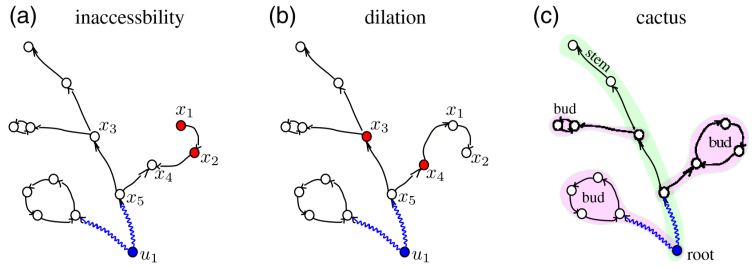


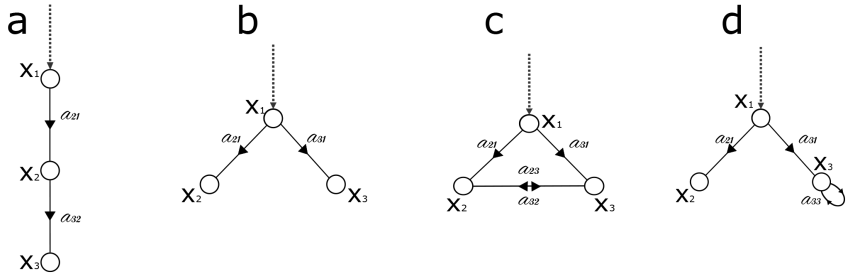
FIGURE 1.8 – (a) Inaccessibility. The red nodes x_1 and x_2 are inaccessible from the input (blue) nodes, which can not control the states of x_1 and x_2. (b) Dilations. The red nodes in the set $S = \{x_3; x_4\}$ cause a dilation. Indeed, their neighborhood set $T(S) = \{x_5\}$ contains only one node, implying that a single node in $T(S)$ aims to control two nodes in S. (c) A cactus contains neither inaccessible nodes nor dilations and it is a minimal structure for structural controllability. Note that in the cactus structure $T(S) = \{x_2; x_5\}$, hence the previous dilation has been removed. There is only one stem (highlighted in green) and multiple buds (highlighted in purple) are allowed in one cactus [9] .

Summarizing, according to Lin's Structural Controllability Theorem, the following statements are equivalent:

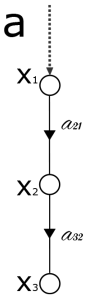
1. A linear system $(A;B)$ is structurally controllable;
2. The digraph $G(A;B)$ contains neither inaccessible nodes nor dilations;
3. $G(A;B)$ is spanned by cacti.

1.3.0.3 Examples of structural controllability

Likewise to the inverted pendulum, here we analyze the controllabilty of four systems, apparently similar to each other. Similarity which is not reflected in controllability terms.



Example A

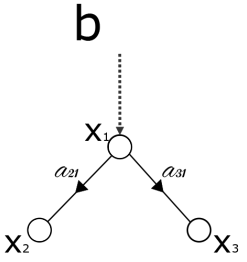


$$\begin{bmatrix} \dot{x_1}(t) \\ \dot{x_2}(t) \\ \dot{x_3}(t) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ a_{21} & 0 & 0 \\ 0 & a_{32} & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} + \begin{bmatrix} b_1 \\ 0 \\ 0 \end{bmatrix} u(t)$$

$$C = [B, AB, A^2B] = b_1 \begin{bmatrix} 1 & 0 & 0 \\ 0 & a_{21} & 0 \\ 0 & 0 & a_{32}a_{21} \end{bmatrix}$$

Since $rankC = 3 = N$, the system is controllable. Note that the system is always controllable despite the detailed values of matrices' entries. This is an example of the so called strong structural controllability.

Example B



$$\begin{bmatrix} \dot{x_1}(t) \\ \dot{x_2}(t) \\ \dot{x_3}(t) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ a_{21} & 0 & 0 \\ a_{31} & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} + \begin{bmatrix} b_1 \\ 0 \\ 0 \end{bmatrix} u(t)$$

$$C = [B, AB, A^2B] = b_1 \begin{bmatrix} 1 & 0 & 0 \\ 0 & a_{21} & 0 \\ 0 & a_{31} & 0 \end{bmatrix}$$

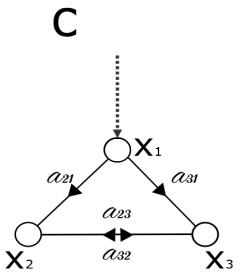
Since $rankC = 2 < N$, the system is uncontrollable. Note that this is independent of the detailed values of a_{21}, a_{31}, and b_1. No matter how we tune

them, the system is uncontrollable. Indeed, the dynamics equation suggests that the system will get stuck in the plane $a_{31}x_2(t) = a_{21}x_3(t)$ in the state space.

Example C

$$\begin{bmatrix} \dot{x_1}(t) \\ \dot{x_2}(t) \\ \dot{x_3}(t) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ a_{21} & 0 & a_{23} \\ a_{31} & a_{32} & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} + \begin{bmatrix} b_1 \\ 0 \\ 0 \end{bmatrix} u(t)$$

$$C = [B, AB, A^2B] = b_1 \begin{bmatrix} 1 & 0 & 0 \\ 0 & a_{21} & a_{23}a_{31} \\ 0 & a_{31} & a_{32}a_{21} \end{bmatrix}$$

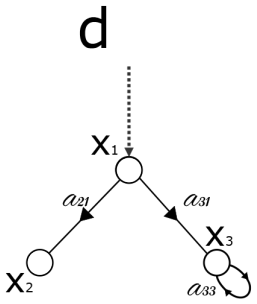


In most cases, we have $rankC = 3 = N$, so the system will be controllable. In case $a_{32}a_{21}^2 = a_{23}a_{31}^2$, it turns out that $rank(C) = 2 < N$ and the system will be uncontrollable. Therefore, this system is not strongly contollable.

Example D

$$\begin{bmatrix} \dot{x_1}(t) \\ \dot{x_2}(t) \\ \dot{x_3}(t) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ a_{21} & 0 & 0 \\ a_{31} & 0 & a_{33} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} + \begin{bmatrix} b_1 \\ 0 \\ 0 \end{bmatrix} u(t)$$

$$C = [B, AB, A^2B] = b_1 \begin{bmatrix} 1 & 0 & 0 \\ 0 & a_{21} & 0 \\ 0 & a_{31} & a_{33}a_{31} \end{bmatrix}$$



Since $rankC = 3 = N$, the system is controllable. In this example, the controllability is independent of the detailed values of a_{21}, a_{31} , a_{33}, and b_1, as long as they are non-zero. This is another example of strong structural controllability.

CHAPTER 2

Complex Brain Networks

2.1 Introduction

One of the most important functions of the brain is to process information and the primary information processing element is the *neuron*, which is a specialized brain cell that puts together many inputs to create a sole output.

A normal brain can be composed of a few neurons as well as more then a hundred billion. The output of one cell provides for the input of another, in order to create a *neural network* that is able of extraordinary achievements of calculations and decision making.

Indeed, there is a very long history in neuroscience behind the notion that the nervous system is a network of interconnected neurons.
In the second half of the nineteenth century was born the most important concept of the organization of the nervous system that came from studies of anatomical and physiological researches that organized the cellular basis of the brain function. At that time there was a huge debate regarding two different views of neural organization. On one side there was the one that now is synonymous of "neuron doctrine" which affirms that the nerve cell (neuron) was the anatomical, physiological, genetic and metabolic unit of the nervous system. On the other side it was denied the concept that neurons were delimitated structures, and it was suggested that the thin branches of neuronal fibers formed a continuous nerve network, that favorited neural activity to spread freely across the brain.[12] The debate was established by the turn of the century when it was stated that the neuron doctrine, was and still is one of the foundations of modern neuroscience.

Even after the debate ended, the researchers put a big effort into comprehending the way in which discrete cellular elements can fulfill continuity and collective action. This fact is far from resolved.
Grant that the brain is a complex dynamical system and addressing the problem with a network-based approach, the intent is to build an exhaustive understanding of how these networks are structurally organized and how they generate complex dynamics.

2.2 MAPPING THE BRAIN

In first approximation, a neuron be considered as a node that receives a certain number of inputs, produces an output result that is sent to one or more others neurons. The connections between neurons can be modelled with edges. The most important thing to realize a graph description of the brain system is outline the network's nodes.
Node definition can be inferred from electroencephalography or multielectrode-array electrodes, or anatomically defined regions of histological MRI or diffusion tensor imaging data.
Cortical parcellation, that consists in the definition of anatomical regions, it is essentially the division of the continuous cortical sheet into discrete areas. Parcellation schemes can be informed by the functional connectivity profiles of different regions or can use prior anatomical criteria. For example, one of the most famous parcellation scheme is Brodmann's division of the cortex into areas defined by their cytoarchitectonic criteria. Consequently parcellation schemes are not only already defined, and choosing other schemes can alter the resulting network features.
A different approach implicates defining nodes equivalent to individual electrodes or sensors, like in most magnetoencephalography and electroencephalography studies.

The second step that corresponds to the definition of edges is to determine the connectivity among the brain regions. In other terms, connectivity in the brain can be described as *structural*, *functional* or *effective*.

Structural connectivity is the description of anatomical connections linking a set of neural elements. Speaking at the scale of human brain, these connections usually relate to white matter projections linking cortical and subcortical regions. For example, diffusion MRI is capable to measure the anisotropic diffusion of water along fiber bundles, highlighting the connectivity between brain regions.
This kind of structural connectivity it is considered to be moderately stable on shorter time scale, but may fluctuate at longer time scales.

Functional connectivity is customarily assumed from time series observations, and describes patterns of statistical dependence among neural elements. The studies that allow to detect patterns of functional connections between cortical areas, are based on functional MRI (fMRI), electroencephalography (EEG), magnetoencephalography (MEG) or multielectrode array (MEA) data. For example, fMRI detects changes in regional brain activity through their effects on blood flow and blood oxygenation, which affect magnetic susceptibility and tissue contrast in magnetic resonance images; while EEG is a technique used to measure neural activity by monitoring electrical signals from the brain, usually through scalp electrodes.
Time series data can be calculated in different ways, as well as cross-correlation, mutual information or spectral coherence.
The presence of a statistical relationship between two neural elements is considered sometimes as a sign of functional coupling. It is necessary to understand that the presence of such coupling does not imply a causal relationship.
Because functional connections are repeatedly adjusted by sensory stimuli and task context, functional connectivity is highly time-dependent.

Effective connectivity is a very important way of representing and analysing brain networks. It strives to capture a network of directed causal effects between neural elements. That means that, effective connectivity embody a generative and mechanistic model that considers the observed data, selected from a range of possible models using objective criteria.
Most recent studies of brain networks are still accomplished on either structural or functional connectivity data sets, meanwhile effective connectivity represents a big promise for the future.

As a matter of fact, to obtain a brain network form, the defined nodes are coupled according to structural and/or functional brain connectivity data. The full set of all pairwise couplings can then be aggregated into connection matrix. To remove inconsistent or weak interactions, connections matrices is subjected to averaging across imaging runs or individuals, or to thresholding. A very important point is the choice of threshold used to create an adjancency matrix from the connection matrix: different threshold will generate graphs of different sparsity or connection density, and that's why network properties are often explored over a range of plausible thresholds.

All these steps are illustrated in 2.1

2.3 NETWORKS MEASURES

The resulting networks can be examined with the tools and methods of network science in order to analyze, and visualize network architecture.

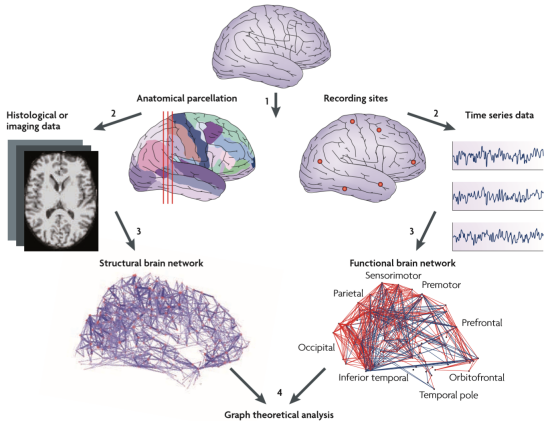


FIGURE 2.1 – A pictorial representation of the steps leading from data acquisition to a network-based brain model[13].

For example, one of the most fundamental network measure is the node degree, already encountered in Section 1.2:

$$k_i = \sum_j a_{ij}$$

Indeed, the *degree distribution* can be highly informative: the degrees of all nodes together form the degree distribution of the network, which shows whether the network contains nodes with nearly equal degrees or whether node degrees vary.

In random networks all connections are equally probable, resulting in a Gaussian and symmetrically centred degree distribution. In regular graphs, all the nodes have the same degree. On the other hand, complex networks generally have non-Gaussian degree distributions, often with a long tail towards high degrees, as we will see.

Furthermore, node degrees are fundamental because they are strongly related with a number of other network measures. One instance is *assortativity*, which represents the correlation between the degrees of connected nodes. Positive assortativity indicates that high-degree nodes tend to connect to each other.

The main networks measures can be divided in three categories: *segregation measures*, *integration measures* and *influence measures*.

2.3.1 *Segregation measures*

Segregation refers to the degree to which a network's elements form separate cliques or clusters.
The *clustering coefficient* of an individual node measures the density of connections between the node's neighbors. Densely interconnected neighbors form a cluster around the node, contrarily this phenomenon does not emerge with sparsely interconnected neighbors. The average of the clustering coefficients for each individual node is the clustering coefficient of the graph.

For example, random networks have low average clustering while complex networks have high clustering, usually associated with high local efficiency of communication, i.e. information travelling.

Different local neighborhoods or clusters often engage in different patterns. According to these patterns, large networks can be decomposed into smaller "building blocks". The distribution of such subgraphs, known as motifs, in a network provides information about the types of local interactions that the network can support.

Networks with high levels of clustering are often composed of local communities or modules of densely interconnected nodes. These modules are segregated from each other, such that most edges link nodes within modules, and few edges link nodes between different modules. The relation among the density of within-module and between modules connections defines a measure of network modularity.

Clustering, motifs, and modularity capture aspects of the local connectivity structure of a graph and the segregation of the network into communities. These measures can be traduced in mathematical terms as follows. [14]
Introducing the number of triangles around a node i,

$$t_i = \frac{1}{2} \sum_{j,h} a_{ji} a_{hi} a_{hj}$$

the clustering coefficient of the network is

$$C = \frac{1}{N} \sum_i C_i = \frac{1}{N} \sum_i \frac{2t_i}{k_i(k_i - 1)}.$$

Modularity is instead defined as

$$Q = \frac{1}{2k} \sum_{i,j} \left(a_{ij} - \frac{k_i k_j}{2k} \delta_{m_i, m_j} \right)$$

where m_i is the module containing the node i, and $\delta_{m_i, m_j} = 1$ if $m_i = m_j$, and 0 otherwise.

2.3.2 *Integration measures*

Integration refers to the capacity of the network as a whole to become interconnected and exchange information.

Many of these measures are based on paths and distances between nodes, such as the *path length*, i.e. the global average of the graph's distance matrix. In other words, it is the average of all the shortest path between each pair of nodes.
Denoting with L_i the average distance between node i and all other nodes,

$$L = \frac{1}{N}\sum_i L_i = \frac{1}{N}\sum_i \frac{\sum_{j\neq i} d_{ij}}{N-1}$$

A short path length indicates that, on average, each node can be reached from any other node along a path composed of only a few edges. A low path length usually can be interpreted as short communication distances.

The integration measures capture the capacity of the network to pass information between its nodes, and they are therefore of significance in a neurobiological context. For instance, structural paths that are shorter or are composed of fewer steps generally allow signal transmission with less noise, or signal degradation. [12]

Segregation and integration capture opposite features of networks construction, indeed, both segregation and integration turn out to be essential for structural and functional organization of brain networks.

2.3.3 *Influence measures*

Measures of influence attempt to quantify the "importance" of a given node or edge for the structural integrity or functional performance of a network.[15] Many of these measures capture the "centrality" of network elements, such as the *betweenness centrality* expressed as the number of short communication paths $\rho_{hj}(i)$ that travel through each pair of nodes h and j passing through the node i:

$$b_i = \frac{1}{N-1}\frac{1}{N-2}\sum_{h\neq j, h\neq i, j\neq i} \frac{\rho_{hj}(i)}{\rho_{hj}}$$

Indeed, the idea of betweenness centrality is that a node is central if it has great control over the flow of information within the network and that this control results from its participation in many of the network's short paths.
On the other hand, the *closeness centrality* of an individual node is the inverse

of the average path length between that node and all other nodes in the network.

$$L_i^{-1} = \frac{N-1}{\sum_j d_{ij}}$$

Furthermore, important nodes are often more highly or densely connected to the rest of the network, facilitate global integrative processes, or play a critical compensatory role when the network is damaged. Such nodes are often referred to as *hubs*, a term that is often used notwithstanding imprecisely defined. Hubs can be identified on the basis of several different criteria, including the degree, participation in modular connectivity, or centrality.

Recently studies carried out in humans and other species had suggested a tendency for hubs to be densely interconnected in a "hub complex": a study of cat cerebral cortex[16] has highlighted for the first time the existence of "rich club" structure, a set of hub regions that are densely interconnected. The structural connectivity acquired with MRI shows that even human brain is organized through rich clubs.

2.4 RANDOM NETWORKS ARCHITECTURE

Network science aims to build models that reproduce the properties of real networks. In most real networks, there is no the emergence of regularity, as well as a crystal lattice. Rather, they seems to be characterized by a certain degree of randomness. Random network theory embraces this apparent randomness by constructing and characterizing networks with a random architecture. Some instances are illustrated in Figure 2.2

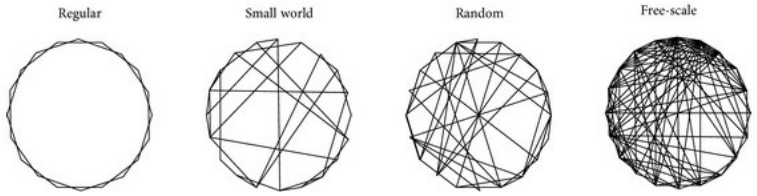


FIGURE 2.2 – Pictorial representation of the architecture of the networks according to the way in which they were built.

Generally, there are different ways to generate random networks, whose topological architecture is tuned by characteristic random parameters, according to the generative model taken into account.

2.4.1 *Erdös-Rényi*

From a modeling perspective a network is a relatively simple object, consisting of only nodes and links. The real challenge, however, is to decide where to place the links between the nodes so that we reproduce the complexity of a real system. A particularly rich source of ideas has been the study of random graphs, graphs in which the edges are distributed randomly. Networks with a complex topology and unknown organizing principles often appear random; thus random-graph theory is regularly used in the study of complex networks.

The theory of random graphs was introduced by Paul Erdös and Alfréd Rényi, after the first one discovered that probabilistic methods were often useful in tackling problems in graph theory.

In their classic first article on random graphs [17], Erdös and Rényi define a random graph as N labelled nodes connected by m edges, which are chosen randomly from the $N(N-1)/2$ possible edges.

An alternative and equivalent definition of a random graph is the binomial model. Here we start with N nodes, every pair of nodes being connected with probability p, or equivalently all graphs with N nodes and M edges have equal probability of:

$$p^M(1-p)^{\binom{N}{2}-M}$$

Each node's degree is characterized by the a binomial distribution, i.e. each nodes is the endpoint of x edges with probability:

$$p_k = \binom{N-1}{x} p^k (1-p)^{N-1-k} \tag{2.1}$$

From 2.1, it is easy to deduce that

$$< k >= p(N-1)$$

The 2.1 is said the *degree distribution* of the network. Indeed, in a given realization of a random network, some nodes gain numerous links, while others acquire only a few or no links. These differences are captured by p_k which is the probability that a randomly chosen node has degree k.

Most real networks are sparse, meaning that for them $N \gg < k >$: in this limit the degree distribution is well approximated by the Poisson distribution

$$p_k = e^{-<k>} \frac{< k >^k}{k!}$$

In the mathematical literature the construction of a random graph is often called an evolution: starting with a set of N isolated vertices, the graph develops by the successive addition of random edges. The graphs obtained at different stages of this process correspond to larger and larger connection probabilities p, eventually obtaining a fully connected graph.

According to the evolving p, or equivalently $< k >$, a transition occurs in the Erdös-Rényi networks: the emergence of a giant component, i.e. a connected subgraph.
We can distinguish four topologically distinct regimes, each with its unique characteristics:

- Subcritical Regime: $p < 1/N$
  For $< k >= 0$ the network consists of N isolated nodes. Increasing $< k >$ means that we are adding $N < k >= pN(N-1)/2$ links to the network. We have only a small number of links in this regime, hence we mainly observe tiny clusters, without the emergence of a giant component
- Critical Point: $p = 1/N1$
  The critical point separates the regime where there is not yet a giant component ($< k >< 1$) from the regime where there is one ($< k >> 1$). At the critical point most nodes are located in numerous small components: these numerous small components are mainly trees, while the giant component may contain loops.
- Supercritical Regime: $p > 1/N$
  This regime has the most relevance to real systems, as for the first time we have a giant component. The giant component contains a finite fraction of the nodes. The further we move from the critical point, a larger fraction of nodes will belong to it. In the supercritical regime numerous isolated components coexist with the giant component. These small components are trees, while the giant component contains loops and cycles. The supercritical regime lasts until all nodes are absorbed by the giant component.
- Connected Regime: $p > lnN/N$
  For sufficiently large p the giant component absorbs all nodes and components and in the absence of isolated nodes the network becomes connected.

The random network model predicts that the emergence of a network is not a smooth, gradual process: The isolated nodes and tiny components observed for small $< k >$ collapse into a giant component through a phase transition. As we vary $< k >$ we encounter four topologically distinct regimes.

2.4.2 *Small world*

Ordinarily, the connection topology is assumed to be either completely regular or completely random. But many biological, technological and social networks

lie somewhere between these two extremes. Here we explore simple models of networks that can be tuned through this middle ground: regular networks 'rewired' to introduce increasing amounts of disorder. Indeed, in order to interpolate between regular and random networks, we consider the following random rewiring procedure. Starting from a ring lattice with n vertices and k edges per vertex, each edge is rewired at random with probability p. This construction, illustrates in figure 2.3 allows us to 'tune' the graph between regularity ($p = 0$) and disorder ($p = 1$), and thereby to probe the intermediate region $0 \leq p \leq 1$

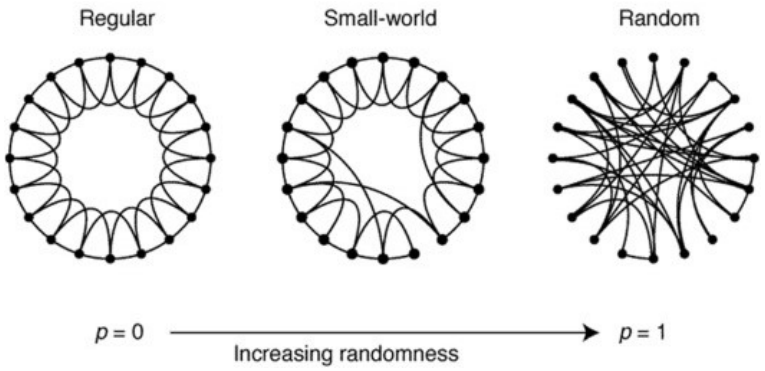


FIGURE 2.3 – Rewiring processes that leads from a regular networks to the random. In the middle there is this small world regime network.

We find that these systems can be highly clustered, like regular lattices, yet have small characteristic path lengths, like random graphs. We call them 'small-world' networks, by analogy with the small-world phenomenon. This last one, popularly known as the *six degrees of separation problem*, states that two individuals, anywhere in the world, can be connected through a chain of six or fewer acquaintances.
In the language of network science six degrees, also called the small world property, means that the distance between any two nodes in a network is unexpectedly small.

We quantify the structural properties of these graphs by their characteristic path length $L(p)$ and clustering coefficient $C(p)$. Here L(p) measures the typical separation between two vertices in the graph (a global property), whereas C(p) measures the cliquishness of a typical neighbourhood (a local property). The networks of interest to us have many vertices with sparse connections, but not so sparse that the graph is in danger of becoming disconnected. Specifically, we require $N \gg k \gg ln(N) \gg 1$, where $k \gg ln(N)$ entures that the network is connected.
Under this condition, we find that the regular lattice at $p = 0$ is a highly clustered, large world where L grows linearly with N, whereas the random

network at $p = 1$ is a poorly clustered, small world where L grows only logarithmically with n. These limiting cases might lead one to suspect that large C is always associated with large L, and small C with small L.

ON the contrary, the model for $0 < p < 1$ interpolates between a regular lattice, which has high clustering but lacks the small-world phenomenon, and a random network, which has low clustering, but displays the small-world property.

Indeed, in [18], it is provided that numerical simulations show that there is a broad interval of p over which L(p) is almost as small as L_{random} yet $C \gg C_{random}$. These small-world networks result from the immediate drop in L(p) caused by the introduction of a few long-range edges. Such 'short cuts' connect vertices that would otherwise be much farther apart than L_{random}. For small p, each short cut has a highly nonlinear effect on L, contracting the distance not just between the pair of vertices that it connects, but between their immediate neighbourhoods, neighbourhoods of neighbourhoods and so on.

In this case, defining $f(k, < k >) = min(k - \frac{<k>}{2}, \frac{<k>}{2})$, the degree distribution follows:

$$P(k) = \sum_{n=0}^{f(k,<k>)} C^n_{\frac{<k>}{2}} (1-p)^n p^{\frac{<k>}{2}-n} \frac{(p\frac{<k>}{2})^{k-\frac{<k>}{2}-n}}{(k-\frac{<k>}{2}-n)!} e^{-p\frac{<k>}{2}}$$

and it has a pronounced peak at $< k >$, decaying exponentially for large k.

2.4.3 *Scale-free*

Networks with power-law degree distribution, such as the Internet one, are called *scale-free networks*.

$$P(k) \propto k^{-\gamma}$$

It is interesting the fact that a number of empirical networks presents scale-free structure with $2 < \gamma < 3$.
The main difference between a random and a scale-free network comes in the tail of the degree distribution, representing the high-k region of $p(k)$, as shown in Figure 2.4.

For small k the power law is above the Poisson function, indicating that a scale-free network has a large number of small degree nodes, most of which are absent in a random network. While, for k in the vicinity of $< k >$ the Poisson distribution is above the power law, indicating that in a random network there is an excess of nodes with degree $k\sigma < k >$. Again, for large k the power law is again above the Poisson curve.
Indeed, power-law distribution is heavy tailed, fact that involves the occurrence of hubs, which do not compare in random graphs. Hubs represent a

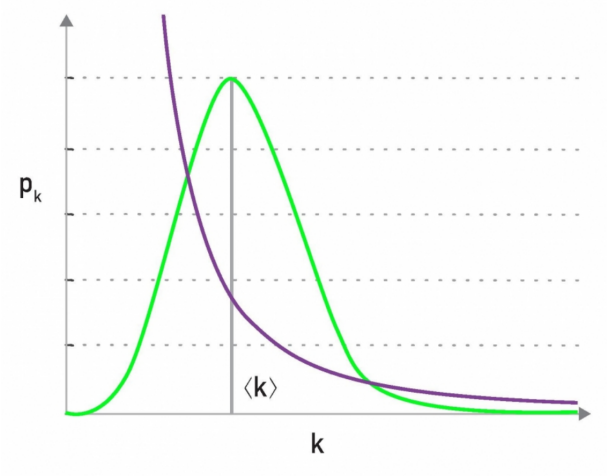


FIGURE 2.4 – Scale free degree distribution (violet) and a random degree distribution (green).

signature of a deeper organizing principle that we call the scale-free property.

As the name suggests, scale-free networks lack of an internal scale, a consequence of the fact that nodes with widely different degrees coexist in the same network. To best understand the meaning of the scale-free term, we consider the generating moment function

$$< k^n >= \int_{k_{min}}^{\infty} k^n p(k) dk$$

Therefore, in the thermodynamics limit, the first moment $< k >$ is finite, but the second one diverges for $2 < \gamma 3$. The divergence of the second moment involves that the fluctuations around the average can be arbitrary large. This means that when we randomly choose a node, we do not know what to expect: the selected node's degree could be tiny or arbitrarily large. Hence networks with $\gamma < 3$ do not have a meaningful internal scale, but are "scale-free".

The case $2 < \gamma < 3$ shoes another property: networks are characterize by ultra-small regime, as the hubs radically reduce the path length through the linking of a large number of small-degree nodes, creating short distances between them.

CHAPTER 3

Controllability and Matching Problems

As previously said, the whole network is controllable if one controls each node individually. Nevertheless, this is nearly impossible, thus the goal is to develop a technique to identify the minimum input required to control the system.
Unfortunately, structural controllability only provides conditions to classify whether a system is controllable, not mentioning the input required.
However, minimum inputs, or equivalently the minimum driver node set, can be identified by mapping the control problem into a purely graph-theoretical problem called maximum matching.

3.1 Matching

Given a graph, a **matching** is a set of non-loop edges with no common endpoints. In other words, a matching is the set of edges with the property that no pair of edges from this set shares a node in common.
The vertices incident to the edges of a matching M are saturated by M; the others are unsaturated.
A **perfect matching** in a graph is a matching that saturates every vertex.
For example, a complete graph of odd vertices does not have a perfect matching, but a complete graph of even vertices always has a perfect matching. A graph may have many perfect matchings: for example, $K_{n,n}$ has $n!$ perfect matchings.

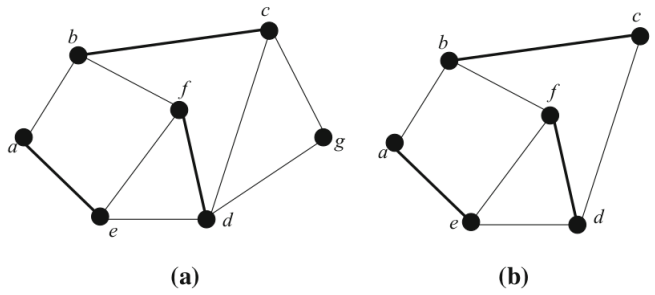


FIGURE 3.1 – (a) the set $\{(a,e),(b,c),(d,f)\}$ of edges is a matching: vertices a, b, c, d, e, and f are saturated, whereas the vertex g is unsaturated; (b) a perfect matching of a graph, indeed all vertices are saturated.

A **maximal matching** in a graph is a matching that cannot be enlarged by adding a further edge. A **maximum matching** is a matching of maximum size among all matchings in the graph.

It has to be remarked that a matching M is maximal if every edge not in M is incident to an edge already in M. Thus, every maximum matching is a maximal matching, but the converse doesn't need to hold.

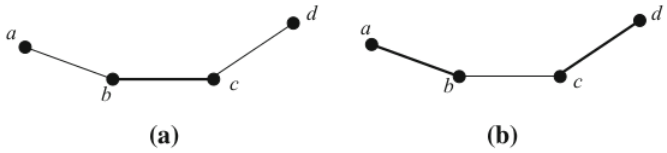


FIGURE 3.2 – (a) The matching $\{(b,c)\}$ is a maximal matching but not a maximum matching, (b) since there is a larger matching $\{(a,b),(c,d)\}$ of this graph.

In order to determine the existence of a matching, it is useful introduce the concepts of **alternating path** and **augmenting path**.
Given a matching M, an alternating path is a path that alternates between edges in M and edges not in M. If the endpoints of the M-alternating path are unsaturated the path is said augmenting path.

Given an M-augmenting path P, we can replace the edges of M in P with the other edges of P to obtain a new matching M' with one more edge. Thus when M is a maximum matching, there is no M-augmenting path. In fact, we prove in Appendix A that maximum matchings are characterized by the absence of augmenting paths as follows:

THEOREM 3.1 (Berge Theorem) – A matching M in a graph G is a maximum matching in G if and only if G has no M-augmenting path.

In other words, the algorithmic problem of finding such matchings turns into the one of finding augmenting paths, which is an accessible algorithmic problem, specially in bipartite cases.
Matching in bipartite graphs is a well known problem with a number of real-world applications.
Furthermore, it will be shown that the bipartite case is related to directed graphs, therefore it is useful to deepen the framework of matching in graphs with a bipartition.

3.2 CHARACTERIZATION OF MATCHING IN BIPARTITE GRAPHS

Within a set of people, some pairs are compatible as roommates: under what conditions can we pair them all up?
Alternatively, a company has received applications for its job vacancies. Assume that the company has a set J of job vacancies and has received a set A of applicants. An applicant in A may have applied for several jobs, but each applicant will be given at most one job. The number of applicants is larger than the number of jobs. But does it ensure that every vacancy will be filled out?
These are some of real-world problems that can be modelled with a bipartite graph and which require matching tools for an optimized resolution.

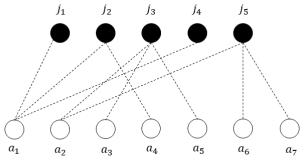


FIGURE 3.3 – A representation of the *job and applicants problem*.

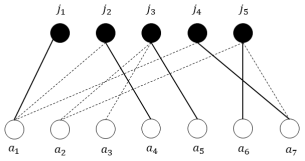


FIGURE 3.4 – A matching indicating a feasible solution for job vacancy fill out.

Thus, let define a biparite graph, with partition $\{A, B\}$, thus the vertices labelled with a_i belong to the partition A and the ones labelled with b_i to B

Let U be a set such that $U \subseteq V$ and every edge of G is incident with a vertex in U, than U is a **cover** of G.

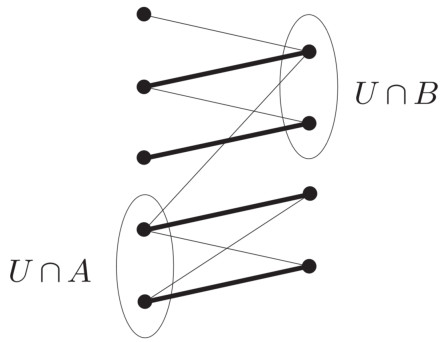


FIGURE 3.5 – The vertex cover U.

Our first theorem characterizes the maximal cardinality of a matching in G by a kind of duality condition:

THEOREM 3.2 (König's Theorem) – The maximum cardinality of a matching in G is equal to the minimum cardinality of a vertex cover of its edges.

A condition clearly necessary for the existence of a matching of A is that every subset of A has enough neighbours in B, i.e. that:

THEOREM 3.3 (Hall's Theorem) – A bipartite graph G has a matching which covers every vertex in A if and only if

$$|N(S)| \geq |S| \qquad \forall S \subseteq A$$

where N(S) is the set of all neighbours of the vertices in S

For example, in the graph in 3.2, the vertex set $S = \{j_1, j_4\}$ has only one neighbor, and hence it does not satisfy the condition of Hall's Theorem.

The generalization of Hall's theorem deals with bipartite graphs whose partition has the same size, i.e. cardinality.
The *marriage theorem* offers a sufficient and necessary condition to have a

perfect matching:

THEOREM 3.4 (Marriage Theorem) – A bipartite graph G has a perfect matching if and only if $|X| = |Y|$ and $|N(S)| \geq |S|$ for all $S \subseteq X$.

Summarizing, *Frobenius' theorem* characterizes those bipartite graphs which have a perfect matching. *Hall's Theorem* characterizes those bipartite graphs which have a matching of A into B . *König's Theorem* gives a formula for the matching number of a bigraph.

3.2.1 *Maximum matching in digraph and algorithm perspective*

The aim is to generalize the tools of maximum matching theory for oriented graphs.
For a digraph D, an edge subset M is a matching if no two edges in M share a common starting vertex or a common ending vertex.
A vertex is matched if it is an ending vertex of an edge in the matching. Otherwise, it is unmatched.
Similarly to the undirected case, a matching of maximum cardinality is called a maximum matching. A maximum matching is called perfect if all vertices are matched. For example, in a directed elementary cycle, all vertices are matched.

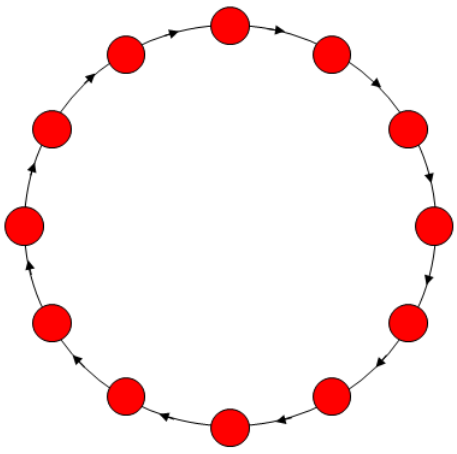


FIGURE 3.6 – A perfect matching (red nodes) in a elementary directed cycle.

The maximum matching for a digraph can be identified by mapping the digraph to its bipartite representations $H(D)$.

More precisely, $H(D) = \{V_G^+ \cup V_G^-, \Gamma\}$, where $V_G^+ = \{x_1^+ ... x_N^+\}$ are the starting vertices of the directed edges, conversely $V_G^- = \{x_1^- ... x_N^-\}$ are the end ones. While the edge set is defined from adjacency matrix as $\Gamma = \{(x_j^+, x_i^-) | a_{ij} \neq 0\}$.
In other words, we split each node x_i of the original digraph into two "nodes" x_i^+ and x_i . We then place an edge (x_j^+, x_i^-) in the bipartite graph if there is a directed edge $x_j^+ \rightarrow x_i^-$ in the original digraph, as shown in 3.7.

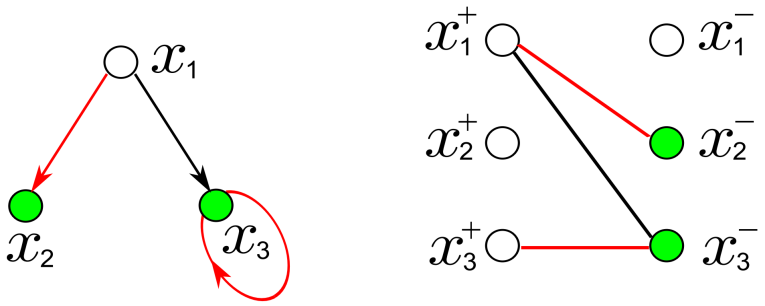


FIGURE 3.7 – A digraph and its bipartite representations.

From a numerical point of view, the bipartite representation is powerful because the maximum matching problem in a digraph is not NP hard, but can be solved in polynomial time.
A maximum matching of a bipartite graph can be found efficiently using the Hopcroft-Karp algorithm, which runs in $O(\sqrt{V}E)$ time, as illustrated in Appendix B

3.3 CONTROLLABILITY AND MATCHING PROBLEMS

The usefulness of matching in network control comes from a theorem that provides the minimum number of inputs required to control the entire network.
Indeed, for an LTI system described by the digraph $G(\mathbf{A})$, the problem of finding the minimum number of driver nodes is related to matching theory as follow:

THEOREM 3.5 (Minimum Input Theorem) – The minimum number of inputs N_{input} or equivalently the minimum number of driver nodes N_D needed to fully control a network $G(\mathbf{A})$ is the number of unmatched nodes with respect to any maximum matching M. Whether the maximum matching is perfect, the system is controllable through a single driver node, independently of which one.

In mathematical terms:

$$N_{input} = N_D = max\{1; N - |M|\}$$

According to the interpretation of structural controllability, related to the occurrence of dilations or isolated nodes, in order to fully control the network through suitable inputs to the system, each node has to have its own "superior". The minimum inputs needed is determined by the maximum matching of the network. Indeed the idea of this theorem is that a matched node has already been controlled by its "superior", i.e. the node pointing to it. But unmatched node has to be controlled directly by an external "superior" or input. Thus, they are the driver nodes for the whole system. Therefore, injecting inputs into all the driver nodes, then each node has its own "superior" and the system is fully controllable.

Furthermore, the extended definition of matching on a digraph connects more naturally to the cactus structure. Indeed, a matching of a digraph can be decomposed into a set of directed paths and/or directed cycles, which are the basic elements of the cactus structure.
Hence, matching in digraphs connects naturally to the cactus structure.

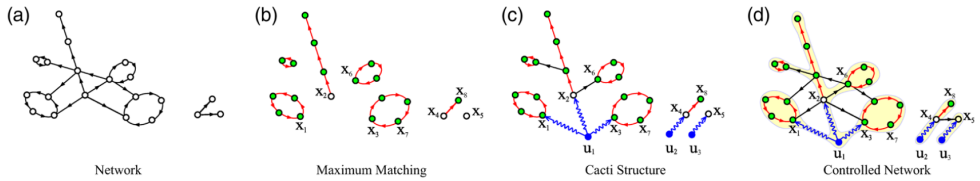


FIGURE 3.8 – (a) A directed network. (b) The maximum matching. All maximum matchings can be decomposed into a set of vertex-disjoint directed paths and directed cycles, shown in red. If a node is the head of a matching edge, then this node is matched (green nodes). Otherwise, it is unmatched (white nodes). The unmatched nodes must be directly controlled to control the whole network, hence they are the driver nodes. (c) By injecting signals into driver nodes, we get a set of directed paths whose starting points are the input nodes. The resulting digraph is a cactus. (d) According to the structural controllability theorem, since there is a cacti structure (highlighted in yellow) underlying the controlled network, the system is structurally controllable. Maximum matching identifies the minimal cacti, i.e., the cacti structure with the minimum number of roots. The minimal cacti structure serves as the control skeleton that maintains the structural controllability of the system.

3.3.1 *Analytical framework*

While the maximum matching allows to efficiently identify the minimal driver nodes set, the algorithmic approach provides no physical insights about the impact of the network topology on N_D.
For a random digraph ensemble with a given degree distribution $P(k_{in}, k_{out})$, it is possible to analytically calculate $n_D = N_D/N$, representing the fraction of driver nodes averaged over all network realizations of the ensemble.
Given a matching M in a digraph $G = \{V(G), E(G)\}$, it can be described introducing a binary variable $s_a = s_{(i\to j)} \in \{0, 1\}$ assigned to each edge $a = (i \to j) \in E(G)$ with $s_a = 1$ if a belongs to the matching M and $s_a = 0$ otherwise. According to the definition of matching in a digraph, one has two constraints on each vertex $i \in V(G)$:

$$\sum_{j\in\partial^+ i} s_{(i\to j)} \leq 1$$

$$\sum_{k\in\partial^- i} s_{(k\to i)} \leq 1$$

with $\partial^- i$ and $\partial^+ i$ indicating the sets of nodes which point to i or are pointed by i, respectively.

The quantity $\epsilon_i(\{s\}) = 1 - \sum_{k\in\partial^- i} s_{(k\to i)}$ tells us the state of each vertex: vertex i is matched if $\epsilon_i(\{s\}) = 0$ and unmatched if $\epsilon_i(\{s\}) = 1$.
Consequently, introducing the cost (or energy) function which gives, for each matching $M = \{s\}$:

$$\epsilon_G(\{s\}) = \sum_{i\in V(G)} \epsilon_i(\{s\}) = N - |M|$$

it is possible to define the Boltzmann probability in the space of matching

$$P_G(\{s\}) = \frac{e^{-\beta\varepsilon_G(\{s\})}}{Z_G(\beta)}$$

where, as usual, β is the inverse of the temperature and Z_G is the partition function

$$Z_G = \sum_{\{s\}} = e^{-\beta\varepsilon_G(\{s\})}$$

Following the statistical physics description, it is possible to deduce from the partition function both the internal energy $\epsilon_G(\beta)$ and the entropy function $S_G(\beta)$.
In the zero temperature limit, i.e. $\beta \to \infty$, the internal energy and the entropy provide the ground state properties, i.e., the properties of the maximum matchings. In particular, $\epsilon_G(\beta)$ represents the number of unmatched vertices (with respect to any maximum matching), and the entropy $S_G(\beta)$ yields the

logarithm of the number of maximum matchings.
In [9], it is provided an analytical expression for the ratio n_D of driver nodes, through the use of the cavity method, a versatile tool of statistical physics.

In the zero temperature limit, the minimum density of unmatched nodes can be derived from the average energy density.

Introducing the out- and in-degree distributions of node i when one selects uniformly at random a directed edge $i \to j$ from the digraph:

$$Q(k_{out}) = \frac{k_{out} P(k_{out})}{\langle k_{out} \rangle}$$

$$\widehat{Q}(k_{in}) = \frac{k_{out} \widehat{P}(k_{in})}{\langle k_{in} \rangle}$$

Thus, defining the generative function as follow:

$$G(x) = \sum_{k_{out}=0}^{\infty} P(k_{out}) x^{k_{out}}$$

$$\widehat{G}(x) = \sum_{k_{in}=0}^{\infty} \widehat{P}(k_{in}) x^{k_{in}}$$

$$H(x) = \sum_{k_{out}=0}^{\infty} Q(k_{out}+1) x^{k_{out}}$$

$$\widehat{H}(x) = \sum_{k_{in}=0}^{\infty} \widehat{Q}(k_{in}+1) x^{k_{in}}$$

It turns out that the minimum density of driver nodes is given by:

$$n_D = \frac{1}{2}\left\{ [G(\widehat{w_2}) + G(1-\widehat{w}_1) - 1] + [\widehat{G}(w_2) + \widehat{G}(1-w_1) - 1] + \frac{< k >}{2} [\widehat{w}_3 + w_3] \right\} \quad (3.1)$$

where $< k >$ is the average degree of the network and $w_1, w_2, w_3, \widehat{w}_1, \widehat{w}_2, \widehat{w}_3$ satisfy the set of self-consistent equations:

$$w_1 = H(\widehat{w}_2) \qquad w_2 = 1 - H(1-\widehat{w}_1) \qquad w_3 = w_1(1-\widehat{w}_2)$$
$$\widehat{w}_1 = \widehat{H}(_2) \qquad \widehat{w}_2 = 1 - \widehat{H}(1-w_1) \qquad \widehat{w}_3 = \widehat{w}_1(1-w_2)$$

Therefore, it is possible to characterize the distribution of the driver nodes as function of the average degree $< k >=< k_{in} > + < k_{out} >$, according to $P(k_{in}, k_{out})$ which defines the architecture of the networks belonging to the same ensemble.

3.4 CHARACTERIZATION OF NODES AND EDGES

As previously said, the maximum matching can be not uniquely determined. Consequently, a system can be controlled by multiple driver node configurations, each corresponding to a different maximum matching.
Some nodes or edges may appear more often in a maximum matching than other elements.
Therefore, one can characterize the nodes and the edges in terms of their role with respect to the possible configurations of maximum matching.

3.4.1 *Edges classification*

It is natural to wonder how the removal or addition of a link can impact the controllability of the network.
According to this criterion, an edge can be:

1. *critical* if in its absence we must increase the number of driver nodes to maintain full control over the system. In this case the link is part of all maximum matchings of the network;
2. *redundant* if it can be removed without affecting the current set of driver nodes (i.e., it does not appear in any maximum matching);
3. *ordinary* if it is neither critical nor redundant (it appears in some but not all maximum matchings).

It turns out that most real networks have few or no critical links. Most links are ordinary, meaning that they play a role in some control configurations, but the network can be still controlled in their absence.
It is easy to understand that the occurrence of critical link is related to the network's architecture: for example, meanwhile $< k >$ is small all the edges are important in order to control the entire system, and therefore the ratio of critical edges tends to 1

3.4.2 *Nodes classification*

Given the existence of multiple driver node configurations, we can classify nodes based on their role with respect to the minimum driver nodes set, namely MDNS.
A node is:

1. critical if that node must always be controlled to control the system, implying that it is part of all MDNSs;
2. redundant if it is never required for control, implying that it never participates in an MDNS;
3. intermittent if it is a driver node in some control configurations, but not in others.

Similarly to the edges classification, one can classify a node with respect to the impact of its removal on controllability.

1. A node is deletion critical if in its absence we have to control more driver nodes;
2. A node is deletion redundant if in its absence the driver nodes are less to control the network;
3. A node is deletion ordinary if in its absence the number of minimum driver nodes required does not change.

For example, moving a node in the middle of a directed path will increase N_D, removing a leaf node in a star will decrease N_D by 1 and the central hub in a star is a deletion ordinary node.

A further classification can be applied on driver nodes themself.
The MDNS can divided into three groups:

1. source nodes (N_s) that have no incoming links,hence they must be directly controlled, being always driver nodes;
2. external dilations (N_e) arise due to a surplus of sink nodes that have no outgoing links;
3. internal dilations (N_i) occur when a path must branch into two or more paths in order to reach all nodes (or equivalently a subgraph has more outgoing links than incoming links).

CHAPTER 4

Structural Controllability in Complex Brain Networks

The only animal whose connectome has been completely visualized, is the Caenorhabditis elegans, a nematode worm about 1mm in lenght.
Thus, it is a good test case to apply the developed tools of controllability. Furthermore, it will be investigate some null model networks, in order to compare the found results.

4.1 Why Caenorhabditis elegans?

Around the world thousands of scientists are investigating the biology of *Caenorhabditis elegans*. This nematode is evolutionary rudimentary, nevertheless shares many of the essential biological characteristics that are central problems of human biology.
The microscopic roundworm Caenorhabditis elegans lives in the soil of temperate climates. Millions of individuals can be found underneath a single square meter of moist vegetated ground. The worm's tube-like body reaches a length of about 1 millimeter, and lacking vision or hearing, its neural structures provide a series of sense organs in the head, which mediate responses to chemical, thermal, and tactile stimulation. Feeding mostly on bacteria in the ground, its behavioral repertoire ranges from relatively simple activities like locomotion or swimming to complex activities involving reproduction and even rudimentary forms of social interactions.

It has a nervous system with a "mind", composed of nearly 300 neurons connected through chemical synapses and gap junctions. The neurons are divided into various sub-types and are classified based on their functional roles and location within the body of the animal. According to functional roles, neurons are primarily of three types: sensory neurons, motor neurons and inter neurons.
Due to the richness of biological functions, for many years, C. elegans has been a favored model organism for developmental biologists, in part due to the ease with which it is grown in the laboratory and the relative simplicity of its body structure.
Due to all these reasons, C. elegans offers a great compromise between complexity and tractability and lines of research occur in a number of fields, even working on experiments finalized to the construction of ever larger data sets, in order to achieve a better understanding of the worm organism and function.
An example of the results of studies carried on this nematode, C. elegans is also among the very first organisms whose genome was sequenced and mapped in its entirety.
Thanks to the work of White [19] on Caenorhabditis elegans, the structure and connectivity of the complete nervous system was investigated and published in 1986 in *Philosophical Transactions B*, marking a critical milestone in the worldwide effort to study this nervous system.
An important feature of the worm's nervous system has helped the study of the nervous wiring diagram: the spatial position, number, and connectivity of its neurons are largely constant across individuals.
To this day, it remains the only nervous system of any organism whose connectivity structure is completely mapped at the level of individual cells and synapses.

4.1.1 *A simple brain*

The mapping of nervous system of C. elegans was accomplished by painstaking reconstruction of the three-dimensional wiring pattern from electron micrographs (EMs) of a complete stack of serial sections. The reconstruction work was performed largely by hand and took more than ten years to complete. The invariance of the structure of the nervous system across individuals, as well as the relatively simple morphology of many of its neurons aided in the reconstruction effort.
For example, recently works are finalized to generate a more complete reconstruction of the brain of C. elegans, such as the determination of a ever more precisely connection matrix.
A unique feature of the data set about this worm is that the spatial position of each neuron and hence the length of all synaptic connections are known.

These data on the spatial layout of the worm's nervous system allowed a detailed analysis of wiring length, providing important insights into spatial embedding and wiring minimization as possible constraints on neuronal placement and connectivity.

Although the availability of the complete map of all cells and connections in the nervous system of C. elegans, a full-scale computational model of the worm's nervous system lacks, as well as a complete description of its functional behaviors.
This situation reminds us that the complete wiring diagram is insufficient for reconstructing functional dynamics of a neural system in the absence of complementary information about the biophysical properties of neurons and synapses. These biophysical properties have a large role in determining the dynamic characteristics of neuronal activations, neural transmission, and synaptic plasticity.

From a network perspective, the connection pattern of the nematode is strongly nonrandom. Connections between neurons predominantly occur within local neighborhoods, involving high clustering feature.
Indeed, as first noticed in [18], the worm brain is can be described as a small-world network.
According to its small-world nature, in [20], it is highlighted the occurrence of a small number of highly connected neurons as a rich club interconnected with high efficiency and high connection distance. The rich club neurons are connector hubs, with high betweenness centrality, and many intermodular connections to nodes in different modules. Finally, another important aspect is the emergence of a large number of densely connected three-nodes motifs. Furthermore, the motifs that are found most frequently traverse the rich club.

4.2 REAL NETWORK

As it said, branches of graph theory and network science have entered into the domain of neuroscience, and are bringing with them novel quantitative perspectives on the complexity of nervous systems.
According to the formalism provided by these disciplinary, the worm brain can be visualized as a network, where the nodes represent the neurons and the edges model both synapses.
Approaching a real-world data set, it has been investigated a local subnetwork of neurons within C. elegans rostral ganglia (anterior, dorsal, lateral, and ring). [21]

The network is composed of 131 neurons with 764 directed connections. Connections includes both chemical and electrical synapses. In chemical synapses, an electrical signal is transformed to a chemical signal (release of a neurotransmitter) and then transformed back to an electrical signal in the responding cell (neuron or muscle) through postsynaptic neurotransmitter receptors that are ion channels. While electrical synapses, or gap junctions, are channels by which electrical current can flow between coupled cells. In principle, current can flow in either direction through the electrical synapse, however at any given time it can only flow in one direction, determined by the relative membrane potentials of the coupled cells.

Neuronal connectivity is largely based on the dataset of White [19] in which connections were identified by electron microscope reconstructions, as previously mentioned.

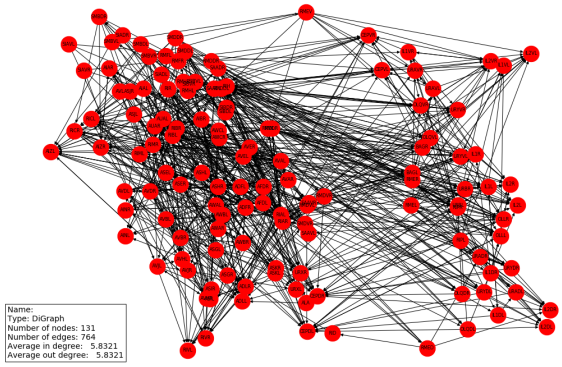


FIGURE 4.1 – Graphical representation of the network of frontal ganglia neurons of *C. elegans*.

The dataset is available at `https://www.dynamic-connectome.org`. [21] [22] [23]

4.2.1 *Network analysis and controllability results*

The aim is to apply control principles on the real-world network examined. The problem of identifying the minimum driver nodes set has been addressed through the algorithmic approach discussed about maximum matching in digraphs, determined with the *Hopcroft-Karp algorithm*, illustrated in Appendix B

Preliminary, the network has been analyzed and the main features can be summarized in the following way.
The network is weakly connected, i.e. if the underlying network is connected. While it presents a giant strongly-connected component, with 109 nodes and 637 edges.

Number of nodes:	131
Number of edges:	764
Nodes of giant weakly-connected component	131
Edges of giant weakly-connected component	764
Nodes of giant strongly-connected component	109
Edges of giant strongly-connected component	637
Average in-degree:	5.8321
Average out-degree:	5.8321
Cluster coefficient:	0.149496
Path lenght:	3.1277

In order to fully control the network, maximum matching provides the minimum driver nodes set.
In 4.2, it is shown a pictorial representation of the found results:

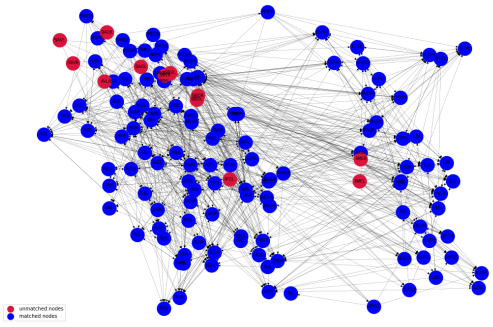


FIGURE 4.2 – The saturated nodes belonging to the maximum matching are those in blue; the unmatched nodes, i.e. the driver nodes required to control the whole network, are highlighted in crimson.

In this context, it is found that 12 *neurons* are needed in order to control the system.
In particular, the driver nodes highlighted in 4.2 are the following ones:

Neuron	Type
AFDL	sensory neuron (thermosensory and CO_2- sensory)
AVL	Polymodal (interneuron, motor neuron)
RMEL	Motor neuron
RMER	Motor neuron
SIADL	Interneuron, (sublateral) motor neuron
SIADR	Interneuron, (sublateral) motor neuron
SIAVL	Interneuron, (sublateral) motor neuron
SIAVR	Interneuron, (sublateral) motor neuron
SIBDL	Interneuron, (sublateral) motor neuron
SIBDR	Interneuron, (sublateral) motor neuron
SIBVL	Interneuron, (sublateral) motor neuron
SIBVR	Interneuron, (sublateral) motor neuron

4.3 RANDOM NETWORKS ANALYSIS

In general, a random graph is a model network in which some specic set of parameters take fixed values, but the network is random in other respects, as seen in Chapter 2.
An important part of any graph-theoretical analysis is the comparison of measures obtained from empirical networks to networks representing a "null hypothesis." A commonly used random null model is generated by randomizing the global topology of a network while preserving local node statistics, most importantly the graph's degree sequence.
The following analysis aims to investigate how the architecture of the network impacts on the controllability, comparing the results of random networks with the biological network of C. elegans, previously illustrated.

4.3.1 *Erdös-Rényi*

First, we generated an ensemble of directed networks, whose architecture is typical of *Erdös-Rényi graphs* with variable parameter p, i.e. the probability to create an edge between nodes.
According to p, the measures characterizing the network change, such as the average degree, while the number of nodes is fixed to 131, as well as C. elegans frontal ganglia neurons.
For example, in 4.14, it is provided a graphical representation of two networks with different p, and therefore different distribution of edges.

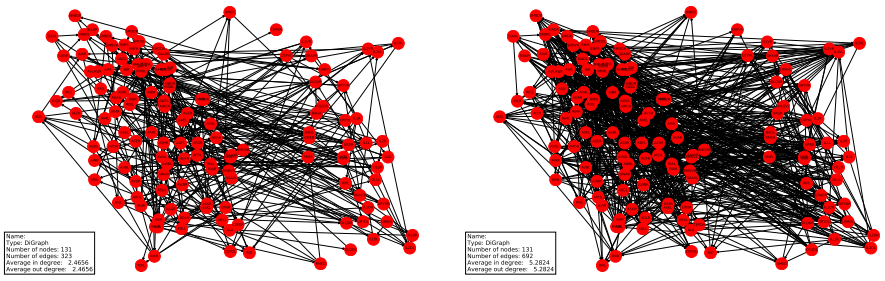


FIGURE 4.3 – Two networks belonging to the same ensemble of Erdös-Rényi digraphs. On the left the case $p = 0.02$, on the right $p = 0.04$.

As seen in 3.3.1, the equation 3.1 tell us the distribution of the density n_D with respect to the average degree of the networks belonging to the ensemble. For directed Erdös-Rényi networks, both $P(k_{in})$ and $P(k_{out})$ follow a Poisson distribution, in the thermodynamic limit:

$$P(k_{in}) = P(k_{out}) = e^{-<k_{in}>} \frac{< k_{in} >^k}{k!}$$

In this case, it holds:

$$G(x) = H(x) = \widehat{G}(x) = \widehat{H}(x) = e^{-<k_{in}>(1-x)}$$

From the equations which relates w_1 to w_2:

$$w_1 = e^{-<k_{in}>(1-w_2)} \qquad w_2 = 1 - e^{-<k_{in}>w_1}$$

and the self-consistent equation for w_1

$$w_1 = e^{-<k_{in}>e^{-<k_{in}>w_1}}$$

it turns out the expression of 3.1 for Erdös-Rényi digraphs:

$$n_D = w_1 - w_2 + < k_{in} > w_1(1 - w_2) \tag{4.1}$$

In [9], it has been estimated the asymptotic behaviour of n_D.
Indeed, as $< k >\gg 1$, it turns out:

$$\begin{aligned} w_1 &\sim e^{-<k_{in}>} \qquad w_2 = 1 - e^{-<k_{in}>w_1} \\ n_D &\sim e^{-<k_{in}>} - < k_{in} >^2 e^{-2<k_{in}>} \end{aligned} \tag{4.2}$$

Thus, it has been compared the analitycal results with the ensemble of simulated networks. We compared the distribution of n_D with respect to $< k_{in} >$ of the simulated networks with both the asymptotic behavior deduced in 4.2 and the numerical solution of 4.1. These comparisons are illustrated in Figure 4.4 and in Figure 4.5.

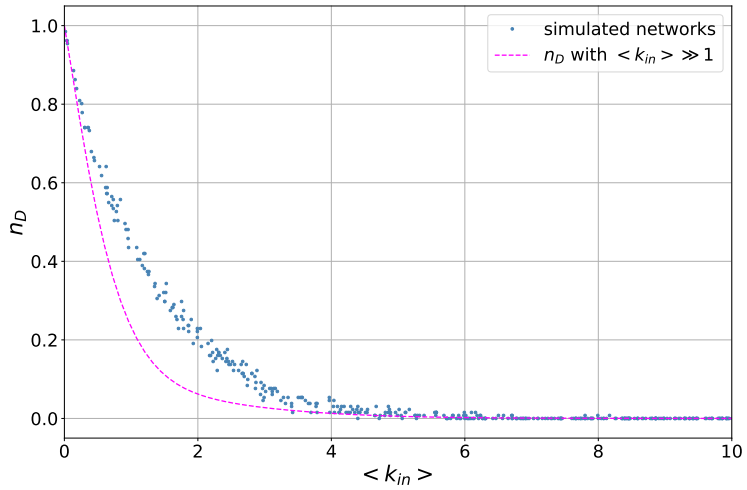


FIGURE 4.4 – The distribution of the ratio $n_D = N_D/N$ of the simulated networks (blue dots) has been compared with the asymptotic solution 4.2, that well fits the data in the $< k_{in} >\gg 1$ regime.

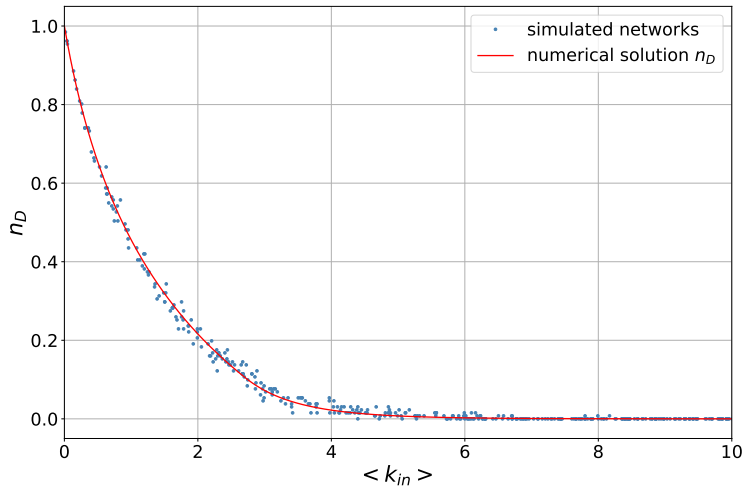


FIGURE 4.5 – The distribution of n_D of the simulated networks (blue dots) has been compared with the numerical solution of 4.1, with the self-consistent system of equations.

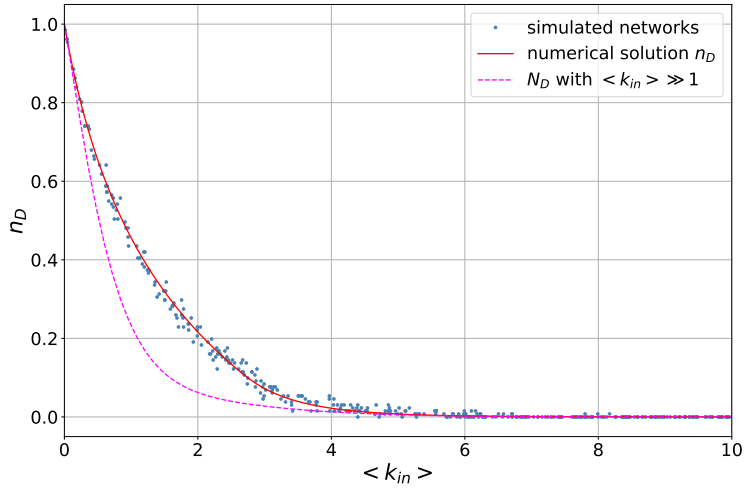


FIGURE 4.6 – Both the numerical solution and the asymptotic behavior are compared with the ensemble of simulated networks.

In order to compare more precisely the controllability of Erdös-Rényi networks, with the one of C. elegans, another ensemble has been simulated as follow.
The parameter p has been generated according to a Gaussian distribution, centered in $\bar{p} = \frac{<k_{in}>}{N-1}$. This sample aims to reproduce Erdös-Rényi networks with average degree compatible with the one of the nematode.

In Figure 4.7 is presented the behavior of N_D with respect to the average in-degree, while in 4.8 it is provided a graphical representation of the incidence of networks in the sample which are controllable with a certain number of driver nodes.

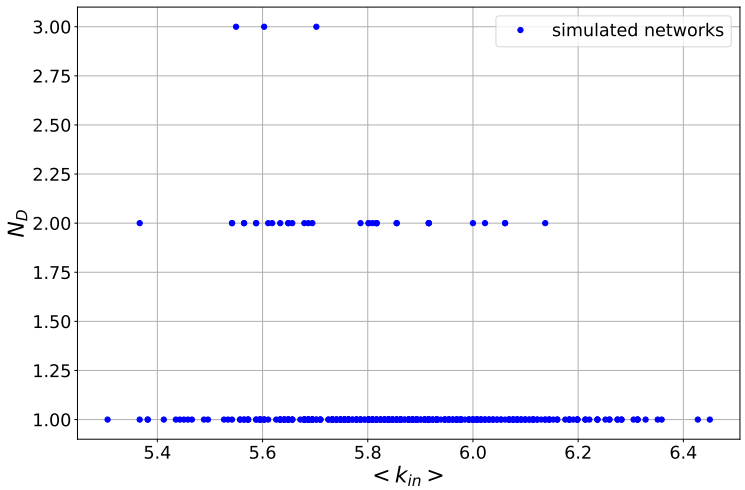


FIGURE 4.7 – The distribution of N_D with respect to the Gaussian distibuted average degree.

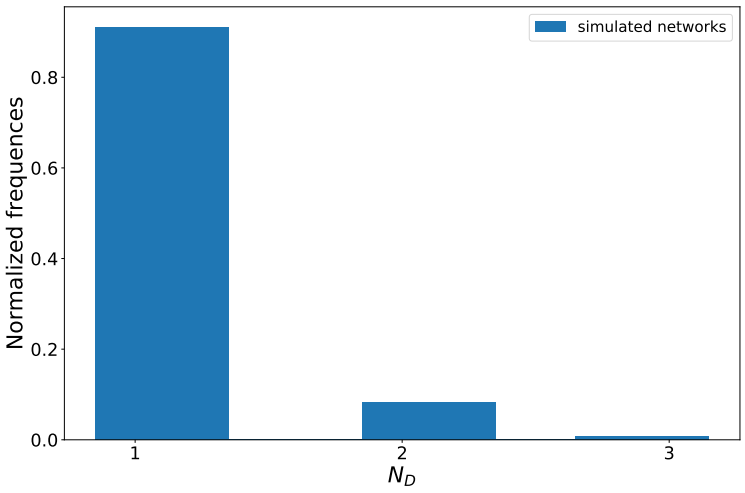


FIGURE 4.8 – The occurrence of the cardinality of the minimum set of driver nodes among the sample of Erdös-Rényi networks.

Taking into consideration other network measures, the trend of the driver nodes required to control the entire network is visualized in Figure 4.9 and

in Figure 4.10, with respect to the clustering coefficient and the path length, respectively.

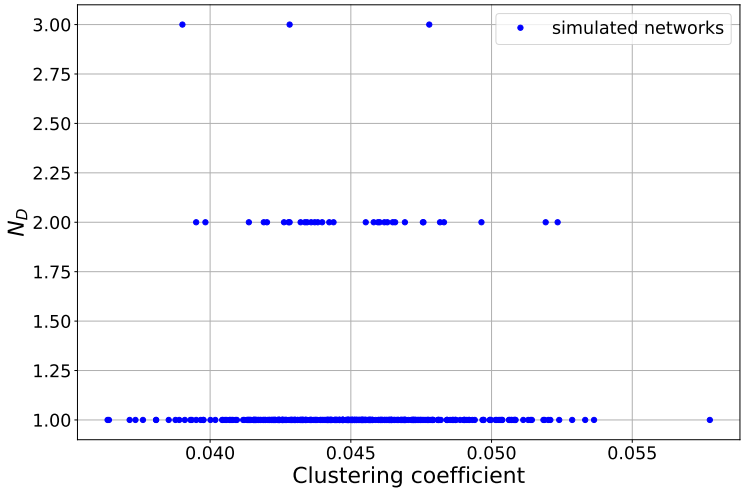


FIGURE 4.9 – The distribution of N_D with respect to the clustering coefficient.

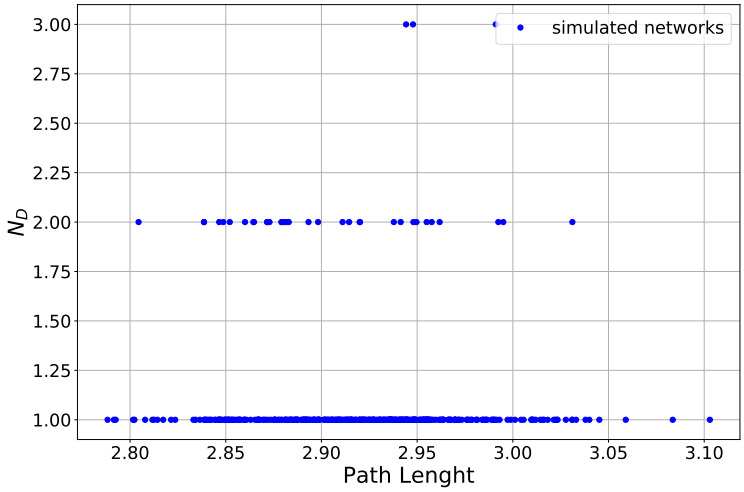


FIGURE 4.10 – The distribution of N_D with respect to the path length.

In [17], it is provided another way to generate Erdös-Rényi networks: instead of the parameter p, it is possible generate networks tuned by the parameter m, which quantifies the number of edges in the networks.
Setting $m = 764$, the sample of simulated network is characterized by average degree equal to the one of C. elegans.
Similarly to the ensamble tuned by the Gaussian distributed p, in 4.11 is represented the frequency (normalized among the sample) of networks controllable with a precise number of driver nodes, in 4.12 and 4.13 the behavior of N_D as function of the clustering coefficient and the path length, respectively.

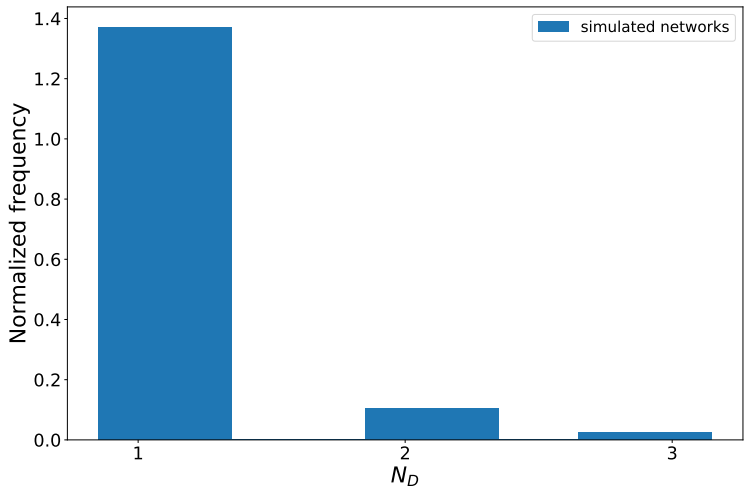


FIGURE 4.11 – The occurrence of the cardinality of the minimum set of driver nodes among the sample of Erdös-Rényi networks with fixed average degree.

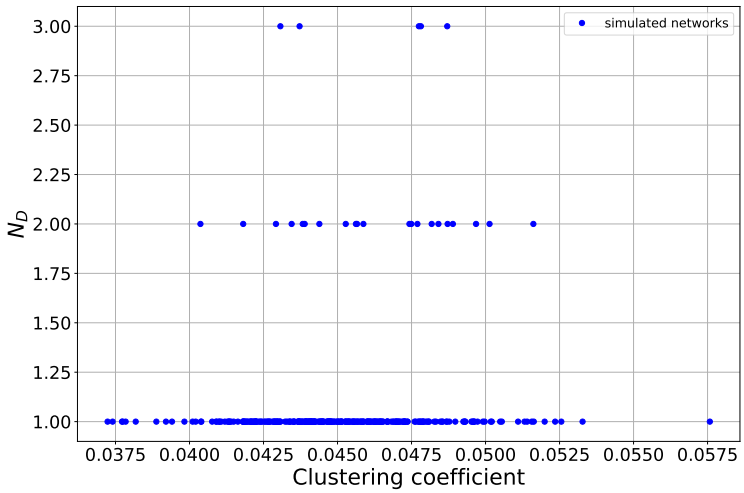


FIGURE 4.12 – Fixed the average degree, the distribution of N_D with respect to the clustering coefficient.

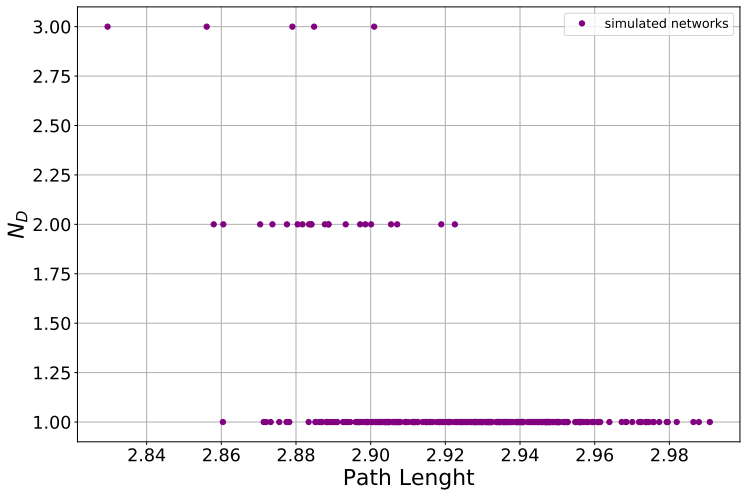


FIGURE 4.13 – Fixed the average degree, the distribution of N_D with respect to the path length.

4.3.2 *Scale free networks*

As seen in Chapter 2, the scale free networks are those characterized by power-law degree distribution.
The *static model* is one of the used model to generate *scale free* networks through a control parameter, according to which the degree exponent is determined. In this case, it has been chosen the same control parameter in order to determine the in-degree and the out-degree distributions, i.e. $\alpha_{in} = \alpha_{out} = \alpha$
There are N vertices in the system from the beginning, which are labelled by an integer $i = (1, ..., N)$. An expected weight $p_i = i^{-\alpha}$ is assigned to each node, with α the control parameter in the range $[0, 1)$. Next, we select two different vertices (i, j) with probabilities equal to normalized weights $p_i / \sum_k p_k$ and $p_j / \sum_k p_k$, respectively, and add an edge between them unless one exists already. This process is repeated until m edges are made in the system. Then the mean degree is $< k_{out} >=< k_{in} >= m/N$.

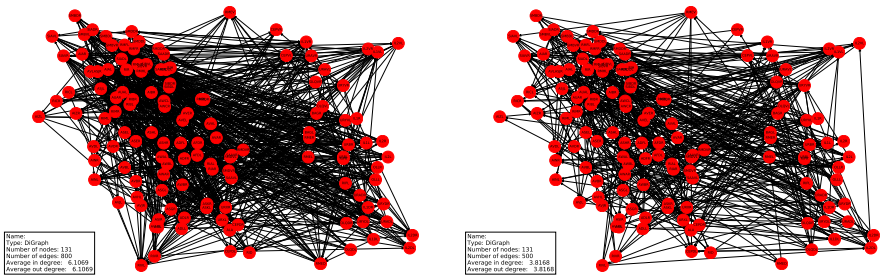


FIGURE 4.14 – Artificial networks with scale-free architecture. The number of nodes is fixed to 131 and random parameters are γ and M (the number of edges).

Since edges are connected to a vertex with frequency proportional to the weight of that vertex, the degree at that vertex is given as

$$\frac{k_i^{in}}{< k_{in} >} \sim \frac{1-\alpha}{N^{1-\alpha} i^{\alpha}}$$

Defining $\gamma = \frac{1+\alpha}{\alpha}$, it follows that the degree distribution follows the power-law:[24]

$$P(k_{in}) = k_{in}^{-\gamma} \tag{4.3}$$

Actually, 4.3 is not normalized, and it can not be a distribution. However, introducing a constant of normalization ζ and the exponential cutoff $e^{-k/\kappa}$

$$P(k_{in}) = \zeta k_{in}^{-\gamma} e^{-k_i n/\kappa} \tag{4.4}$$

the 4.4 is normalizable for any γ

It is useful to remark that the case $\alpha \to 0$ is equivalent to the Erdös-Rényi random model.

Once the degree distribution is defined, it is possible to particularized the distribution of n_D.

Indeed, for $\kappa \to \infty$, the generating functions and the self-consistent equations are then given by:

$$G(x) = \tilde{G}(x) = \frac{1}{\alpha} E_{1+\frac{1}{\alpha}}[(1-x) < k_{in} > (1-\alpha)]$$

$$H(x) = \tilde{H}(x) = \frac{1-\alpha}{\alpha} E_{\frac{1}{\alpha}}[(1-x) < k_{in} > (1-\alpha)]$$

$$w_1 = H(\widehat{w}_2) = \tilde{H}(x) = \frac{1-\alpha}{\alpha} E_{\frac{1}{\alpha}}[(1-\widehat{w}_2) < k_{in} > (1-\alpha)]$$

$$\widehat{w}_2 = 1 - \tilde{H}(1-w_1) = 1 - \tilde{G}(x) = \frac{1}{\alpha} E_{1+\frac{1}{\alpha}}[w_1 < k_{in} > (1-\alpha)]$$

where $E_n = \int_1^\infty e^{-xy} y^{-n} dy$ is the exponential integral.
In [9], it is provided the asymptotic behavior for $n_D(< k_{in} >, \gamma)$:

$$n_D = G(\widehat{w}_2) + \tilde{G}(1-w_1) - 1 + < k_{in} > w_1(1-\widehat{w}_2) \sim e^{-<k_{in}>(1-\frac{1}{\gamma-1})} \tag{4.5}$$

Following the same strategy used for the Erdös-Rényi case, we generated an ensamble of artificial networks, tuning the power-law exponent γ and the number of edges.
The parameter γ has been estimated through a linear fit of the the degree distribution in logarithmic scale.

We compare the distribution of the driver nodes density produced by the artificial networks with the asymptotic behavior described by 4.5.
In order to find the n_D distribution, first we simulated a sample of scale-free digraphs with $\gamma = 2.38 \pm 0.02$ and the number of edges as free parameter, as shown in Figure 4.15.
As previously mentioned, for great γ the behavior of the ensemble coincides with the one of Erdös-Rényi, as illustrated in figure Figure 4.16, where the power-law exponent generating the networks is $\gamma = 4.5 \pm 0.9$.

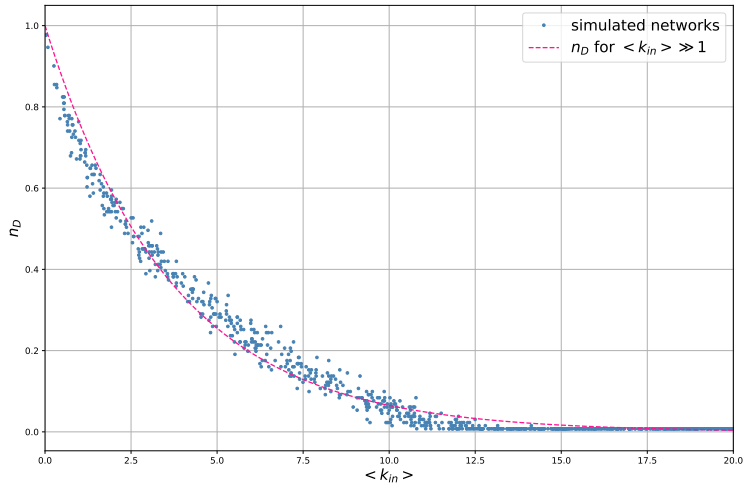


FIGURE 4.15 – The distribution of n_D for the sample of the networks with fixed γ (blue dots) and the expected asymptotic behavior provided by 4.5 .

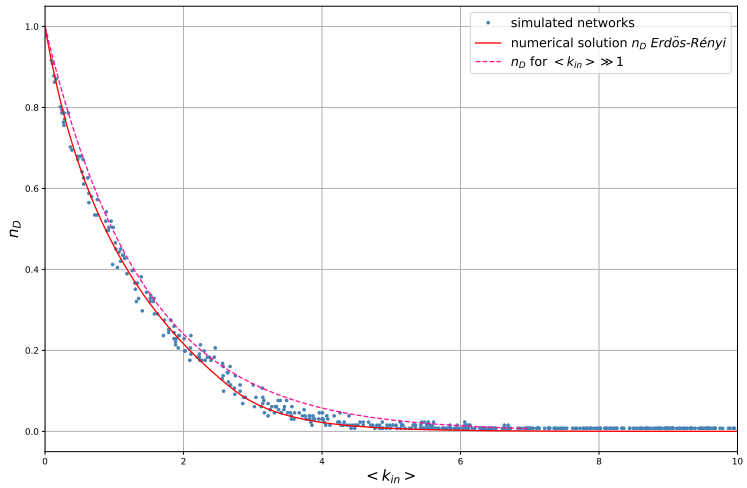


FIGURE 4.16 – The distribution of n_D for the sample of the networks with fixed high γ (blue dots) is compared with the expected analytical behavior of both asymptotic scale-free networks (fuchsia) and Erdös-Rényi ones (red lines).

In order to investigate the n_D distribution with respect to γ, we simulated a sample of digraphs with 131 nodes and 764 edges, as the C. elegans network, while γ is a free parameter.
However, there is a significantly discrepancy between the simulated result and the analytical one, as shown in figure 4.17. That fact can be explained as follow: the equation 4.5 has been deduced with the hypothesis of thermodynamics limit. With this assumption, a critical exponent occurs, i.e. for $\gamma_c = 2$ all the nodes are required to control the entire systems.
Indeed, simulating a sample of digraphs with $N = 100000$ nodes, in figures 4.18 the effect of finite size starts to vanish.

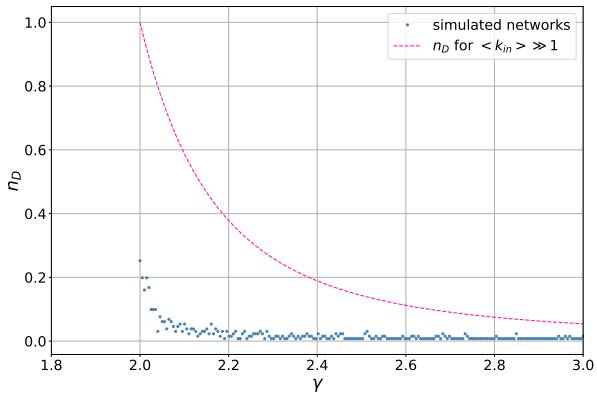


FIGURE 4.17 – The distribution of n_D for the sample of the networks with fixed average degree (blue dots) compared with the expected asymptotic behavior provided by 4.5. Here we have strong effect of finite size.

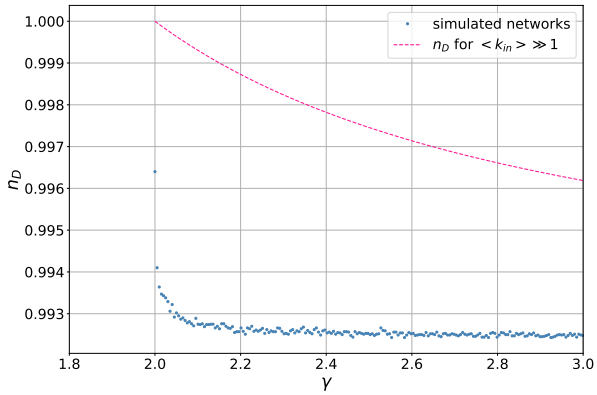


FIGURE 4.18 – A sample of scale-free networks with 100000 nodes and fixed average degree, here the finite size effect starts to be less important on the system.

It is interesting to analyze the cluster coefficient and the shortest path length as function of γ and fixed average degree.
For each gamma, we simulated a sub-sample of scale-free networks and we deduced the mean value and variance of both measures among each sub-sample. These results are shown in Figure 4.19 and in Figure 4.20
The aim is to deuce a γ which could reproduce digraphs comparable with the empirical network.
In figures 4.21 and 4.22, it is illustrated the the distribution of N_D with respect to the clustering coefficient and the path length, respectively.

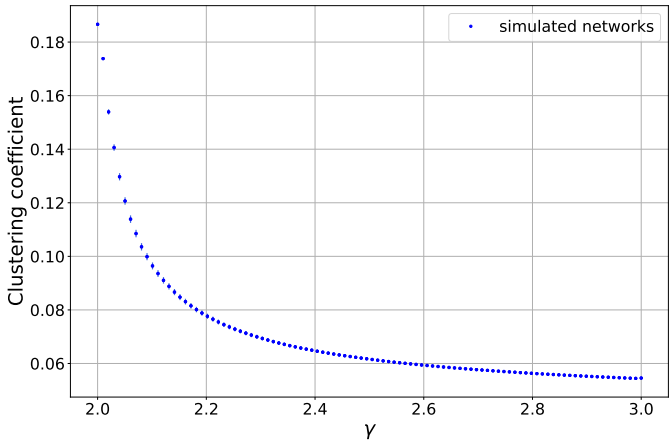


FIGURE 4.19 – Averaged clustering coefficients with respect to γ.

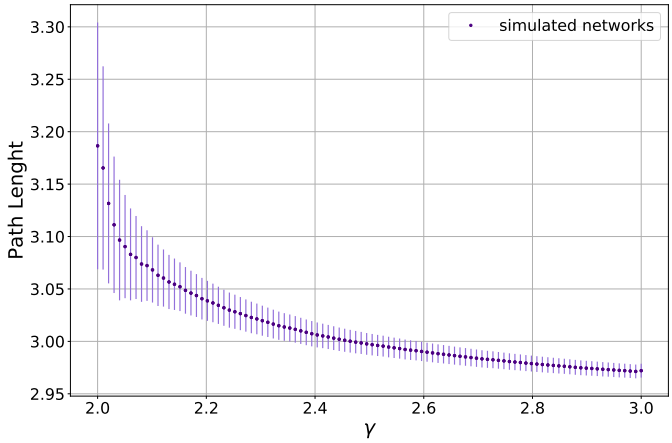


FIGURE 4.20 – Averaged path lengths with respect to γ.

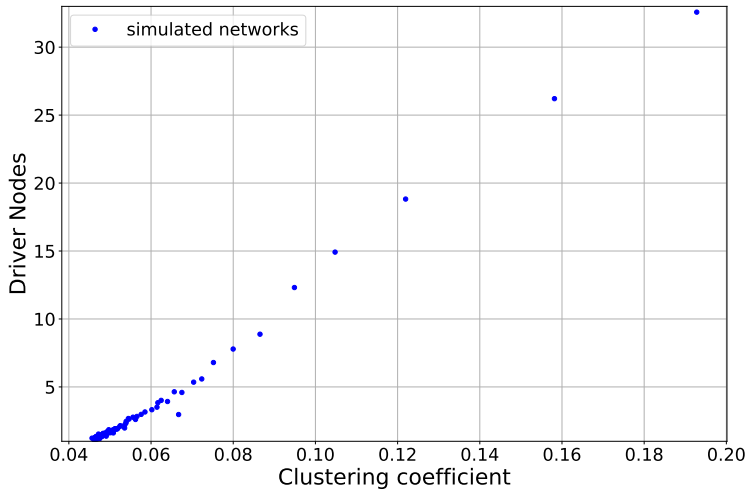


FIGURE 4.21 – The distribution of N_D with respect to the clustering coefficient.

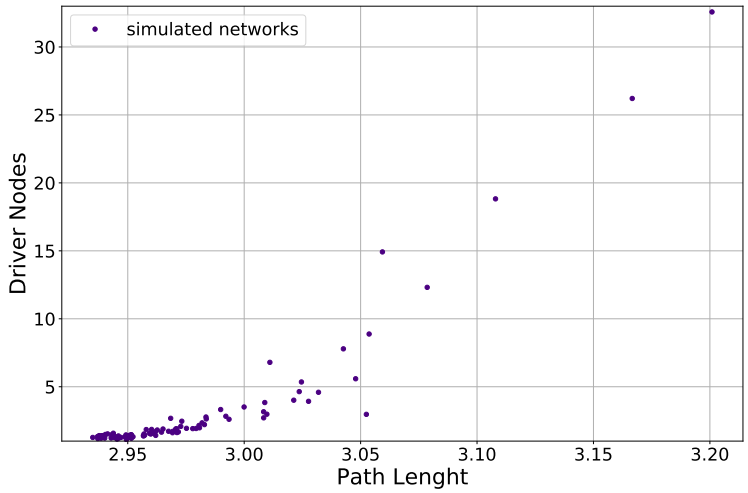


FIGURE 4.22 – The distribution of N_D with respect to the path length.

Similarly, for the Erdös-Rényi networks, we aim to compare the controllability of scale-free networks with the one of the worm brain.
Therefore, we generated a sample of scale-free digraphs tuned by a Gaussian distributed γ and fixed average degree (equal to the one of C. elegans).
Figure 4.23 shows the trend of driver nodes with respect to γ. On the other hand, 4.24 provides a graphical representation of the normalized occurrence of the cardinality of MSDN among all the networks.

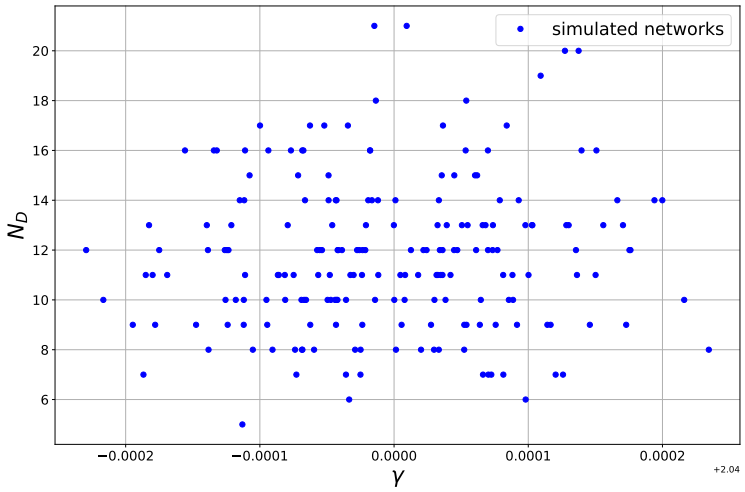


FIGURE 4.23 – The distribution of N_D according to the Gaussian distributed γ.

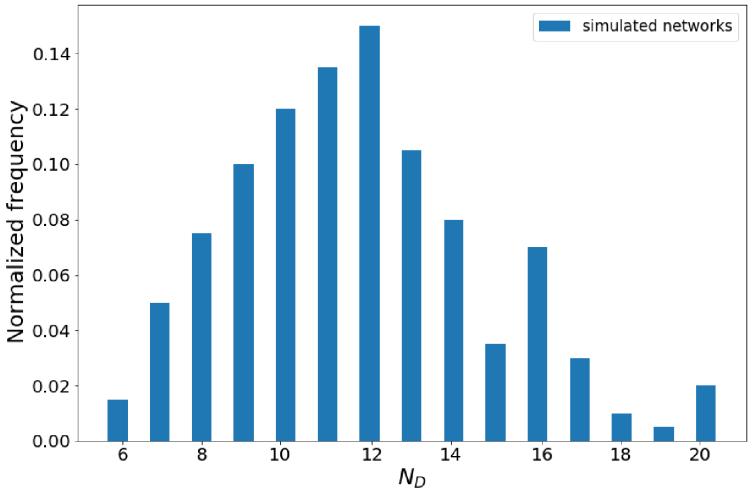


FIGURE 4.24 – Normalized frequency of the number of driver nodes required to control the network among the sample.

4.4 THE IMPORTANCE OF DIRECTION

As previously illustrated in Section 1.3, once we have an LTI system described by the structured matrices **A** and **B** it is possible to apply the tools provided by structural controllability. According to this last one, the structured system can be represented as a digraph $G(\mathbf{A}, \mathbf{B})$, whose directed edges correspond to the non zero entries of **A** and **B**, and its controllability is related to the absence of dilations and inaccessible nodes.
Furthermore, as seen in Section 3.3, we are interested in the topology of the digraph $G(\mathbf{A})$, the one representing the internal interaction of the system without external inputs. Indeed, the maximum matching of the directed graph $G(\mathbf{A})$ provides a powerful tool in order to identify the minimum set of nodes needed to fully control the system: those are the ones that are not saturated by the maximum matching.

All these tools involve directed interaction among the elements. A natural question is whether the directionality of the edges impacts the system in terms of its controllability.
To construct a network equivalent to an undirected graph starting from a digraph, the simplest way is adding to each edge of the digraph its counterpart, i.e. a new edge with the same end-vertices of the original link but with opposite direction, an example is provided in figure 4.25.

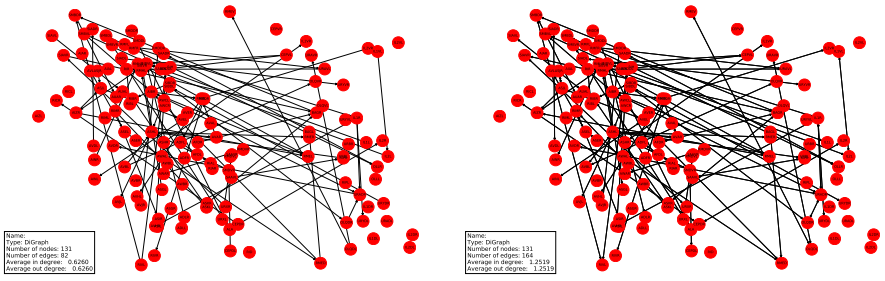


FIGURE 4.25 – On the left a random directed network, on the right its undirected analog network with the further inverse connectivity, i.e. all edges are bidirected.

For example, the controllability of the two networks illustrated in 4.25 is shown in 4.26

Therefore, in order to investigate the direction related issue, we have simulated a sample of random digraphs and its n_D distribution has been compared with the one derived from a sample of equivalently-undirected networks, generated as previously explained.

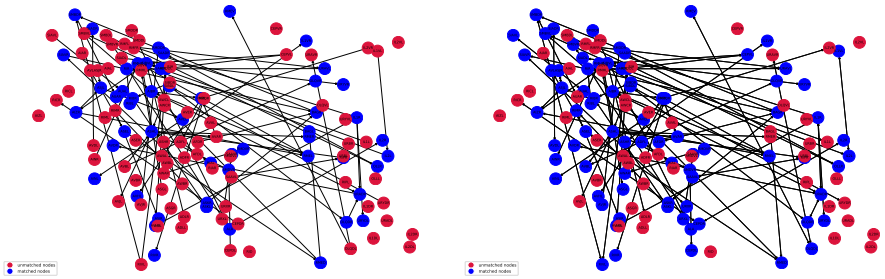


FIGURE 4.26 – Red dots are the driver nodes required to fully control the network. the networks are the same represented in 4.25 .

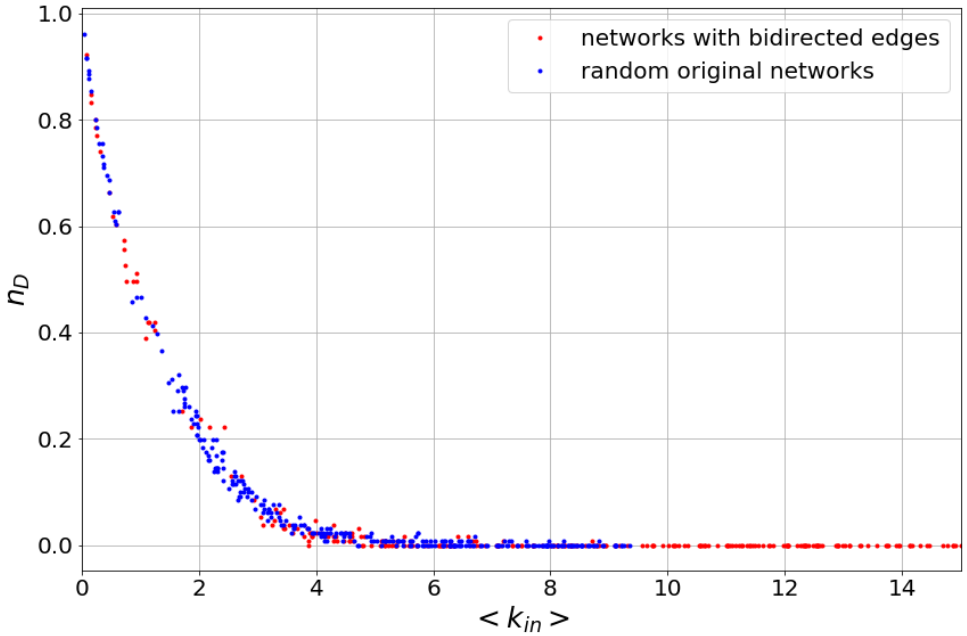


FIGURE 4.27 – Original networks (blue dots) and networks whose edges are the ones of the original networks, but having bioriented directions (red todt).

We noticed two regimes: for great average degree of the networks having directed interaction or not does not impact on controllability, for small average degree the controllability is sensitive of directionallity.
In the following table, are reported some representative points of directed networks and their counterpart equivalently-undirected ones.

Original Network		Bidirected connectivity		
N_D	k_{in}	N_D	k_{in}	$\Delta(N_D)$
55	1.099	81	0.550	26
33	1.664	68	0.832	35
13	2.611	48	1.305	35
2	4.031	27	2.023	25
1	5.985	8	3.023	7
1	6.443	12	3.275	11
1	7.282	5	3.664	4
1	8.137	2	4.137	1
1	11.603	1	5.924	0
1	11.725	1	6.015	0
1	13.786	1	7.015	0
1	15.573	1	8.061	0
1	17.389	1	9.046	0

CHAPTER 5

Conclusion

Control theory has a long and vibrating history in engineering and mathematics. It addresses a fundamental and ambitious question: how to control a system's behaviour, i.e. how and whether it's possible to push the output of a dynamical system to a desired final state through suitable external inputs.
Tools of control and controllability have been recently applied in the study of complex networks. The aim is to characterize networks, representing a certain complex system (brain system for example), in terms of their controllability.
In this work, we investigate, it has been investigated the problem concerning the identification of a minimum set of nodes required to control the entire networks, known as *minimum input problem*.
Graph theory, in particular maximum matching, provides a powerful approach to address the minimum input problem when the systems are characterized by linear and time-invariant dynamics.
Indeed, the minimum set of nodes required to control the system, namely the driver nodes, is identified by the maximum matching of the network: the nodes which are not saturated by the maximum matching M are the driver nodes. Eventually, if the maximum matching is a perfect one, i.e. all the nodes are saturated, it is necessary at least one node to perform the fully controllability of the network.
Using a numerical approach, the results have been compared with analytical results, provided by literature, mainly from [9].

Motivated by the aim of applying the control principles on a real- world network, we investigated the Caenorhabditis elegans brain. By a biological point of view, this nematode represents an interesting animal due to the rich-

ness of its biological functions with a relative simplicity of its body structure, offering a great compromise between complexity and tractability.
For all these reasons, a number of studies on C. elegans have been carried out. For example, C. elegans is the only animal whose complete connectome of nearly 300 neurons has been mapped. This fact makes Caenorhabditis elegans the proper test bench in order to apply control tools on its brain network.
We have analyzed and characterized with respect to its controllability the sub-network, representing frontal ganglia neurons of C. elegans. Indeed, the minimum set of driver nodes (MSDN) needed to control the system is constituted by 12 neurons.

This work aims to investigate how the topology of the networks can impact their controllability.
Indeed, we investigate two typologies of random networks. The first one is represented by Erdös-Rényi networks, which randomness is modelled through the probability p of creating an edge between two vertices or through the number of edges M. The second one describes static scale-free networks, whose degree distribution follows a power-law. The random parameters of these networks are the number of edges of the graph and the power-law exponent γ.
Both the typology of random networks present that controllability depends from the average degree of the networks.

A sample of Erdös-Rényi digraphs has been simulated and compared with the analytical expected result: in 4.4, 4.5 and 4.6 it is provided a graphical representation of the density of driver nodes (both simulated and analytically expected) distributions.
In order to compare the random topology with the real-world one, we analyzed another sample of Erdös-Rényi digraphs having similar, or even equal, average degree of the nematode's brain network.
The surprisingly result is that in the almost case the system is fully controllable with the smallest possible MSDN: indeed, from figures 4.8 and 4.11, it is clear that almost all networks of the sample require just one driver node to achieve the fully controllability.
Figure 4.12 and 4.13 show how, fixing the number of edges equal to the C.elegans case, the number of driver nodes changes with respect to two typical measures characterizing networks: the cluster coefficient and the shortest path length.
We found that that Erös-Rényi digraphs with fixed average degree $< k >$ present that both the shortest path length and the cluster coefficient are smaller than the real world system.

The second type of network topology that we investigated is the one with power-law degree distribution, known as *scale-free*.

In this case, there occur two random parameters performing the network architecture: the number of edges and the power-law exponent γ. In **??** and 4.17, it was compared the simulated result with the analytic expected curve. In 4.17 [*it*] is interesting to note that there is a significant discrepancy between simulated networks controllability and analytical expectation. This phenomenon is motivated by the fact that all the analytical results are found under the thermodynamic limit hypothesis: indeed, simulating a sample of scale-free networks with $N = 100000$ nodes, the effect of finite size starts to vanishes, as shown in 4.18.
Therefore, we found that the effect of finite size has a great impact on controllability, making the system strongly easier to control.

Fixed the number of edges, equal to the biological case, in order to determine which γ can describe networks similar to the one of C. elegans, we investigated the shortest path length and cluster coefficient behavior with respect to γ.
Figure 4.23 provides a visual representation of networks which are simulated according to a Gaussian distributed γ.
The occurrence of the driver nodes within the sample is represented in 4.24: the distribution is peaked in $N_D = 12$, equal to the real-world case.

It is interesting that the empirical network requires a greater number of driver nodes to control the whole system than random networks Erdös-Rényi. On the other hand, the scenario provided by the scale-free is more variable: indeed, a further development can be represented by a more precisely estimation of *gamma* describing the nematode's brain system, for example using the entire worm brain network and using other measures which characterize the network topology.

The harder controllability of empirical digraph, than the one of random digraphs, is certainly related with its architecture: this analysis suggest that small-world features and modularity of a real brain system affect the controllability of the system itself, i.e. the fact that biological systems have a topology which presents a great complexity makes controllability more difficult than the case of networks characterized by randomness.

The last aspect investigated is the role of directed edges.
Indeed, the tool of maximum matching in order to characterize the controllability has been developed for directed graphs. A natural question is whether the directionality can affect the controllability.
A simple way to yield an undirected graph starting from a digraph is to add to the edges their inverse, i.e. the same edge with opposite direction. With a great average degree, the directionality doesn't influence the resulting controllability, but, with small average degree, having directed edges or bidirected ones

has a great impact in terms of controlling the system, as we show in Section 4.4

However, having a system with directed interactions is a consequence of a further upstream assumption: as illustrated in Chapter 1, the controllability framework is developed for linear systems, as well for systems linearized around a fixed point. Under the linearized hypothesis the systems are linear and time-invariant, namely LTI systems, and the relation between controllability and maximum matching occurs only whit this type of systems.

Indeed, the current idea of applying network control theory to neuroscience is based on the hypothesis that the dynamics of neuronal activity at brain resting state can be described via a set of differential equations linearized around the resting state. The linearization assumption is crucial and actually strong because brain networks are characterized by deeply non linear dynamics and furthermore it only provides information about controllability in the neighborhood of that state.
Therefore, in order to deeply understand brain networks and the further development, it is fundamental to approach the problems in terms of non-linear dynamics, and to enlarge the controllability framework from both analytical and numerical point of view, looking for new paradigms and techniques.

APPENDIX A

Maximum matching and Augmenting Path

Given a matching M in a graph G, an M-alternating path is a path that alternates between edges in M and edges not in M. An M-alternating path is an M-augmenting path if the two endpoints of the path are unsaturated by M. If G has an M-augmenting path P, one can obtain a new matching M' with one more edge by replacing the edges of M in P with the other edges of P. Thus the following fact holds.

Theorem A.1 – M is not a maximum matching in G if G has an M-augmenting path.

On the other hand, as shown in [6], it can be proved that if M is not a maximum matching in G then G has an M-augmenting path. Let M and M' be two matchings in $G = (V, E)$. The *symmetric difference* of two matchings $M\Delta M'$ is the graph with the vertex set V and the edge set consisting of all edges appearing exactly one of M and M'. Let S be the set of edges which is contained in M but not in M' and let S' be the set of edges which is contained in M' but not in M. Then $M\Delta M' = S \cup S'$

LEMMA A.1 – Every connected component of the symmetric difference of two matchings is a path or an even cycle.

Proof Let M and M' be two matchings in a graph G and let $H = M\Delta M'$. We first claim that the maximum degree of H, (i.e. the maximum value among the degrees of all the vertices of H, denoted by $\Delta(H)$), $\Delta H \leq 2$. At most one edge from a matching is incident to a vertex of G. Since M and M' are both matchings, at most two edges can be incident to a vertex of H . Hence $\Delta(H) \leq 2$. Since $\Delta(H) \leq 2$, every connected component of H is either a path or a cycle. Therefore, it is remained to show that if a connected component of H is a cycle C, then C is an even cycle. One can observe that the edges on C alternate between edges of M and M' . Then to close the cycle C, it must have equal number of edges from M and M' , and hence C is an even cycle.

LEMMA A.2 – A matching M in a graph G has an M-augmenting path if M is not a maximum matching in G.

Proof Let M' be a matching in G larger than M. We will prove that G has an M-augmenting path. We give a constructive proof. Let $H = M\Delta M'$. Every connected component of H is either a path or a cycle. If every component of H is a cycle then $|M| = |M'|$, a contradiction. Since $|M'| > |M|$, H has a connected component which is a path P containing more edges of M' than of M. Then P starts with an edge of M' and also ends with an edge of M' . Since P contains edges from M and M' alternately, P is an M-augmenting path.

From A.1, A.1 and A.2, it naturally turns out:

THEOREM A.2 (Berge Theorem) – A matching M in a graph G is a maximum matching in G if and only if G has no M-augmenting path.

APPENDIX B

Hopcroft-Karp Algorithm

It is now presented the *Hopcroft-Karp Algorithm* to find a maximum matching in a bipartite graph with $|V| = n$ and $|E| = m$, such as the one in B.1.

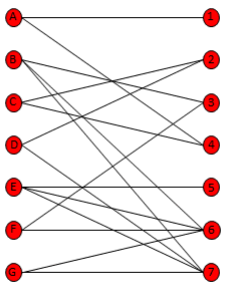


Figure B.1 – a bipartite graph with two sets: on the left the nodes labelled with letters, on the right the ones labelled with numbers.

The basic idea of the algorithm is:

1. Initialize Maximal Matching M as empty.
2. While there exists an augmenting path p, remove matching edges of p from M and add not-matching edges of p to M.
3. Return M.

We need to find an augmenting path (A path that alternates between matching and not matching edges, and has free vertices as starting and ending points). Once we find alternating path, we need to add the found path to

existing Matching. Here adding path means, making previous matching edges on this path as not-matching and previous not-matching edges as matching.
The idea is to use the Breadth First Search strategy to find augmenting paths: according to BFS algorithm, it stars from a vertex and explores all of the neighbor nodes, "marking" them as visited.
Since BFS traverses level by level, it is used to divide the graph in layers of matching and not matching edges, building the alternating level graph rooted at unmatched vertices in the left side of B.1.
Then, it is used a Depth Fist Search Algorithm, or DFS, to find maximal set of vertex disjoint shortest-length paths joining auxiliar vertices S and T.
DFS is similar to BFS, but it explores as far as possible along each branch before backtracking.

For example, we can initialize a first initial matching such as:

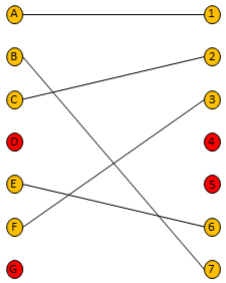


FIGURE B.2 – an initial matching: red nodes are unmatched, yellow nodes are matched.

The multilayer tree is built with BFS and DFS as follow:

S1: The unmatched nodes on the left side are the roots of the tree;
S2: The second layer is added with the adjacent nodes of the unmatched ones in the initial bipartite graph;
S3: The firth layer is linked with the second one through the edges occurred in the matching;
S4: The fourth layer is linked with the previous one through the edges not occurred in the matching;
S5: Auxiliary nodes S and T are linked to the layers of unmatched nodes;
S6: Find the shorter paths linking S and T;

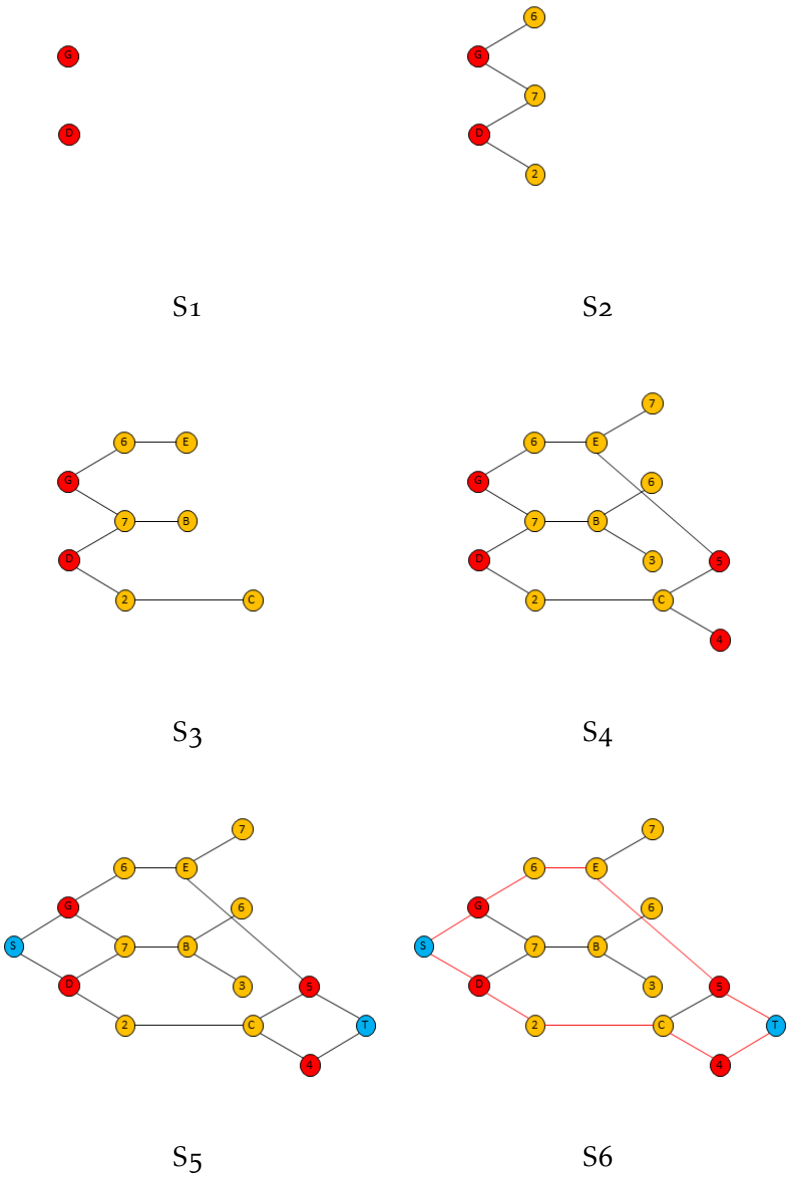
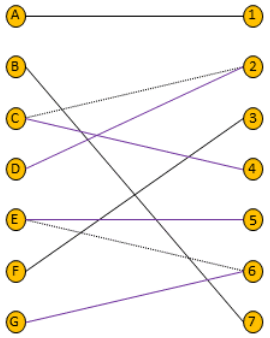
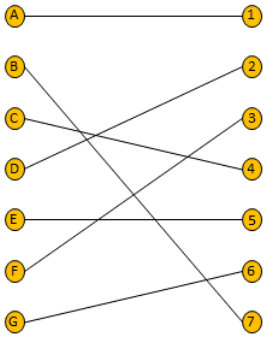


Figure B.3 – Breadth First Search and Depth First Search strategy.

Founded the shorter paths, between them and the initial matching is applied the symmetric difference in order to find a new larger matching.
The algorithm is reiterated till it is not possible to find an augmenting path.



Dashed edges are the removed ones from the initial matching, purple edges are the new edges added to the matching.



The resulting maximum matching: in this case all the nodes are matched, thus it is a perfect matching

Time Complexity

Both BFS and DFS has complexity $O(m)$ where m is the number of the edges.

LEMMA B.1 – After N iterations augmenting path must be at least length N

LEMMA B.2 – Given M and M', the current and the maximum matching respectively, and l the length of the shortest augmenting path, it holds

$$|M'| - |M| \leq \frac{|V|}{l}$$

Using B.1 and B.2, we can say that after $\sqrt{n}$ iterations augmenting path must be at least length $\sqrt{n}$ and the number of th remaing augmenting path after $\sqrt{n}$ iteration is $n/\sqrt{n} = \sqrt{n}$.
While, after $\sqrt{n}$ iterations, the algorithm can be iterate for a maximum of $\sqrt{n}$ times before to not have more augmenting paths.
Thus, at most $2\sqrt{n}$ iterations can occur, for a time complexity of $O(\sqrt{n})$.
Therefore, the time complexity of Hopcroft-Karp algorith, is overall of $O(m\sqrt{n})$.

Bibliography

[1] N. Goldenfeld. "Simple Lessons from Complexity". In: *Science* 284 (1999), pp. 87–89.

[2] Albert-László Barabási. *Network Science*. Cambridge University Press, 2016.

[3] Mark E J Newman. *Networks an introduction*. Oxford Oxford University Press, 2018.

[4] Jacob Levy Moreno. *Who shall survive?: A new approach to the problem of human interrelations.* Nervous and mental disease publishing co, 1934, p. 38.

[5] Danielle S Bassett and Olaf Sporns. "Network neuroscience". In: *Nature neuroscience* 20.3 (2017), p. 353.

[6] Md. S. Rahman. *Basic graph theory*. Springer, 2017.

[7] Maarten Van Steen. *Graph theory and complex networks : an introduction*. Maarten Van Steen, 2010.

[8] David G Luenberger. *Introduction to dynamic systems; theory, models, and applications*. Tech. rep. 1979.

[9] Yang-Yu Liu and Albert-László Barabási. "Control principles of complex systems". In: *Reviews of Modern Physics* 88 (2016).

[10] R. E. Kalman. "Mathematical Description of Linear Dynamical Systems". In: *Journal of the Society for Industrial and Applied Mathematics Series A Control* 1 (1963), pp. 152–192.

[11] Ching-Tai Lin. "Structural controllability". In: *IEEE Transactions on Automatic Control* 19 (1974), pp. 201–208.

[12] Olaf Sporns. *Networks of the Brain*. MIT press, 2010.

[13] Ed Bullmore and Olaf Sporns. "Complex brain networks: graph theoretical analysis of structural and functional systems". In: *Nature reviews neuroscience* 10.3 (2009), p. 186.

[14] Mikail Rubinov and Olaf Sporns. "Complex network measures of brain connectivity: uses and interpretations". In: *Neuroimage* 52.3 (2010), pp. 1059–1069.

[15] Olaf Sporns. "Structure and function of complex brain networks". In: *Dialogues in clinical neuroscience* 15.3 (2013), p. 247.

[16] Gorka Zamora-López, Changsong Zhou, and Jürgen Kurths. "Cortical hubs form a module for multisensory integration on top of the hierarchy of cortical networks". In: *Frontiers in neuroinformatics* 4 (2010), p. 1.

[17] P Erdös and A. Rényi. "On Random Graphs. I". In: *Publicationes Mathematicae* 6 (1959), pp. 290–297.

[18] Duncan J Watts and Steven H Strogatz. "Collective dynamics of 'small-world'networks". In: *nature* 393.6684 (1998), p. 440.

[19] J. G. White et al. "The Structure of the Nervous System of the Nematode Caenorhabditis elegans". In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 314 (1986), pp. 1–340.

[20] Emma K Towlson et al. "The rich club of the C. elegans neuronal connectome". In: *Journal of Neuroscience* 33.15 (2013), pp. 6380–6387.

[21] Marcus Kaiser and Claus C. Hilgetag. "Nonoptimal Component Placement, but Short Processing Paths, due to Long-Distance Projections in Neural Systems". In: *PLoS Computational Biology* 2 (2006), e95.

[22] Rolf Kötter. "Online retrieval, processing, and visualization of primate connectivity data from the CoCoMac database". In: *Neuroinformatics* 2.2 (2004), pp. 127–144.

[23] Yoonsuck Choe, BH McCormick, and W Koh. "Network connectivity analysis on the temporally augmented C. elegans web: A pilot study". In: *Soc Neurosci Abstr*. Vol. 30. 921.9. 2004.

[24] K-I Goh, Byungnam Kahng, and Doochul Kim. "Universal behavior of load distribution in scale-free networks". In: *Physical Review Letters* 87.27 (2001), p. 278701.

[25] Chengyi Tu et al. "Warnings and caveats in brain controllability". In: *NeuroImage* 176 (2018), pp. 83–91.

[26] Emma K. Towlson et al. "Caenorhabditis elegans and the network control framework—FAQs". In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 373 (2018), p. 20170372.

[27] Olaf Sporns, Giulio Tononi, and Rolf Kötter. "The Human Connectome: A Structural Description of the Human Brain". In: *PLoS Computational Biology* 1 (2005), e42.

[28] Evelyn Tang and Danielle S. Bassett. "Colloquium : Control of dynamics in brain networks". In: *Reviews of Modern Physics* 90 (2018).

[29] Reinhard Diestel. *Graph Theory*. Vol. 173. 2012.

[30] László Lovász and Michael D Plummer. *Matching theory*. Vol. 367. American Mathematical Soc., 2009.

[31] Maarten Van Steen. "Graph theory and complex networks". In: *An introduction* 144 (2010).

[32] Douglas Brent West et al. *Introduction to graph theory*. Vol. 2. Prentice hall Upper Saddle River, NJ, 1996.

[33] Dieter Jungnickel and D Jungnickel. *Graphs, networks and algorithms*. Springer, 2005.

[34] Shi Gu et al. "Controllability of structural brain networks". In: *Nature communications* 6 (2015), p. 8414.

[35] Réka Albert and Albert-László Barabási. "Statistical mechanics of complex networks". In: *Reviews of modern physics* 74.1 (2002), p. 47.

[36] Cornelis J Stam and Jaap C Reijneveld. "Graph theoretical analysis of complex networks in the brain". In: *Nonlinear biomedical physics* 1.1 (2007), p. 3.