Politecnico di Torino

DET - Department of Electronics and Telecommunications

Master's Degree in Communications and Computer Networks

Opinion Dynamics on Co-Evolving Complex Networks

Supervisors:
Prof. E. Leonardi
Prof. M. Garetto

Student:
F. Galante

23 October 2020

To my parents

Contents

Introduction

The study of social phenomena identifies humans as the essential entity and attempts to describe human behaviour. Probably, the opinions an individual holds represent the most important factors behind human behaviour. Human beings take actions according to their beliefs therefore, the investigation of the mechanism of opinion formation assumes great interest in the study of social phenomena. This mechanism depends on a large number of variables and looking for a mathematical model to describe such a complex process might seem pretentious. However, many models have been proposed since the late 50's which have provided many insights into the process of opinion formation.

Opinion dynamics research originated in French in 1956 [1], in which a formal theory has been developed to investigate to what extent "the influence process in groups can be explained in terms of interpersonal relations". Since then various models have been proposed such as DeGroot model [2], considered as the classical model in opinion dynamics, and the voter model [3], which will be studied in more detail in the remainder. These two models can be regarded as the starting point of the models which will be discussed below. Other models worth mentioning which will note be discussed below are the Sznajd model [4] [5] and the majority rule model [6]. This research field gained more attention with the advent of social media. Individuals have access to a huge amount of information and social media allow for intense interaction. This is clearly beneficial however, "the exchange of opinions can lead to the propagation of counterfactual rumours and can even give rise to the formation of radicalized groups" [7]. Examples of such groups are the Flat Earth Society or the anti-vax movement. Investigating the process of collective opinion formation may reveal the conditions which lead to a beneficial outcome.

To study opinion dynamics a set of agents [1], which represent the population taken into consideration, is considered jointly with the degree of interaction among them. Each agent holds an opinion which is subject to changes when an interaction with the other agents occurs. Opinion formation is a complex process function of different aspects, such as individual predisposition to change mind, preconceived ideas, information exposure, positive and negative peer interaction, and many others. [8]. It appears to be natural to represent the opinion dynamics setting on a graph making the equivalence between agents and nodes and between agents' relationships and links. The network approach has contributed significantly in the understanding of complex systems. There exist many network models which are widely used to describe complex social

[1] Throughout this work the terms agent, individual and node are used interchangeably, similarly also the terms network and graph.

interactions such as the Erdos-Rényi (ER) graph, the small-world (SW) network and the scale-free (SF) network. Initially static graphs have been considered for the dynamic process. Lately, graphs whose structure varies dynamically are of interest also because they model well the user's behaviour on many social networks, where a user follows another if their interests are similar.

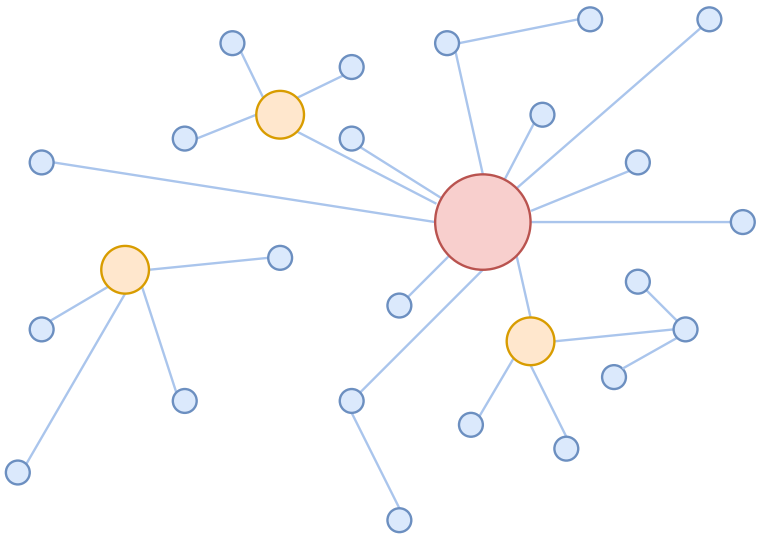


Figure 1: Schematic representation of a network, nodes with higher degree are represented with bigger circles and brighter colours.

This thesis work is organised as follows: in chapter 1 an eagle-eye view of opinion dynamics on complex network is provided, together with some classical models that have been studied in the literature. The voter model is introduced and in chapter 2 a series of voter-like models is brought to attention. At the end of the chapter the adaptive voter model proposed by Durrett et al. [9] is discussed. Chapter 3 is devoted to the description of the simulator that has been developed in c++ in order to study in depth the dynamics of the adaptive voter model. Particular attention is posed on the choice of the data structures and at the development of efficient random graph generators. A discussion of the phase transition that the model undergoes is given in chapter 4, where it is also studied the response of the model to random graphs different from the Erdős-Rényi, in particular to the Stochastic Block Model (SBM). The last chapter of this work aims at investigating the time complexity of variants of the model. In particular, it is shown how minor changes in the update rule can lead to dramatic changes in the behaviour of the model.

Chapter 1

Opinion Dynamics on Complex Networks

This chapter is devoted to a gentle introduction to opinion dynamics on complex networks. First, a framework [10] valid for any opinion dynamics model is described to provide a context for the problem. There exist many ways of differentiation between the various models however, the underlying structure holds for all of them. Then, complex networks are presented more in detail and their role in this context is discussed. It will be seen how a group of individuals with acquaintance relations can be mapped onto a graph and how this mathematical description can be exploited to gain insights of the opinion-formation process.

Lastly, some classical models are described such as the De Groot model [2] and the voter model [3] [11]. These models represent the foundation of this research field and have been largely investigated. In particular, the latter is the starting point of this work. Extensions of the voter model will be presented and analyzed, of particular interest for this work is the *adaptive* voter model of Durrett [9] examined in detail in Chapter 2.

1.1 Framework

The field of opinion dynamics attempts at gaining insights and making predictions about the complex process of collective opinion formation. The sociological mechanisms behind it are of various nature and strongly vary from individual to individual. It goes without saying that these processes are also strongly influenced by the the socio-economical context of the individuals. The modelling process in this research field is undoubtedly challenging. However, rather simple models are able to capture some social aspects and mechanism of opinion formation. *Homophily* and *heterophily* are examples of such mechanisms which can be easily translated into mathematical language and will be further discussed later in this work.

Aside from the intrinsic complexity of the *opinion-formation* process, models for opinion dynamics have some common characteristics. First of all, the main element is of course a set of individuals, who constitute the population under observation. Then, models are usually composed of three basic elements in the more general setting: the opinion expression format (how opinion is represented), the fusion rule (how agents interact to form their own opinion) and the opinion dynamic environment [10]. According to the opinion formats, models can be distinguished into continuous and discrete opinion models. The process undertaken by an agent to form his/her new opinion is described by the fusion rule. Each agent's opinion emerge from the merging of the opinions of the interacting agents. This comes in different flavours and characterize the opinion model, different variables can be incorporated into the fusion rule and different combinations of these variables considered. Examples of different fusion rules will be introduced in the remainder of this work. The dynamical environment influences the process as well, for instance some models introduce a random noise to the agents' opinions with the aim of describing the tendency of individuals to spontaneously change opinion. Another example is given by the social structure of a given community, some agents are more influential than others or are less prone to change, such as the *stubborn* agents [12] introduced later. The process may ultimately lead to a consensus among the agents (one opinions' cluster) or to polarization (two opinions' clusters) or more generally to fragmentation [13]. Figure 1.1 depicts the framework that has just been described.

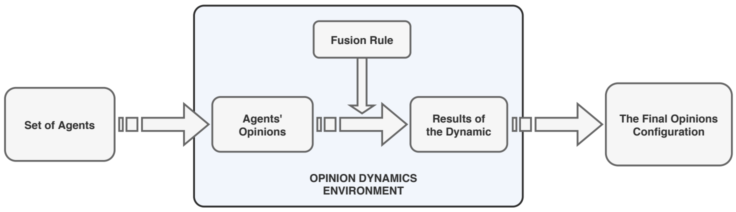


Figure 1.1: Opinion dynamics framework as presented in [10]

1.2 The Role of Complex Networks

The setting described above can be naturally translated in mathematical terms through graph theory. A graph $G(V, E)$ is a structure composed by a set of vertices V which are interconnected through links of the type $(i, j) \mid i, j \in V$ contained in the set of edges E. The translation is straightforward as depicted in Figure 1.2, each agent of the population is mapped onto a vertex (node) of the set V and each edge (i, j) shows the interconnection between two individuals. These bounds can be interpreted as acquaintance relationships when looking at social networks. Usually, the fusion rule of the opinion dynamics model involves nodes which are connected even though it will be seen that models (such [7]) randomly select agents from the entire network to resolve conflicts.

Complex networks are just a special type of graphs which have gained popularity in the last years. These graphs exhibit non-trivial topological properties that resemble more closely behaviours observed in real-world systems. Heavy-tailed degree distribution is one of these characteristics, it implies that social networks are strongly non-homogeneous, with few hubs having a remarkably high degree resulting connected to a large number of other nodes. This behaviour is observed in graphs such as the Chung-Lu where it is possible to select the desired degree distribution. Whereas, Erdős-Rényi graphs have a binomially distributed degree distribution concentrated around the mean degree, this strongly differentiate from power-law or scale-free distributions. The other "most important universal characteristics" [14] of complex networks are the small-world structure (proposed by Watts and Strogaz [15]), preferential attachment, community structure (Stochastic Block Model will be discussed) and remarkable robustness against random breakdowns. There exist many generative models such as the already mentioned Erdős-Rényi, Chung-Lu, Watts-Strogatz each of them better captures one (or more) of the aforementioned aspects.

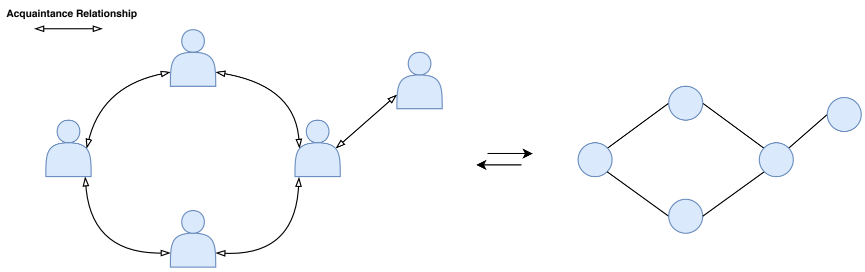


Figure 1.2: Equivalence between a network of individuals (on the left) and a graph (on the right).

1.3 Opinion Dynamic Models

In this section the models which are referred as the classical in opinion dynamics are discussed, namely the DeGroot model, the bounded confidence model and the voter model. These represent the basic building blocks of more complicated models which are modifications or extensions of the aforementioned models [10]. More attention will be devoted to the voter model which will be studied in more detail representing the starting point of this research.

1.3.1 DeGroot Model

The DeGroot model is regarded as the classical model in opinion dynamics. The more general formulation of the model, as presented in [2], considers a population of k agents which interact with each other, each agent specifies his subjective opinion on the unknown value of some parameter θ. The opinion can be specified as a subjective probability distribution for θ. However, the model can also be applied when the opinion of each member of the population is a just a point estimate of the parameter instead of the entire probability distribution. Following [10] the DeGroot model can be stated as follows. $A = \{A_1, A_2, .., A_n\}$ is the set of agents and $x_i(t)$ represents A_i's opinion at time t. Let w_{ij} be the weight A_i gives to A_j representing how agent A_j is able to influence agent A_i. The updating equation of the opinion of A_i is:

$$x_i(t+1) = w_{i1}x_1(t) + w_{i2}x_2(t) + .. + w_{in}(t), \quad t = 0, 1, 2, .. \tag{1.1}$$

The matrix $\mathbf{W}$ is a stochastic matrix, namely $0 \le w_{ij} \le 1$ and $\sum_j w_{ij} = 1$. The DeGroot model can be classified as continuous, according to the distinction made in Section "Framework", since the agents' opinions are continuous. It is generally assumed that $x_i(t) \in \mathbb{R}$. The model can be easily restated in matrix form:

$$\mathbf{x}(t+1) = \mathbf{W} \times \mathbf{x}(t), \quad t = 0, 1, 2, .. \tag{1.2}$$

Even though the model has not been defined on a network its interpretation on a graph is straightforward. The nodes of the graph represent the agents while the weighted links represent the relationships among agents. The weight of the links are exactly those contained in the matrix $\mathbf{W}$.

Friedkin-Johnsen Model

In the above formulation the weights w_{ij} in Equation (1.1) are static. However, the weights w_{ij} may change with time and opinion profile. An example of such a case is the Friedkin-Johnsen model [16][17]. The model encompass two different processes of opinion formation. In fact, unlike the DeGroot model, individuals not only are influenced endogenously by other agents' opinions, but also at each step in the process are influenced exogenously by the the conditions that have formed their initial opinions [17]. The exogenous determinants of the opinions include attributes such as gender, age and socioeconomic status but also ubiquitous roles (worker, father, husband) [17]. Let be $\mathbf{y}(t)$ be the $n \times 1$ vector of individual's belief at time t, $\mathbf{X}$ a $n \times k$ matrix containing k exogenous variables, $\mathbf{b}$ is a $k \times 1$ vector of coefficients weighting the exogenous contributions. Lastly, the matrix $\mathbf{W}$ is equivalent to the $n \times n$ weight's matrix in the DeGroot model and represent the endogenous interpersonal influences. Two equations describe the theory, the first concerns the origins of actor's initial opinions:

$$\mathbf{y}(1) = \mathbf{X}\mathbf{b} \tag{1.3}$$

The second equation describes the subsequent evolution of the process, beyond the initial opinions:

$$\mathbf{y}(t) = \alpha \mathbf{W} \mathbf{y}(t-1) + (1-\alpha)\mathbf{y}(1) \quad for\, t = 2, 3, ... \tag{1.4}$$

It is interesting the role covered by the parameter α, a scalar weight of the endogenous interpersonal influences $0 \le \alpha \le 1$. As observed in [17] this parameter describes the balance-of-forces of the endogenous and exogenous influences and is called the coefficient of social influence. The opinions that are formed following this model reflect the competing influences of the personal circumstances of an actor and the influences of the peers belonging to the same social network. [17] Such a balance-of-force has also a psychological motivation in the speculation of Festinger:

" *When a person or a group attempts to influence someone, does that person or group produce a totally new force acting on the person, one which had not been present prior to the attempted influence? Our answer is No - an attempted influence does not produce any new motivation or force. Rather, what an influence attempt involves is the redirection of psychological forces which already exist.*[1] "

[1] Taken from the paper of Friedkin and Johnsen [17] who cited Festinger [18].

The Friedkin-Johnsen model, presented in matrix form, can be stated condensing the two equations in just one as in [13]:

$$\mathbf{y}(t+1) = (1-\alpha)\mathbf{y}(0) + \alpha\mathbf{W}\mathbf{y}(t) \tag{1.5}$$

It appears clear that the DeGroot model is a special case of the Friedkin-Johnsen model. We obtain the DeGroot model when the coefficient of social influence α is equal to one and so the exogenous contribution is excluded.

1.3.2 Bounded Confidence Model

Bounded confidence models represent an entire class of models in which interactions between pair of agents happen only if the difference in opinion between the two individuals is smaller than a certain threshold, namely if the opinions are sufficiently similar. These models are becoming popular due to their consideration of psychological factors [10]. Such models allow to capture social behaviours such the homophily, which is observed in real settings. However, their analysis appears to be demanding due to the non linearity of the equations describing the interactions. The bounded confidence model can be stated in a general form as follows below, then we will discuss two bounded confidence models which differs by the fusion rule employed.

As before we can introduce the set of agents $A = \{A_1, A_2, ..., A_n\}$ each one of them has opinion $x_i(t) \in [0,1]$ at a discrete time $t \in 0, 1, 2, ...$ In addition, the threshold needs to be defined, for each individual we can define a ϵ_i. If this parameter is equal for each agent ($\epsilon_i = \epsilon_j \, \forall i \neq j$) the bounded confidence model is homogeneous otherwise, it is referred as heterogeneous.

Hegselmann-Krause model

This model can be considered as an extension of the classical DGroot model in which the weights in the matrix W depend on the particular configuration of opinions of the agents and has been introduced by Hegselmann and Krause in [13]. An agent A_i in the group A takes into account the opinion of another agent A_j only if A_j's opinion does not differ from his own opinion more than a certain confidence level ϵ_i. For a given agent A_i and opinion profile $x = (x_1, ..., x_n)$ the set containing the agents satisfying this property is defined as:

$$I(A_i, x) = \{1 \leq j \leq n \, s.t. \left|x_i - x_j\right| \leq \epsilon_i\} \tag{1.6}$$

The agents in this set are the only with which agent A_i interacts. Therefore, considering the matrix of weights W defined above, one can state that $w_ij(x) = 0 \; for \, j \notin I(A_i, x)$ and $w_ij(x) = \frac{1}{\left|I(A_i,x)\right|} \; for \, j \in I(A_i, x)$. The model assumes that agent A_i assigns an equal weight to all the connections to agents $A_j \; with \; j \in I(A_i, x)$. The equation describing the dynamics of the opinion of agent A_i in the Hegselmann-Krause model is:

$$x_i(t+1) = \frac{1}{\left|I(A_i, x)\right|} \sum_{j \in I(A_i, x)} x_j(t) \quad fort = 0, 1, 2, .. \tag{1.7}$$

Deffuant-Weisbuch Model

The Deffuant-Weisbuch model [19] is similar to the Hegselmann-Krause model but simpler since it is not necessary to define the set $I(A_i, x)$. In fact, two agents are randomly selected from A at each time instant. The two individuals influence each other only if $\left|x_i(t) - x_j(t)\right| \leq \epsilon$, the fusion rule is the following:

$$\begin{cases} x_i(t+1) = x_i(t) + \mu(x_j(t) - x_i(t)) \\ x_j(t+1) = x_j(t) + \mu(x_i(t) - x_j(t)) \end{cases} \qquad t = 0, 1, 2, .. \tag{1.8}$$

The parameter μ controls the agent's movement towards the opinion of the other agent when the condition of bounded confidence is satisfied.

1.3.3 The Voter Model

This model is the first among those presented having a discrete opinion space indeed, opinions in the voter model are binary ($x_i \in \{0, 1\}$). The model was introduced independently by Holley and Liggett [3] and Clifford and Sudbury [11]. The firsts, given a countable set A, defined a proximity process on the state space $\{0, 1\}^A$ governed by a transition function, whose definition relies on the identification of subsets of A for each one of the elements in that set. Each site $i \in A$ examines the subsets belonging to him and if one of the sites' process value is one, it rearranges its own probability of assuming the value one at the following time step. The subsets in the definition correspond to the neighbourhood of a node in terms of graph theory. The seconds studied three different processes (swapping, invasion and alternation process) in which the node placed on a regular lattice interact with each other.

Voter Model on a Regular Lattice

Initially, the voter model has been studied placing the agents on regular lattices, such as in the studies mentioned above. But, soon the model's behaviour has been investigated on heterogeneous networks showing that it differs dramatically from that on regular lattices [20]. The agents change their opinions selecting randomly the opinion of one of their neighbours $A_j \in A = \{A_1, A_2, ..., A_n\}$ the set of agents holding opinion $x_j(t)$ ($x_j \in \{0, 1\}$), being t a discrete time. The 2-D lattice L ($\lceil\sqrt{n}\rceil \times \lceil\sqrt{n}\rceil$) represents the structure on which the agents are placed. Excluding the nodes on the boundary, each agent has four neighbors, which are the only agents to which he can communicate. At each discrete time instant agent A_i selects randomly one of his neighbours, say agent A_j, and imitates his opinion. In more mathematical terms:

$$x_i(t+1) = x_j(t) \tag{1.9}$$

It is possible to start the process also on higher order lattices and many results have been proven for such homogeneous graphs.

Voter Model on a Heterogeneous Graph

The study of voter model has been carried out not only on regular graphs but also on heterogeneous graphs. This choice is imposed by the fact that real social networks are not regular and on the contrary, show strong imparity among nodes. In [20] Sood and Redner presented the voter model on a graph. Each agent corresponds to a node and can one of two opinions (*spin up* and *spin down*). The process of evolution consists of (i) selecting randomly a node of the network, (ii) assign to the node the opinion of a randomly chosen neighbor. These steps are repeated until a finite system necessarily reaches consensus.

The *direct* and *reverse* Voter Model

It is possible to provide a definition of this model in more actual terms and in particular by making use of graph theory constructs. Having the agents placed on the nodes of a graph, and the link representing interconnections, nodes are activated according to a Poisson process of rate one. When activated a node looks at its neighbours and updates either its own opinion or the one of its contacted neighbour. This difference gave rise to two different models: the *direct* voter model and the *reverse* voter model [21].

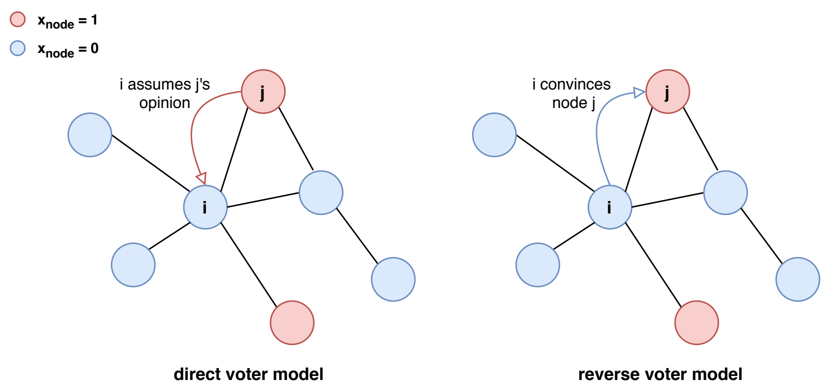


Figure 1.3: Schematic illustration of the voter model fusion rule.

We will now present the pseudo-code for the direct and reverse voter model schematically depicted in Fig. 1.3). The environment is the usual one, having the individuals of the populations paired with the nodes of a graph, each with a binary opinion ($x_i \in \{0, 1\}$). The initial distribution of opinions is randomly assigned. The dynamics develops by picking up a random node (also link-based selection are possible [7]) which selects randomly a neighbour, if any exists, and (i) adopts its opinion (direct voter model) (ii) forces the neighbour to adopt its opinion (reverse voter model).

These variations have minor impact when the underlying network is static and homogeneous (e.g. regular lattice) but, "can change drastically the model's behaviour as soon as the topology can evolve on the same time scale as the agents' opinions" [21]. *Adaptive* voter models are of primary interest in this work and are covered in Chapter 2 and the differences in behaviour, in particular in the convergence time, will be discussed in the last chapter. It will be seen how minor changes in the fusion rule can lead to remarkable changes in the time complexity of the process.

Input: A graph $G(A, E)$
$A_i \in A$ with randomly assigned opinion $x_i(0) \in \{0, 1\}$
Output: $G(A, E)$, A_i has opinion $x_i(t)$ in absorption state

loop
  ▷ Random selection of a node A_i
  if agent A_i is isolated **then**
    ▷ Do nothing
  else
    ▷ Select a random neighbor A_j
    ▷ `[dVM]` A_i copies the opinion of its neighbor A_j
    ▶ `[rVM]` A_i convinces its neighbor A_j to change opinion
  end if
end loop

Figure 1.4: Direct([dVM]) and reverse ([VM]) voter model. They differ just for one line, with ▷ is reported the direct voter model. Whereas, with ▶ the reverse voter model.

The dynamics goes on until all the agents share the same opinion, this condition represent an absorbing state and for any finite population of agent this state will be reached. However, the time to consensus might vary strongly according to the version of the voter model. It has to be noted that the two pseudo-code differ just for one line (written with a different font) but that changes the role of the agents at play, from asking for an opinion (direct) to impose the opinion on one of the acquaintance. Moreover, the two nodes A_i and A_j have different probability of having large degree. In fact, the first node A_i is chosen uniformly among all the nodes of the networks, whereas the second is chosen in the neighborhood of the first node A_i so, it is a hub with larger probability since hubs are connected with a large number of nodes. "The asymmetry in the opinion update between the two interacting nodes can then couple to the asymmetry between a randomly chosen node and its randomly chosen neighbor, leading to different dynamical properties" [21].

The last instance to be discussed is the one which uses link selection instead of node selection. When a link is randomly picked up it can either connect two agreeing nodes or two in disagreement, when the latter happens one of the two ends of the link copies the opinion of the other end. Again, the process continues until a consensus (absorbing) state is reached.

Chapter 2

Adaptive Voter-like Models

The voter model set the ground to the proliferation of many other *voter-like models*, which build upon it, slightly changing the update rule or introducing additional processes other than the *changing-opinion* one. An interesting feature of some of the upcoming models is the *adaptivity* meaning that, not only agents' opinions change in response to the network's structure, but also the network adapts according to the agents' opinions placed on its vertices. This aspect will be further discussed in Section 2.1.

Next sections are devoted to the discussion of various versions of the voter model, each of which aims at better describing the process of opinion formation. For example, the model in [22] accounts also for the habit of human beings to connect not only with like-minded individuals but also, with someone who holds a different point of view. Whereas, others have introduced agents whose opinion cannot change during the dynamic, calling them *stubborn* agents [12]. We have mentioned in the previous chapter how minor modifications in the model may lead to remarkable effects on the dynamics. This claim is in line with the differences in behaviour found by Durrett in [9] studying two instances of the voter model (*rewire-to-same* and *rewire-to-random*).

2.1 Towards Adaptive Models

2.1.1 The Holme and Newman Model

One of the fundamentals questions in the field of network dynamic is "whether the dynamics taking place on a network controls the network structure or the structure controls the dynamic" as stated in [23] by Holme and Newman. They observed that social networks tend to separate into communities of agents holding the same opinion. They argued this happens both because individuals tend to aggregate when they are like-minded and because individual are able to influence the others which are topologically close to them. Therefore, Holme and Newman [23] proposed a model in which both processes happen namely, links arise among nodes with the same opinion and neighbouring nodes influence themselves triggering opinion changes. As will be used also in the remainder, the first process is called *homophily* which indicates the tendency of connecting to agents who have the same traits.

The model leverages on the voter model which has been above described and the setting is similar. A set of N agents is randomly assigned opinions, this time the choice is not only in the set $\{0, 1\}$ but a node's opinion can be any in a set of cardinality G. In the paper Holme and Newman explore the situation in which G scales with the number of nodes N, in order to keep the ratio $\frac{N}{G} = \gamma$ constant. This implies that for N growing towards infinity also the number of possible beliefs grown unboundedly. Also version with a fixed (and small) value of γ have been studied, for example in [22] the first part of the paper is devote to the study of this model for $G = 2$. It has to be mentioned that opinions can become extinct, when none of the agents holds that particular opinion anymore. The N agents are placed on N nodes of a network, which presents M links among the nodes. These links are indicative of the mutual relation between individuals and the value M remains unchanged throughout the dynamic process. The authors justified this stating that at a given moment a human being is able to maintain only a limited number of social bounds. The links are initially placed according to the Erdős-Rényi generation model, that means between any pair of nodes an edge is placed independently on the other choices and with a constant probability. Agents' initial opinions are assigned uniformly at random choosing from the G possibilities. At each step of the dynamic an agent A_i is selected and if the degree of it is different from zero it follows one of the two processes described above. So, it either suppresses one of its connections to create a new one with an agent A_k who has the same opinion as A_I, or it changes its own opinion in order to find agreement with one of its neighbors selected at random. The model is described in more detail in the following pseudo-code.

Input: $G(A, E)$ random graph with mean degree $\delta = \frac{2L}{N}$
$A_i \in A$ has randomly assigned opinion $x_i(0) \in G$
G set of opinions
Output: $G(A, E)$, A_i has opinion $x_i(t)$, no *"discordant"* links exist

while *discordant* links are present **do**
  ▷ Random selection of a node A_i
  if agent A_i is isolated **then**
    ▷ Do nothing
  else
    ▷ Select a random neighbor A_j
    (1) With probability ϕ
      ▷ disconnect the link (A_i, A_j)
      ▷ connect A_i to a new agent A_k such that $x_i = x_k$
    (2) Otherwise, with probability $1 - \phi$
      ▷ A_i copies the opinion of its neighbor A_j
  end if
end while

Figure 2.1: *Home and Newman model.*

2.1.2 Study of the Model

This model is of particular interest because the adaptive voter model proposed by Durrett [9] represents an extension of it. The latter borrows the idea of letting the network adapt in response to opinions (links are rewired to connect like-minded individuals) and to make opinions change in response to the network structure (agents influence their neighbours).

Referring to a *discordant* link as one which connects two nodes holding different opinions, both the processes in the pseudo-code -(1) and (2)- reduce the number of such links. In fact, the homophilous process (1) tends to withdraw *discordant* links, forming new connections only between agents in agreement. Also the opinion-changing process (2) attempts to find an agreement among neighbours, even though in this case the update might also increase the number of *discordant* links since every time an agents change opinion all the connections with its neighbours flip from *discordant* to *concordant* [1] and viceversa. The process ultimately reaches a fragmented state, in which the initial graph has separated into clusters composed by agents with the same opinion. Reached this point, the structure of the network will not change anymore, any additional iteration will just rewire links within the clusters (just rewiring between like-minded nodes is allowed). Random rewiring of the clusters leads to, in the limit, components arranged according to the Erdős–Rényi model. Holme and Newman [23] called this state with no more *discordant* link the *consensus state*. In the pseudo-code above we have the number of discordant nodes as stopping condition and we deemed that to be true, since Holme and Newman were interested in the number and size of the components arising from the dynamic.

It has been revealed a phase transition between a phase in which a giant component forms, with an exponential distribution of small communities (for small values of the rewiring ϕ) and a phase with an highly fragmented network with clusters having mean size equal to γ. This transition in studied in more depth in [22] in which a mathematical description is also provided (for high values of the rewiring ϕ). It has to be noted that the number of components for small values of the rewiring ϕ depends on the characteristic of the initial graph $G(A, E)$. The number of connected components depends on the average degree δ, and if it is big enough the graph is connected and for $\phi = 0$ all the network reaches a consensus. In fact, only an agreement between all the nodes represent a stable state since no rewiring is permitted when $\phi = 0$.

2.2 Homophily and Heterophily

Building upon the Holme and Newman model (HN) presented above, Kimura and Hayakawa developed their model in [22]. The authors argued that the HN model captures only one social tendency namely *homophily*, which consists in the formation of connections with individuals who hold the same opinion. They introduced a process modelling *heterophily* which, on the contrary, makes rise to the formation of links between agents with different opinions. Therefore, in this model together with the *changing-opinion* and the *rewire-to-same* processes

[1] *Concordant*, as opposite of *discordant* represents those links which connect nodes holding the same opinion. These links are sometimes referred as inert in the literature and on the contrary, *discordant* links are also called *active*.

there exist also a *rewire-to-diverse* process, when this latter is suppressed, the model reduces to the Holme and Newman model [23].

Input: $G(A, E)$ random graph with mean degree $\delta = \frac{2L}{N}$
$A_i \in A$ has randomly assigned opinion $x_i(0) \in G$
G set of opinions
Output: $G(A, E)$, A_i has opinion $x_i(t)$, *"discordant"* links might still be present

loop
 ▷ Random selection of a node A_i
 if agent A_i is isolated **then**
  ▷ Do nothing
 else
  ▷ Select a random neighbor A_j
  (1) With probability ϕ
   ▷ disconnect the link (A_i, A_j)
   ▷ connect A_i to a new agent A_k such that $x_i = x_k$
  (2) With probability ψ
   ▷ disconnect the link (A_i, A_j)
   ▷ connect A_i to a new agent A_k such that $x_i \neq x_k$
  (3) Otherwise, with probability $1 - \phi - \psi$
   ▷ A_i copies the opinion of its neighbor A_j
 end if
end loop

Figure 2.2: Co-evolutionary model with homophily and heterophily

Kimura and Hayakawa argued that *heterophily* is a fundamental process in social networks which otherwise would fragment into isolated groups when only *homophily* is at play. This behaviour has been highlighted in the previous Section 2.1. *Heterophily* allows to create bridges between community-like structures holding different opinions, since it creates connections between agents who disagree. In [22] has been demonstrated that for the case $G = 2$ the stability of the coexistence of two states. This shows that the dynamical behaviour strongly changes compared to the behaviour of the Holme and Newman model just adding another process which captures a different sociological aspect.

2.3 Voter Model Seeking for a Third Opinion

In [7] yet another voter-like model has been proposed, which introduces a new parameter ρ interpreted as a "measure of social conformity". It uses the idea of *of adaptivity* as presented by Holme and Newman [23] and extends their model. As always, individuals are represented as the N nodes of a graph while K edges are the social relationship among them. The proposed model allows just for two opinions, that would correspond to $G = 2$ in the HN model. When rewiring does not take place (so, with probability $1 - \alpha$); instead of having one of the two ends of the selected link adopting the opinion of a neighbor, the two agents look for a third opinion. Therefore, another node is selected from the entire graph and with probability ρ both nodes accept the opinion of this node whereas, with probability $1 - \rho$ they assume the opposite probability. Again we sketch the pseudo-code describing the model, it is just slightly different from the one presented above in Figure 2.1.

Input: $G(A, E)$ random graph with K links
$A_i \in A$ has randomly assigned opinion $x_i(0) \in G$
$G = \{0, 1\}$ binary set of opinions
Output: $G(A, E)$, A_i has opinion $x_i(t)$, no *discordant* links exist

while *discordant* links are present **do**
 ▷ Random selection of a link (A_i, A_j)
 if link (A_i, A_j) is *discordant* **then**
  (1) With probability ϕ
   ▷ disconnect the link (A_i, A_j)
   ▷ connect A_i to a new agent A_k such that $x_i = x_k$
  (2) Otherwise, with probability $1 - \phi$
   ▷ randomly select a node A_z in A
   (a) With probability ρ
    ▷ A_i and A_j copy the opinion of A_z
   (b) Otherwise, with probability $1 - \rho$
    ▷ A_i and A_j assume the opposite opinion as A_z
 else
  ▷ Do nothing
 end if
end while

Figure 2.3: Voter model accounting for *social conformity*.

What is interesting in the study carried out in [7] is that the time to consensus of the adaptive voter model of Holme and Newman (linear) is the critical case between a logarithmic and exponential complexity. This change in behaviour can be observed varying the parameter ρ. When $\rho = \frac{1}{2}$ the model reduces to the HN model since the third opinion does not bias the decision, with probability $\frac{1}{2}$ node A_i assumes the opinion of node A_j and with probability $\frac{1}{2}$ the opposite occurs. It has been found that for small values of rho the behaviour is exponential whereas for larger values it is logarithmic. A dynamical phase transition is observed when rho crosses the critical point represented by $\frac{1}{2}$, the two regimes are substantially different and the results of particular interest since they demonstrate how sensitive these models are to the underlying parameters.

2.4 Voter Model with *Stubborn* Agents

Many models have been developed starting from the voter model proposed independently by Holley and Liggett [3] and Clifford and Sudbury [11] such as the model with stubborn agents proposed in [12] by Yildiz et al. They argued that the fact that the voter model converges to consensus (absorbing state for the dynamic) does not reflect what can be observed in reality. In fact, in societies it is rare to observe an overall agreement as a result of the interaction dynamics. On the contrary, "most societies appear to exhibit persistent disagreement" [24]. To capture this aspect the authors have introduced agents who do not change their opinion throughout the process and because of this trait the called them *stubborn* agents. The presence of such stubborn agents with opposing opinions prevents consensus in the social network and moreover, the opinion of the single agent does not settle into a certain value, capturing the "persistent disagreement" that has been mentioned above.

Input: $G(A, E)$, $A_i \in A$ has randomly assigned opinion $x_i(0)$
A_0, A_1 nonempty sets of stubborn agents
Output: $G(A, E)$, where A_i has opinion $x_i(t)$

```
loop
  ▷ Awakening of an agent A_i according to a Poisson process
  if agent A_i is non-stubborn then
    if the neighbourhood N_{A_i} NOT empty then
      ▷ Select a random neighbor A_j
      ▷ A_i assumes the opinion of A_j
    end if
  else
    ▷ Do nothing, stubborn agents cannot change opinion
  end if
end loop
```

Figure 2.4: Voter model with stubborn agents.

The setting for the model is analogous to the one developed above for the voter model, one important difference is that here the underlying graph is directed. So, again being $A = \{A_1, A_2, ..., A_n\}$ the set of agents, $x_i(t)$ the binary opinion held by agent A_i ($x_i \in \{0, 1\}$) and A_0, A_1 the nonempty disjoint sets of stubborn agents, holding respectively opinion 0 and opinion 1. Considering a graph $G(A, E)$, where E is the set of directed graph which describes the bounds between the agents (A_i, A_j). Also in this version one node can be influences only by its neighbours $N_{A_i} = A_j|(A_i, A_j) \in E$. Each non-stubborn is awakened according to a Poisson process with unitary rate and updates its opinion following the same rule described for the *direct* voter model. Even if also every stubborn agent awakens according to a unitary-rate Poisson process, it does never change opinion maintaining the initial opinion it has been assigned with. For the non-stubborn agents an initial opinion $x_i(0)$ is randomly assigned before the process starts.

2.5 Durrett Adaptive Voter Model

This model is of central interest because will be the main subject of investigation for this thesis work. The results found in Chapter 4 and Chapter 5 strongly rely on this model. It follows a description of the model and some results, which have been presented by Durrett [9] [25] and confirmed thanks to the c++ based simulator developed (see Chapter 3).

2.5.1 Model Description

The adaptive voter model is an interesting evolution of the voter model which finds its roots in the aforementioned work of Holme and Newman [23]. Not only the agents can change opinion according to their neighbors but also the network structure of social interactions may change over time. Here, we will present the two versions proposed by Durrett in [9], the *rewire-to-same* and the *rewire-to-random* model.

Input: $G(A, E)$ random graph with mean degree $\delta = \frac{2L}{N}$
$A_i \in A$ has randomly assigned opinion $x_i(0) \in \{0, 1\}$
Output: $G(A, E)$, A_i has opinion $x_i(t)$, no *"discordant"* links exist

while *discordant* links are present **do**

▷ Random selection of a ***discordant*** link (A_i, A_j)

(1) With probability α

▷ disconnect the link (A_i, A_j)

▷ [`RtS`] connect A_i to a new agent A_k such that $x_i = x_k$

▶ [`RtR`] connect A_i to a new agent A_k such that $x_k \notin N_{A_i}$

(2) Otherwise, with probability $1 - \alpha$

▷ A_i copies the opinion of its neighbor A_j

end while

Figure 2.5: Durrett adaptive voter model. `RtS` stands for "rewire-to-same" and `RtR` for "rewire-to-random". The procedure for the two is equal except for the rewiring policy, it is different highlighted in the pseudo-code. Implement ▷ for the first version and ▶ for the second version

The first version is practically identical to the Holme and Newman model, except for the fact that now the opinions are constrained to 2 instead of a number proportional to the size of the graph and *discordant* links are picked up instead of random nodes. This last shrewdness is intended to speed up the time to convergence, avoiding to picking up concordant links (which would not be modified by the dynamics) and isolated nodes (which do not have neighbors). The *rewire-to-random* version goes more in the direction of the model with *homophily* and *heterophily* in the sense that at each rewiring step (with probability α) a node can either connect to an individual sharing its same point of view or to someone who is in disagreement. As Durrett proved, and as will be demonstrated later the two models show a strongly different dynamical behaviour. In the following the pseudo-code for the Durrett adaptive voter model is presented, the two versions differ just in the selection of the new node to which reattach the suppressed link.

2.5.2 Dynamical Behaviour of the Model

Observing the fraction of nodes in minority state when the process stops (no more discordant edges) the system undergoes a phase transition. In the rewire-to-same case, it is similar to the one firstly observed by Home and Newman [23] and then studied by Kimura and Hayakawa [22]. It consists of a discontinuous phase transition for a critical value of the rewiring α_c which does not depend on the initial distribution of the opinions on the graph. However, it does depend on the mean degree of the nodes of the network as it will confirmed in Chapter 4 by simulations. Differently, the rewire-to-random version exhibits a continuous phase transition according to a curve that Durrett called the *universal curve*. The two models have a quite different behaviour however, after the critical value the fraction of nodes in the minority opinion attests at a percentage equal to the one of one-opinionated nodes at time zero.

Durrett et al. also noted that the number of nodes in the minority opinion is correlated to the number of discordant nodes. This lead to conjecture that the evolving voter model possess a one parameter of quasi-stationary distributions. It has been shown that several statistics (as a function of the number of ones at time t) after a transient, follow a curve: a parabola for the number of discordant nodes, a cubic when the number of oriented triplets $0 - 1 - 0$ is considered. To motivate the long time survival (for α under α_c) and to formalize what has just been explained Durrett et al. used a *separation of time scales* argument:
"The time to converge to equilibrium is much smaller than the time needed for the density to change, so if the time is scaled appropriately then the system is always close to an equilibrium and the parameter follows a diffusion process". [25]

Chapter 3

Simulator Description

3.1 Architecture

The stochastic processes we aim to simulate are particularly demanding in terms of time complexity. The voter model on homogeneous graphs needs $O(N^2)$ updates in order to reach a consensus and the behaviour on heterogeneous graphs is even richer, ranging from logarithmic to exponential complexity. Therefore, it is important to develop a sufficiently efficient code in order to obtain results in a reasonable time since many realizations have to be performed and many points with different system parameters have to be collected. Initially, it has been developed a Python code to implement the pseudo-code presented in Figure 2.5. The language itself is well suited to represent graphs, being dictionaries one of the fundamental types of the language. Moreover, many libraries provide easy-to-use methods which allow for fast development and implement most of the algorithms of interest in the field (e.g. `NetworkX`).

However, execution times appeared to be unacceptable and Python did not provide enough control on the data structures in order to speed up the simulation. So, it has been decided to switch to `c++` which is more low-level but guarantees great control on the underlying data structures. This drastically reduced the time to completion of the simulations even though it increased the development time for the code itself. Python has been anyway employed for plotting the results since it provides easy-to-use libraries (e.g. `matplotlib`) and highly customizable plots. A block diagram is provided in Figure 3.1 the `c++` simulator performs all the computations and provides the results as text files fed then into a Python script providing the plots presented in this work.

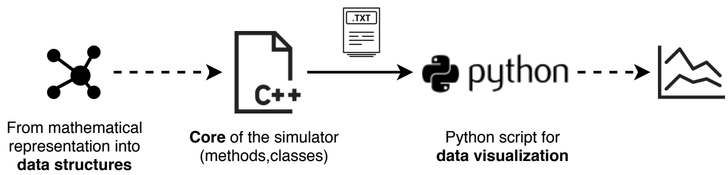


Figure 3.1: Schematic representation of the workflow.

3.2 Graph Representation

A graph $G(V, E)$ is a construct defined by a set of nodes, e.g. $\{0, 1, ..., N-1\}$, and a set of pairs of the type $(i, j) s.t. i, j \in V$, the links. It can be represented in at least three ways: (i) as a list of edges (ii) with its adjacency matrix [1] (iii) keeping an adjacency list for each node. The list of edges of the network is not suitable in our setting. In fact, it is of primary importance for us to be ale to access the neighbourhood of a node since when a node changes its opinion all the connections with the neighbours flip status from active to inert or vice versa. This update is very handy when a list of all the nodes is available and each of them maintains its own neighbours list. Moreover, some of the algorithms which have been previously presented select nodes instead of edges, running the dynamics on a graph represented with a edge list would be more involved. The adjacency matrix form is usually employed when the graph is particularly dense and with programming languages optimized for matricial calculations, such as `Matlab`.

Figure 3.2 presents a schematic representation of the data structures that should be used to represent a graph. In the middle of the figure a container can be seen, it holds the labels of the nodes, which allow to access the neighbouring lists [2] of each node. These lists will be divided in concordant and *discordant* according to the opinion of the neighbour. Agreeing neighbours will be added to the concordant list, opposers in the discordant list, this division facilitates some operations on the data structures. This procedure has been performed for node n_0 in Figure 3.2, the lists have been completed according to the portion of the graph on the top-left corner of the image.

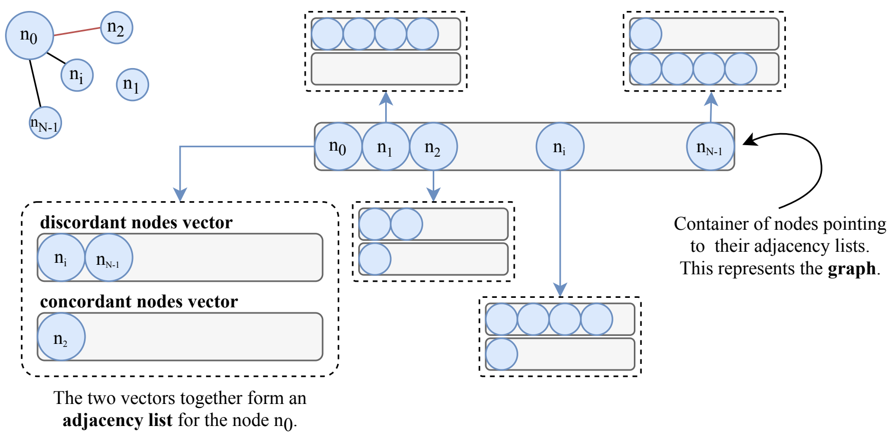


Figure 3.2: High level description of the data structures used to represent the graph.

[1] The adjacency matrix A of a graph $G(V, E)$ is a $|V| \times |V|$ matrix whose elements a_{ij} are equal to 1 if $(i, j) \in E$ and 0 otherwise.

[2] List should not be interpreted as the "list" data structure but, as a sequence of elements. In fact, as specified in the next section, these containers will be implemented using std::vectors.

3.3 C++ Class Structure

In this section we give some more implementation details regarding the `c++` simulator. Two classes have been written: `vertex` which represents a generic node and `Graph` which groups nodes together to form a network.

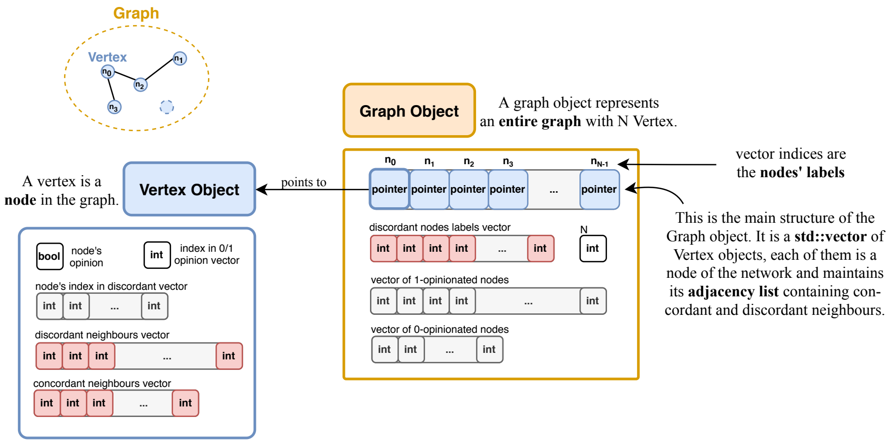


Figure 3.3: `C++` representation of the two fundamental entities: node and graph. Class members are in the boxes, the figure shows the structure of the elements.

Nodes are therefore `Vertex` instances, each of them has a binary opinion and two `std::vector<int>` containing the labels of the concordant and discordant neighbouring nodes. It has been decided to use `std::vector` because allow for random access in constant $O(1)$ time and due to the limited size of these vectors (in the order of the average degree, low for sparse networks) also research and removals are not computationally intensive. From Figure 3.3 can be seen other two members an integer and a vector of int, these keep the index at which the node is located in the 0 or 1 opinioned nodes container (int value) and all the occurrence's index of the node in the discordant vector (vector of int). This last structure is probably the most important in the `Graph` class since it contains all the *unsatisfied* nodes namely those who have at least one discordant neighbour. It has been implemented again with a `std::vector<int>` because of the $O(1)$ random access, vital when a node needs to be picked up randomly in the container. However, this time the size of the structure can be rather large, up to the total number of nodes N. To ameliorate the performance of the deletion in a vector (that is $O(N)$) each `Vertex` object saves the indices at which it resides in the discordant nodes vector. When knowing the index, deletion takes only constant $O(1)$ time. In fact, it is possible to swap the node that needs to be eliminated with the last element of the vector, and then `pop()` the last element (in $O(1)$). The actual graph is represented by a vector of pointers to `Vertex` objects, the indices are seen as the labels of the nodes, then the `Vetex` keeps all the data of interest, most importantly the neighbouring list. In addition, the labels of the nodes holding opinion 1 and opinion 0 are dynamically stored in vectors, this is needed for the rewire-to-same adaptive model which draws new nodes for the detached link only in the set of agreeing nodes.

Algorithm 1: $O(1)$ deletion from an `std::vector`, this procedure is valid only if the order in the vector is not important.

deleteFromVector (v, i)
  Input : A vector v of size N to be updated and an index i
  Output : The vector v without the element at index i
if *v is **not** empty* **then**
  `swap` $v[i]$ with $v[N-1]$;
  `pop()` the last element of v;

We will not go more into the details of the implementation. Below we provide the list of members and methods of the two classes we use in the simulator. The graph generators methods of `Graph` will be covered in great detail in the next section. The last three methods of the `Graph` class are worth mentioning, `DurrettRewireToRand()` and `DurrettRewireToSame()` simply implement the pseudo-code for the adaptive model presented in Chapter 2. The method `findClusters()` uses deep first search DFS to classify the nodes according to the cluster they belong to, this allows also to count the number of clusters in the network.

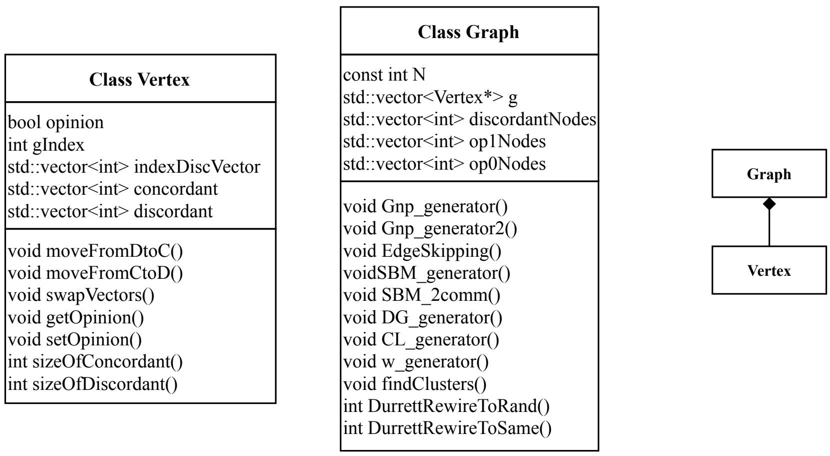


Figure 3.4: Relations between the `Vertex` and `Graph` classes together with the list of the members and the methods of each class.

3.4 Efficient Graph Generating Algorithms

Synthetic graphs are frequently employed to study real-world systems such as computer networks or biological systems. This allows to simulate and to analyze the system without the need of real data which might be difficult to obtain, if not impossible, in most cases. We use them to recreate a social network which represent the starting point of our simulations. It is important that these artificial networks maintain some of the characteristics observed in real complex networks.

The first model presented is the Erdős-Rényi and is the more versatile and analytically tractable since the probability of existence of an edge is constant and independent from the existence of other edges in the network. However, this generative model does not produce graphs with a power-law degree distribution (typically observed in real-world systems) but, with a binomial distribution. Moreover, there exists no clear community structure in a Erdős-Rényi graph. Nevertheless, it is widely-used and it has been employed in all the studies which have been presented in Chapter 2 as initial configuration for the evolving network.

In this chapter it will be presented a procedure to obtain a graph with a desired degree distribution, it allows to force a scale-free degree distribution. In [26] it is discussed how to generate large scale-free networks with the Chung-Lu random model and how to generate a vector $\mathbf{w} = (w_0, w_1, ..., v_{N-1})$ which contains the expected degree of each node in the network according to this distribution. Then, it is presented an efficient algorithm both in terms of time an memory occupation, which takes this vector as an input and produces a scale-free Chung-Lu graph [27]. A slight modification of this algorithm allows to generate a stochastic block model, which possesses a clear community structure. This last generative model is the most used in the investigations carried out in chapter 4.

Typical complex networks have millions of nodes and this poses a challenge to the generative algorithms of synthetic graphs. Algorithms should be efficient both in terms of time and memory occupation. Algorithms on graphs are regarded as optimal when the asymptotic time complexity is $O(N + m)$, where N is the number of nodes in the network and m the number of links. The algorithms that are presented below are all efficient and are implemented according to the classes and data structures described above.

3.4.1 Erdős-Rényi

The Erdős-Rényi is a very popular random graph model, widely used in the literature. To be precise, there exists two classes of random graphs falling under the "Erdős-Rényi" term: $\mathcal{G}(n, p)$ and $\mathcal{G}(n, m)$. The difference is slight however, worth mentioning. The class $\mathcal{G}(n, p)$ encompass all those graphs in which links are chosen independently on the other choices and with constant probability p. Using such a generative rule the number of links in the network is not fixed but, it is possible to compute the expected number of links $E[L] = p \cdot \binom{N}{2}$, where $\binom{N}{2}$ represent the number of all the possible pairs between the N node (self-loops are excluded). This last result can be rewritten as a function of the average degree of a node δ: $E[L] = p(N-1) \cdot \frac{N}{2} = \frac{\delta N}{2}$. The other class, $\mathcal{G}(n, m)$, represent the random graphs with N nodes and m links. It is possible to construct $\binom{\binom{N}{2}}{m}$ different graphs distributing the m links among the N possible nodes. Each of the possible link has equal probability to belong to a random graph of this class. The fact that the number of edges is fixed induces a weak dependence between the links [14]. We will refer as Erdős-Rényi graph elements belonging to the first mentioned class $\mathcal{G}(n, p)$, the generator aims at generating such a network.

The most straightforward way to generate an ER graph is to take any possible pair of nodes $\binom{N}{2} = \frac{N(N-1)}{2}$ in the graph and keep an edge with probability p and throw it away otherwise. This procedure is immediate however inefficient. In fact, the time complexity appears to be $O(N^2)$, not viable to generate large networks. Batagelj and Brandes [28] proposed an algorithm running in $O(N+m)$ which employs the *geometric method* that avoids to consider edges which would not be added to the graph's edge list. The main observation is that, especially for sparse graphs ($p(n) \in o(1)$), most of the possible node pairs (links) are discarded and only few probabilistic evaluations lead to the creation of a link. The main idea of the method is to skip over the links which will not be created, jumping directly to the next link which will be added to the network. The method is thoroughly explained in the next section.

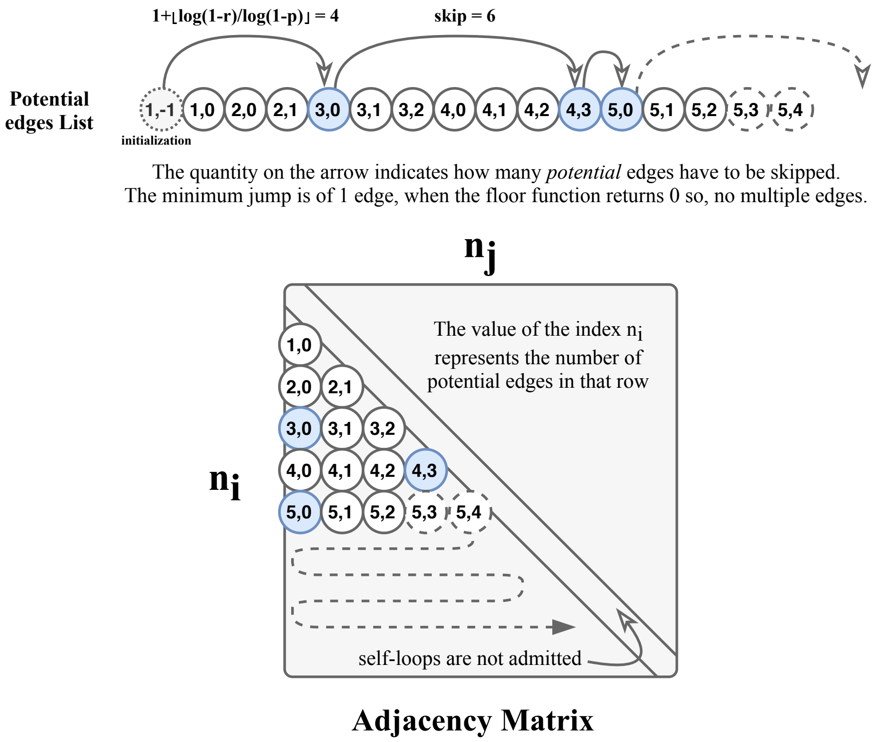


Figure 3.5: Schematic representation of the edge-skipping method and relation with the adjacency matrix. The potential edge list represents the edges in lexicographic order whose position is then shown in the adjacency matrix. Jumps provided by the edge-skipping method allow to consider only links which will be created in the graph avoiding useless evaluation.

The Geometric Method

The *geometric method* also called *edge-skipping technique* in [27] aims at skipping the potential edges that are not created. First of all, the edges need to be listed in lexicographic order so, each of them can be uniquely identified with an integer value [3] (id of the link). At this point, it is sufficient to observe that the jumps over potential edges, leading to two consecutive link creations,

[3]Links are represented as pairs (i, j), since the graphs are undirected (i, j) is the same as (j, i) according to the algorithm, it is considered the superior or inferior portion of the adjacency matrix to avoid this ambiguity. The lexicographic ordering is then unique and each link can be associated with a integer value.

are geometrically distributed namely $P\{jump = k\} = (1-p)^{k-1}p$. So, it is possible to generate edges of an Erdős-Rényi graph just by extracting values of a geometrically distributed random variable which indicates how many ordered edges have to be skipped to pick a link to be added in the edge list. Following [28] every possible value k for the jump is assigned a consecutive sub-interval I_k of $[0,1)$ of size $(1-p)^{k-1}p$. Clearly, the infinite summation of all the possible interval's sizes equals to 1 since those correspond to the discrete probability distribution of a geometric random variable. These intervals are consecutive, the $k-th$ interval ends at $\sum_{i=1}^{k}(1-p)^{i-1}p = 1-(1-p)^k$ namely the sum of the $k-th$ interval's size and all the previous ones. Therefore, the jumps can be obtained by randomly drawing a number r in the interval $[0,1)$ and then returning the integer k of the interval I_k it belongs to:

$$r < 1-(1-p)^k \iff k > \frac{\log(1-r)}{\log(1-p)}$$

The value of the jump over the potential links is then chosen to be:

$$k = 1 + \lfloor \frac{\log(1-r)}{\log(1-p)} \rfloor$$

this explains the increment in the pseudo-code of the generator.

Algorithm 2 exemplifies the procedure to generate a graph in $\mathcal{G}(n,p)$. Lines 5 and 6 are motivated by the *geometric* method just presented. The internal `while` cycle implements the actual *edge-skipping* and updates the elements of the pair representing the link (n_i, n_j) [4]. In fact, as it can be seen from Figure 3.5, n_i indicates the number of *potential* edges having n_j as an extreme. If $n_j \geq n_i$ means that the element (n_i, n_j) is not in the lower triangular portion of the adjacency matrix. When it holds and the `while` is executed, at each iteration n_j (which in line 7 has been incremented with the number of edges to be skipped) is decremented by n_i (the number of possible edges with n_i as one of the ends). The value of n_i is incremented to start visiting a new row of the matrix. The number of n_j is decremented and therefore potential edges are skipped, until the pair (n_i, n_j) will be in the lower triangular portion of the matrix. This procedure follows the rational shown in the first half of Figure 3.5, where the possible edges are presented as a list and translates this procedure onto the adjacency matrix. The algorithm performs a row-wise traversal of the lower triangular part of the adjacency matrix.

To what concerns time complexity it can be noted that the outer `while` loop is executed as many times as the links of the network (m) (plus one, since the last execution would not lead to the generation of an edge). And the inner loop, is executed exactly N times, since at every iteration of it the value of n_i is incremented and when it reaches the value of N both the conditions of the cycles become $false$.

[4] This is basically a translation from the order number in the lexicographic order and the two end-label in the pair (n_i, n_j)

Algorithm 2: Generates a *G(n,p)* graph in $O(n+m)$

GnpGenerator (N, p)

Input : A non-negative integer N and a real $p \in (0,1)$

Output : graph $G(\{0, ..., N-1\}, E) \in \mathcal{G}(n,p)$

Define ad empty set E;

$n_i \longleftarrow 1$;

$n_j \longleftarrow -1$;

while $n_i < N$ **do**

  draw r uniformly from $[0,1)$;

  $n_j \longleftarrow n_j + 1 + \lfloor \frac{\log(1-r)}{\log(1-p)} \rfloor$;

  while $n_j \geq n_i$ *and* $n_i < N$ **do**

    $n_j \longleftarrow n_j - n_i$;

    $n_i \longleftarrow n_i + 1$;

  if $n_i < N$ **then**

    $E \longleftarrow E \cup \{n_i, n_j\}$;

3.4.2 Stochastic Block Model

The differentiating feature of this generative model is the fact that it produces graphs with communities. As shown in Figure 3.6 it is possible to generate a graph with two densely connected components just weakly connected to each other. The parameters of the model are (i) the number of nodes N (ii) a partition of the node set into disjoint communities $\{C_1, ..., C_k\}$ and (iii) a symmetric $k \times k$ matrix M containing the link probabilities. The generating procedure consists in taking any two nodes $n_i \in C_i$ and $n_j \in C_j$ and adding a link to the edge list E of the graph $G(V,E)$ with probability $(P)_{ij} = p_{ij}$.

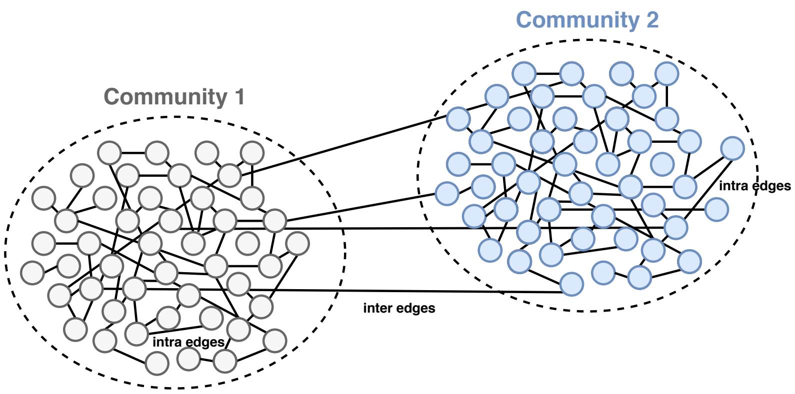


Figure 3.6: Image of a Stochastic Block Model (SBM) with 2 communities weakly interconnected. Intra and inter edges are highlighted.

The algorithm we use to generate a stochastic block model has been presented by Alam et al. [27], it is a modification of a broader-scope algorithm called **DG**, employed to generate graphs from a given degree distribution. The main idea is to group the nodes according to their expected degrees and then use a method referred to as `EdgeSkipping` to create *intra* edges, links between nodes belonging to the same class, and *inter* edges, links between nodes of different classes. In the case of the stochastic block model, the nodes are grouped according to the community they belong to so, the concept of classes is substituted by that of communities. The algorithm can be used to generate graphs with an arbitrary number of communities prescribing the connections among communities as desired. We will mainly focus on graphs with two communities having the same *intra* edges probability and a fixed probability of connection between nodes of different communities. The inputs of the algorithm are a set of N nodes, k disjoint subset of these nodes $C_1, C_2, ..., C_k$ representing the communities and a matrix M containing the connection probabilities among the communities. The element M_{ij} defines the probability that a node belonging to the set C_i is connected to a node in the set C_j. The procedure to create these links is analogous to the one used to generate Erdős-Rényi graphs and in fact, uses the same *edge-skipping* rule already explained. In this instance a slightly different procedure is employed to identify the two ends n_i and n_j of a link from the integer label assigned to each *potential* edge (ordered in lexicographic order). The pseudo-code of the `EdgeSkipping` algorithm is shown below and is very similar to Algorithm 2.

Algorithm 3: Generates edges (n_i, n_j) according to the *edge-skipping* technique

EdgeSkipping $(i, j, p, first, last)$

Input : community indices `i` and `j`
`first`, `last` cardinality of *potential* edge list
`p` link probability between communities

Output : Set of edges between the communities `i` and `j`

$e \longleftarrow first - 1$;
while $n_i < last$ **do**
  draw r uniformly from $[0, 1)$;
  $e \longleftarrow e + 1 + \lfloor \frac{\log(r)}{\log(1-p)} \rfloor$;
  if $n_i < last$ **then**
    if $i{=}j \triangleright$ **then**
      $\Delta_i \longleftarrow \lceil \frac{-1+\sqrt{1+8e}}{2} \rceil$;
      $\Delta_j \longleftarrow e - \binom{\Delta_i}{2} - 1$;
    else
      $\Delta_i \longleftarrow \lfloor \frac{e-1}{N_j} \rfloor$;
      $\Delta_j \longleftarrow (e-1) \mod |C_j|$;
    add $(\lambda_i + \Delta_i, \lambda_j + \Delta_j)$ to the **edge list**;

The `EdgeSkipping` method aims at creating links between communities (or more in general classes of nodes). The size N of the network is not provided to this method because it does not have a meaning anymore since we are considering subsets of nodes. Instead, the procedure needs `first` and `last` which together tell the range of variability of the edge labels, from this value it would then be possible to obtain the pair (n_i, n_j) to be added to the edge list of the overall network. Two scenarios are possible as depicted in Figure 3.7. It is possible to create edges within the same community (intra-edges) and it is equivalent to generate an Erdős-Rényi graph. Or it is possible to create links across communities (inter-edges). In the latter situation, the edge-skipping technique is employed to generate a random bipartite graph between C_i and C_j. To put it in other words, one does not only consider as potential edges those in the lower triangular part of a matrix. Any possible entry of an $C_i \times C_j$ matrix is a potential edge since the sets from which the nodes are drawn are disjoint.

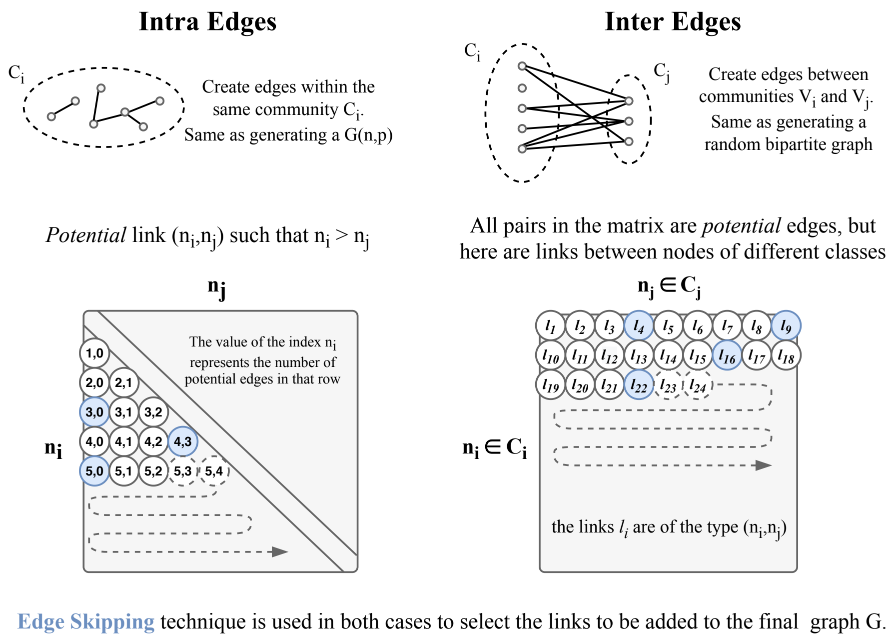


Figure 3.7: The image shows the two possible scenarios for edge generation which imply different structures. If the links have to be constructed within the same set of nodes (same community) they are called *intra edges*. Whereas, when the links are among nodes belonging to different communities then we refer to them as *inter edges*.

The Algorithm

Some of the parameters appearing in Algorithm 3 has not yet been explained, such as $|C_j|$ and λ_i. These belong to the calling function `SBMGenerator` which is the one that generates the overall SBM graph. The nodes of the network are identified with integers value from 0 to $N-1$ and we already said that them are organized in classes. Nodes belonging to the same class are represented with consecutive integers (as shown in Figure 3.8). The first `for` cycle in the pseudo-code aims at identifying the indices λ_i at which the first elements of each group are located. Being the size $|C_i$ of the communities an input of the algorithm, this procedure can be done on the flight. These λ_i values represent the offsets needed to the `EdgeSkipping` method to output the correct edge just knowing the group number. Algorithm 4 just iterates over the possible combinations of the sub-classes of nodes and call the `EdgeSkipping` method with the appropriate parameters.

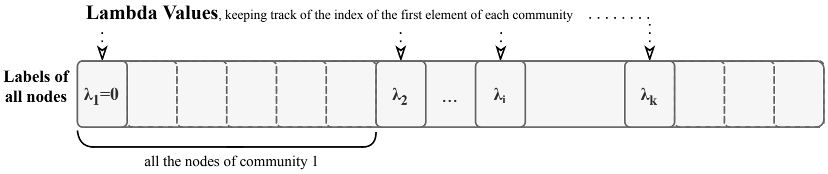


Figure 3.8: Figure that shows how the node labels are organized.

Algorithm 4: Generate a Stochastic Block Model

SBMGenerator $(\{0,1,...,N-1\}, \mathbf{M}, \{C_1, ..C_k\})$
Input : a set of nodes $\{0,1,...,N-1\}$
a matrix $k \times k$ M with connection probabilities
the k community sizes $\{|C_1|, ..|C_k|\}$
Output : a stochastic block model with k communities
interconnected according to M

$\lambda_1 \longleftarrow 0$;
for $i = 2$ **to** Λ **do**
    $\lambda_i \longleftarrow \lambda_{i-1} + |C_i|$;
for $i = 1$ **to** Λ **do**
    for $j = i$ **to** Λ **do**
        if $i = j$ **then** `// intra edges for` C_i
            `EdgeSkipping`$(i, i, M_{ii}, 1, \binom{|C_i|}{2})$;
        else `// inter edges between` C_i `and` C_j
            `EdgeSkipping`$(i, j, M_{ij}, 1, |C_i||C_j|)$;

3.4.3 Chung-Lu

Chung and Lu [29] argued that the Erdős-Rényi model only produces graphs whose nodes have the same expected degree. However, this uniformity does not reflect real-world networks in which few nodes called *hubs* have much larger degree than the other nodes. This property is a consequence of the power-law degree distribution, also called scale-free property since the functional form of such a distribution remains the same (except for a multiplicative factor) for rescaling of the independent variable. So, Chung and Lu proposed a generative model in which a vector with N elements is given as input to the generator, each element representing the expected degree of one of the N nodes of the network.

In case all the entries of the vector assume the same value, the Erdős-Rényi model is obtained. Later it will be presented a method that allows to produce a vector of expected degrees $\mathbf{w} = (w_0, w_1, \ldots, w_{N-1})$ which leads to a scale-free network [30]. The generated network needs also to be sparse, this is an important feature together with the properties of the degree distribution. It has been observed that when dealing with power-law distributions $\alpha k^{-\gamma}$ the graph can be sparse and at the same time have *hubs* only if the power exponent γ assumes values between 2 and 3.

The class of Chung Lu graphs $\mathcal{G}(\mathbf{w})$ [29] is composed by the graphs having $\mathbf{w} = (w_0, w_1, \ldots, w_{N-1})$ as expected degree sequence for the N nodes in the network. Each element of the vector is referred to as the *weight* of the node and is used to calculate the link probability between nodes. In fact, the link existence probability between nodes n_i and n_j, call it p_{ij} is calculated as the product of the node's weights divided by the sum of all the weights S. A link is added to the edge list randomly according to p_{ij} and independently from other choices. It has to be note that for mathematical convenience self-loops are allowed namely p_{ii} is different from zero.

$$S = \sum_i w_i \qquad p_{ij} = \frac{w_i w_j}{S}$$

It must be noted that in some situations the value of p_{ij} could be bigger than 1 following the definition that has been provided above. To avoid this, Chung and Lu assumed that $\max_i {w_i}^2 < \sum_i w_i$, in [26] vectors $\mathbf{w}$ are called *admissible* if $w_0 \geq w_1 \geq ... \geq w_{N-1} \geq 0$ and

$${w_0}^2 \leq \sum_{i=0}^{N-1} w_i$$

Fasino and Tonetto [26] provide formulas to produce a vector of expected degrees $\mathbf{w}$ of a Chung-Lu scale-free network, with prescribed expected average degree d and largest expected degree m. The weights in the vector are calculated according to the following formula:

$$w_i = c(i_0 + i + 1)^{\frac{1}{1-\gamma}} \quad i = 0, 1, ..., N-1$$

Theorem 2 in the paper (see [26] for more details) provides formulas for the parameters c and i_0 under certain hypothesis on the average expected degree $d(N)$ and largest expected degree $M(N)$. These have been used to produce the vector of weights w then fed into the generative algorithm described in the pseudo-code.

$$c = c(N) = (1 - \frac{1}{\gamma - 1})d(N)n^{\frac{1}{\gamma-1}}$$

$$i_0 = i_0(N) = N(\frac{(1 - \frac{1}{\gamma-1})d(N)}{M(N)})^{\gamma-1} - 1$$

The expected average degree and maximum degree has been chosen in order to satisfy the hypothesis of the theorem: $d = 12$ and $M(N) = N^0.45$. Moreover, we are interested in graphs which have a power-law degree distribution but, are sparse at the same time. This is the case if $2 < \gamma < 3$, we choose γ equal to 2.1.

Alam et al. [27] algorithm was primarily intended to generate Chung-Lu graphs, in Section 3.4.2 it has been modified in order to produce a stochastic block model. By using the `EdgeSkipping` procedure (Algorithm 3) and the pseudo-code below (called **DG** algorithm) it is possible to generate Chung-Lu graphs efficiently in terms of both space and time. Space efficiency is attained by grouping the edges according to their expected degree and keeping in memory only the label of the first node of each group λ_i. The input of the algorithm instead of a vector of N entries is a set $D = \{d_1, ..., d_\Lambda\}$ containing only the Λ distinct expected degrees values and a set containing the sizes n_i of these groups of same-degree nodes. Storing just the degree distribution allows to obtain $O(\Lambda)$ space complexity. The idea behind the algorithm is the same as the one presented for the stochastic block model generator, the only difference is that now instead of having communities we have groups of nodes with the same expected degree. The calls to the `EdgeSkipping` procedure needs to be modified as showed in the pseudo-code.

Algorithm 5: Chung-Lu graph generator using DG algorithm

DGGenerator $(\{d_1, d_2, ..., d_\Lambda\}, \{n_1, n_2, ..n_\Lambda\})$

Input : a set of unique expected degrees $D = \{d_1, d_2, ..., d_\Lambda\}$ same-degree classes' size $\{n_1, n_2, ..., n_\Lambda\}$

Output : Chung-Lu graph with prescribed degree distribution $(D, \{n_1, .., n_\Lambda\})$

$\lambda_1 \longleftarrow 0$;
for $i = 2$ **to** Λ **do**
  $\lambda_i \longleftarrow \lambda_{i-1} + n_i$;
$S \longleftarrow \sum_{i=1}^{\Lambda} n_i d_i$;
for $i = 1$ **to** Λ **do**
  for $j = i$ **to** Λ **do**
    if $i = j$ **then** `// intra edges for` V_i
      `EdgeSkipping`$(i, i, \frac{{d_i}^2}{S}, 1, \binom{n_i}{2})$;
    else `// inter edges between` V_i `and` V_j
      `EdgeSkipping`$(i, j, M_{ij}, 1, |C_i||C_j|)$;

Yet, another possible generative algorithm [31] is presented by Miller and Hagbeg, named after them **MH** algorithm. It accepts the vector w with the expected degrees of each node. It is in fact less efficient in terms of memory occupation compared to the previous **DG** algorithm, which takes the degree distribution as input. Moreover, in [27] is claimed that the time complexity of the **DG** algorithm is similar to that of **MH** algorithm but " lower overhead of **DG** algorithm leads to smaller constant associated with the time complexity and make the algorithm approximately three times faster than the **MH** algorithm". However, in the `c++` simulator we need to generate the expected degree sequence $\mathbf{w}$ in order to generate the Cung-Lu graph. It is more immediate to simply pass the generated sequence $\mathbf{w}$ to the Algorithm 8 instead of identifying the groups of nodes having the same expected degree. This decision is justified by the fact that the space complexity has not been a constraint in practice and the time complexity improvement of Algorithm 4 is not dramatic, the performance of the **MH** algorithm is more than sufficient for our purpose.

Algorithm 6: Chung-Lu graph generator MH algorithm

CLGenerator $(w_0, w_1, ..., w_{N-1})$
Input : list of weights $\mathbf{w} = (w_0, ..., w_{N-1})$ in decreasing order
Output : a Chung-Lu graph G with expected degrees per node equal to $\mathbf{w}$
Define ad empty set E;
$S \longleftarrow \sum_i w_i$;
for $n_i = 0$ **to** $N - 2$ **do**
    $n_j \longleftarrow n_i + 1$;
    $p \longleftarrow \min(\frac{w_{n_i} w_{n_j}}{S}, 1)$;
    while $n_j < N$ *and* $p > 0$ **do**
        if $p \neq 1$ **then**
            draw r uniformly at random in $(0, 1)$;
            $n_j \longleftarrow n_j + \lfloor \frac{\log(r)}{\log(1-p)} \rfloor$;
        if $n_j < N$ **then**
            $q \longleftarrow \min(\frac{w_{n_i} w_{n_j}}{S}, 1)$;
            draw r uniformly at random in $(0, 1)$;
            if $r < \frac{q}{p}$ **then**
                $E \longleftarrow E \cup \{n_i, n_j\}$;
            $p \longleftarrow q$;
            $n_j \longleftarrow n_j + 1$;

For the algorithm to work, the list $\mathbf{w}$ of weights needs to be sorted in descending order. The generating procedure is rather similar to the one for ER graphs, the difference resides in the fact that the link probability is not anymore a constant p but depends on the weights w_i of the end-nodes. This slightly changes the link selection procedure that anyways relies on the *edge-skipping* method to jump over the links which will not to be part of the final graph. However, this time the links selected by the edge-skipping procedure are only *potential* links because the link probability p_{ij} decreases as proceeding in the edge list but, the probability used for the `EdgeSkipping` method corresponds to the p_{ij} of the last *potential* node. Therefore, a binary decision needs to be taken on these potential links to decide whether to keep or discard them according to the actual p_{ij} of that pair of nodes.

Let us see it in more concrete terms. The algorithm begins by fixing one of the end-nodes of the link, starting from $n_i = 0$ and looping over the admissible values of $n_j = 1, 2, ..., N - 1$. Given the decreasing ordering of the weights in the vector $\mathbf{w}$ the probability $p_{n_i n_j} = \frac{w_i w_j}{\sum_k w_k}$ is monotonically decreasing with respect to n_j. This observation allows us to avoid to calculate $p_{n_i n_j}$ for every value of n_j in advance. At the beginning of each iteration of the `for` loop the input probability for the edge-skipping procedure (inside the `while`) is set to $p = p_{n_i,n_{i+1}}$ [5]. Lines 9 and 10 implement the edge-skipping method identifying a *potential* link. However, since using $p = p_{n_i,n_{i+1}}$ in the edge-skipping method corresponds to use an upper bound for the links probabilities due to the monotonically decreasing nature of these values, the link has to be rejected with probability $\frac{q}{p}$ where q is the actual link (n_i, n_j) probability computed from the weights $q = \frac{w_{n_i} w_{n_j}}{S}$. This rejection sampling procedure is illustrated in Figure 3.9.

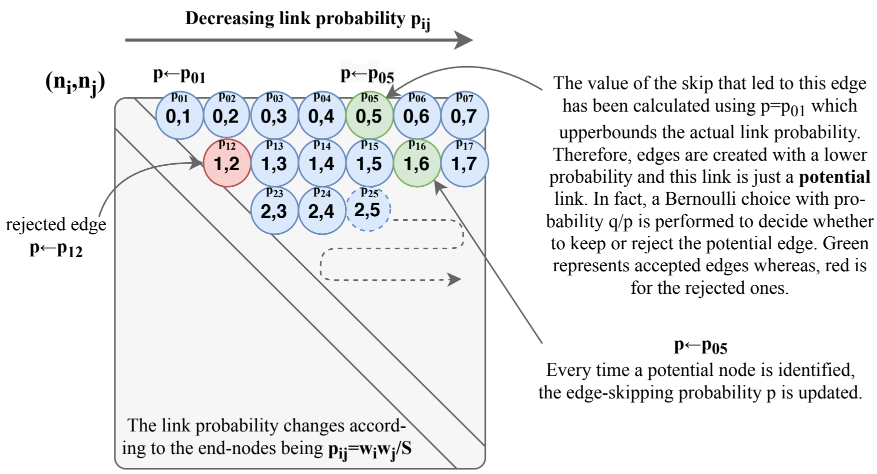


Figure 3.9: Rejection sampling procedure employed in the **HM** algorithm.

Then, p is updated $p = q$ and the algorithm keeps jumping to the next potential neighbour n_j. When the neighbours are finished, the outer `for` loop allows to consider another n_i value and the same procedure is performed, in order to define the edge list E of the Chung-Lu graph. The time complexity is $O(N + M)$, where N is the number of nodes and M the number of links, optimal for this class of problems.

[5] The fact we are not considering $p_{u,u}$ guarantees to avoid self-loops in the network, which is an undesirable property.

Chapter 4

Phase Transition in the Minority Opinion Fraction

This chapter continues from the end of Chapter 2 in which some initial results on the adaptive voter model proposed by Durrett et al. have been discussed. The metric of interest is the fraction of nodes holding the minority opinion at the end of the dynamical process (when an absorbing state is hit). Holme and Newman [23] were the firsts who observed a phase transition as a function of the rewiring, from a state with a giant component of like-minded agents to one with a highly fragmented network whose clusters hold a different opinion. It needs to be reminded that Holme and Newman studied a model in which the number of opinions G grows with the size of the network whereas, Durrett et al. [9] [25] considered just two possible opinions. As a preamble of our results, we will show the phase transitions of the two version of Durrett's adaptive model: *rewire-to-same* and *rewire-to-random* starting on an Erdős-Rényi graph.

4.1 Discrete and Continuous Phase Transition

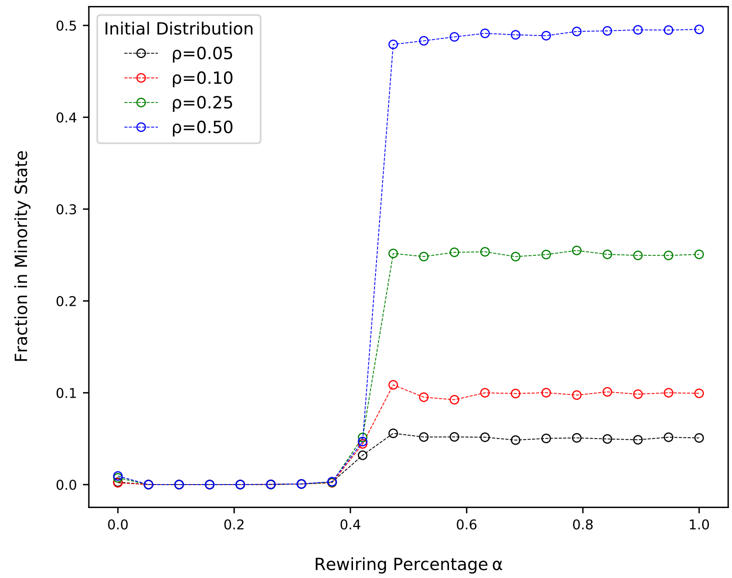
The dynamic process under study is exemplified in Figure 4.1 and is the adaptive voter model of Durrett et al. presented in Chapter 2. To understand the model's behaviour, it needs to be understood in all of its parts. The adaptive voter model arises from the interplay of two different processes alternatively selected during the dynamics. Which process is going to be executed is decided through a Bernoulli choice having parameter α, also referred as the *rewiring probability*. In fact, with probability α the network modifies its own structure. The model dictates to detach one end of a *discordant* link and to reconnect the other end in one of two possible ways, according to the version of the model. In the *rewire-to-same* it is attached to an agent of the network sharing the same opinion whereas, in the *rewire-to-random* to a random agent out of the $N-1$ possible. The first version incorporates only the idea of *homophily*, the second includes to some extent also the idea of *heterophily* even though it is not controlled by a parameter as it as been done in [22] but stochastically depends on the state of the nodes of the network. If the vast majority of the nodes hold the same opinion then, with high probability, a node will be rewired to one with the same opinion also in this *rewire-to-random* case. The other stochastic process involved in the dynamics is an *opinion-changing* process and corresponds to the one in the classical voter model presented in Chapter 1. An agent can update its own opinion consulting one of its neighbours $n_j \in N_{n_j}$.

Input: $G(A, E)$ random graph with mean degree $\delta = \frac{2L}{N}$
$A_i \in A$ has randomly assigned opinion $x_i(0) \in \{0, 1\}$
Output: $G(A, E)$, A_i has opinion $x_i(t)$, no *"discordant"* links exist

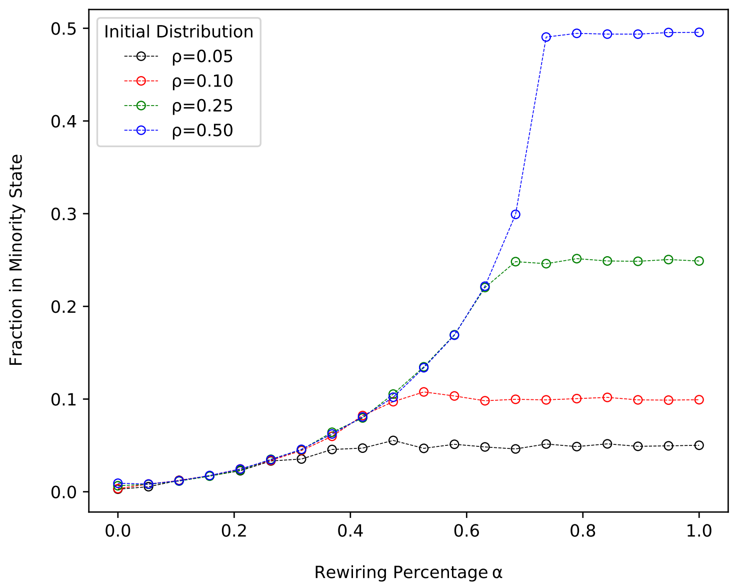
while *discordant* links are present **do**
▷ Random selection of a ***discordant*** link (A_i, A_j)
(1) With probability α
▷ disconnect the link (A_i, A_j)
▷ [RtS] connect A_i to a new agent A_k such that $x_i = x_k$
▶ [RtR] connect A_i to a new agent A_k such that $x_k \notin N_{A_i}$
(2) Otherwise, with probability $1 - \alpha$
▷ A_i copies the opinion of its neighbor A_j
end while

Figure 4.1: Adaptive voter model as proposed by Durrett et al. [9]

To qualitatively understand what to expect from the process one can observe what happens for the extreme values of the rewiring parameter $\alpha \in [0, 1]$. For $\alpha = 0$ the rewiring process is suppressed and therefore, only opinion-changes can occur. This case reduces to the classical voter model on a static graph that has usually been considered as being an Erdős-Rényi. It is know that, for N large enough, if the average degree δ of graph belonging to this class is greater than 1 there exist a giant component with high probability containing a fraction c of all the nodes of the graph. Moreover, the other clusters have sizes of the order of $O(\log N)$ being N the total number of nodes. Since for $\alpha = 0$ the network cannot change, the dynamics stops when each of the clusters in the network reach an internal consensus. The giant component has size cN, a fraction of the nodes so, the number of iterations needed to reach consensus for the voter model is quadratic with the size of the network therefore $O(N^2)$. On the other hand, for pure rewiring ($\alpha = 1$) none of the nodes can change opinion. Therefore, the absorbing state is reached only when the network rearranges its links in such a way that only like-minded agents belong to the same cluster. These clusters clearly depend on the initial distribution of the opinions in the network. In this case, it has been observed that the number of updates needed for consensus is of order $O(N \log N)$ [9]. This results is easy to explain for the *rewire-to-same* case. In fact, if no opinion change can occur and links are rewired just towards agents who share opinion then, the disconnection into agreeing clusters can be attained when all links have been rewired. This is equivalent to the *coupon collectors* problem which requires $O(M \log M)$ where M is the number of edges of the graph. We are dealing with sparse networks in which the average degree $\delta = O(1)$. The number of edges of an Erdős-Rényi graph is $M = \frac{N(N-1)}{2} \cdot p = \frac{N\delta}{2}$, being the average degree $\delta = (N - 1)p$ due to independence of link existence. This demonstrates why the complexity is $O(N \log N)$ in case of sparse graphs.



(a)



(b)

Figure 4.2: Phase transitions of the equilibrium fraction of nodes holding the minority opinion in function of the probability of rewiring α as observed in [9]. The data have been obtained by averaging over 10 realizations of the process over as many different Erdős-Rényi graph instances with $N = 10000$ nodes and average degree $\delta = 4$. The initial configuration of the opinions is random, each node is assigned its opinion from a Bernoulli random variable with parameter ρ. (a) refers to the rewire-to-same version and in (b) to the rewire-to-random.

The behaviour of the system between the extreme values of α strongly differs from the rewire-to-same case to the rewire-to-random. In the first instance, a discrete phase transition is observed from a state in which the fraction of nodes in minority opinion is close to zero to a state in which this value attests around the initial assigned value for the minority fraction ρ. The value of the threshold α_c is the same irrespective of the initial distribution ρ as can be easily seen in Figure 4.6a. Instead, for the rewire-to-random it can be observed a continuous phase transition which joints the two extreme situations. What is interesting in this case is that for values of the rewiring α below the critical threshold α_c the chart follows a curve that Durrett et al. have called *universal* curve. It is also worth noting that the value α_c [1] depends on the initial distribution ρ. These observations have been made from the results in Figure 4.6a and Figure 4.9b, obtained from simulations carried out with the `c++` simulator described in Chapter 3.

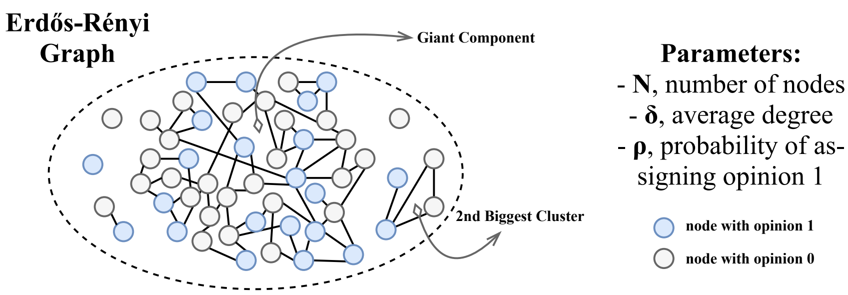


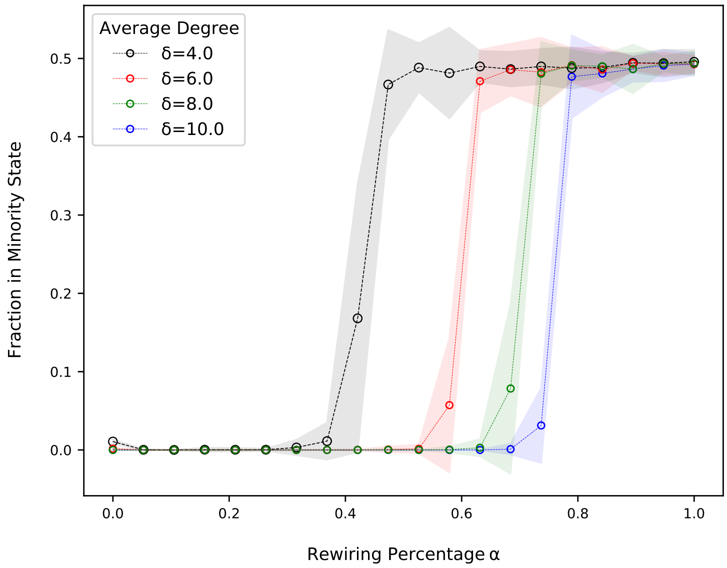
Figure 4.3: Erdős-Rényi random graph with its giant component highlighted. The degrees of freedom for the simulations are listed.

The obvious parameter of the system is the amount of rewiring α but, the others degrees of freedom of the system come from the initial configuration of the network. First of all, in most of the literature and also in the work of Durrett et al. the Erdős-Rényi generative model has been employed. The first graph's parameter is the number of agents N, it mainly impacts the time to convergence. Then, there is the average degree δ, equal for all nodes in Erdős-Rényi graphs. It will be seen in the next section that the average degree affects the value of the threshold α_c. δ is related to the edge probability p as follows: $p = \frac{\delta}{N-1}$. The last system's parameter is the probability ρ of assigning opinion 1 to a node, this also influences the behaviour of the system, in fact the fraction of nodes in the minority opinion for values of α above the threshold α_c attests to this value.

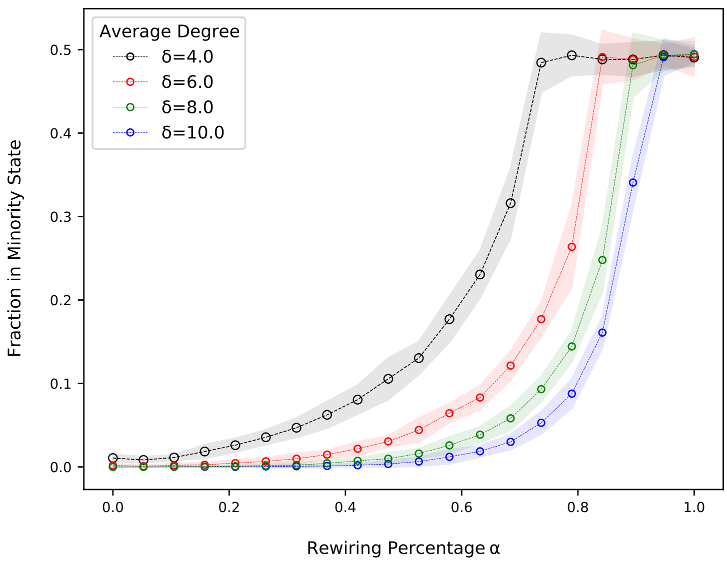
4.2 Threshold Dependence on the Mean Degree

It is very important to understand which are the metrics which influence the dynamics because it will be seen later that the configuration of the initial network unsurprisingly does not affect the dynamics in a meaningful way. However, variations in the mean degree of the initial graph greatly affect the behaviour of the process.

[1] The value after which the fraction of nodes holding the minority opinion attests around ρ.



(a)



(b)

Figure 4.4: Fraction of nodes in the minority state as a function of the rewiring α starting from an Erdős-Rényi graph with N=4000 nodes and varying average degree δ. Each point is obtained by averaging over 10 different realizations of the process and the initial graph. The colored area around the points represents a confidence interval of $\pm 3\sigma$ around the mean value. (a) Rewire-to-same adaptive voter model. As the degree increases, the threshold α_c also increases. (b) Rewire-to-random adaptive voter model. In this case the system exhibits a continuous phase transition and we can observe a shift towards right of the curves as the degree increases.

This behaviour is confirmed by results for example of Kimura and Hayakawa [22]. Studying the phase transition of the Home and Newman [23] model they found by means of pairwise approximation that the critical value can be estimated as $\alpha_c = 1 - \frac{1}{\delta}$ where δ is the average degree of the network. Then, by performing calculations including higher-order structures rather than simply the pairwise approximation, they obtained a more precise bound (better fitting the data from simulations) $\alpha_c = 1 - \frac{\sqrt{3}}{\delta}$. The Holme and Newman model comprises G different opinions, the results of Kimura and Hayakawa have been derived for $G = 2$ that coincide with the binary opinion space of the adaptive voter model. The model is in fact equivalent to the rewire-to-same of Durrett et al. so, these results support the observed relation between the average degree δ and the critical value of rewiring α_c. It has to be noted that this estimate does not closely fit the data from simulations and comparing it to the results we have obtained, it is more an upper bound of the threshold value α_c.

Also Durrett et al. [9] discussed the dependence of the threshold α_c on the mean degree δ however, in this case it regarded the rewire-to-random version of the model. By using pair approximation they found that:

$$\alpha_c(\rho) = \frac{\delta - 1}{\delta - 1 + \rho^2 + (1 - \rho^2)}$$

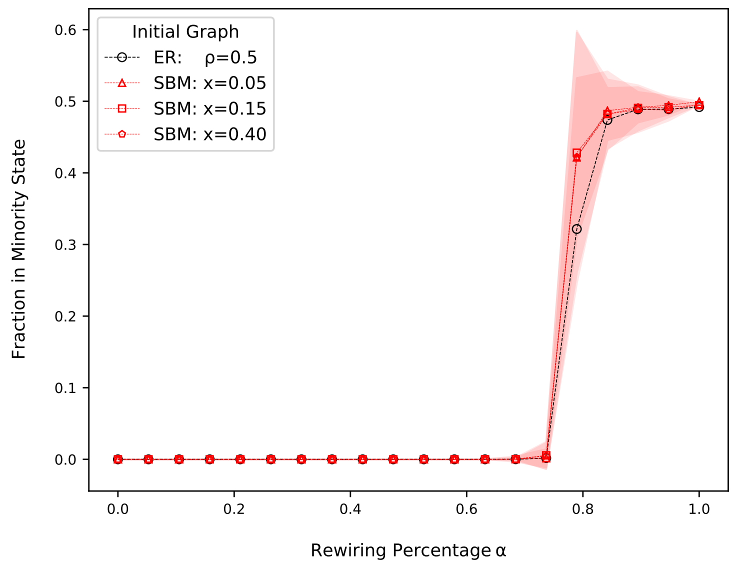
We have already observed that the value of the threshold depends on the initial distribution of opinions ρ in the rewire-to-random model. And without a surprise the formula just presented depends on ρ. Durrett et a. argued that this method "drastically overestimates the value α_c [9] and by applying another framework, the AME (approximated master equation), it is possible to obtain much better results. However, the AME appears to be more complicated and requires to solve numerically differential equations to obtain predictions.

4.3 Threshold Dependence on the Starting Network

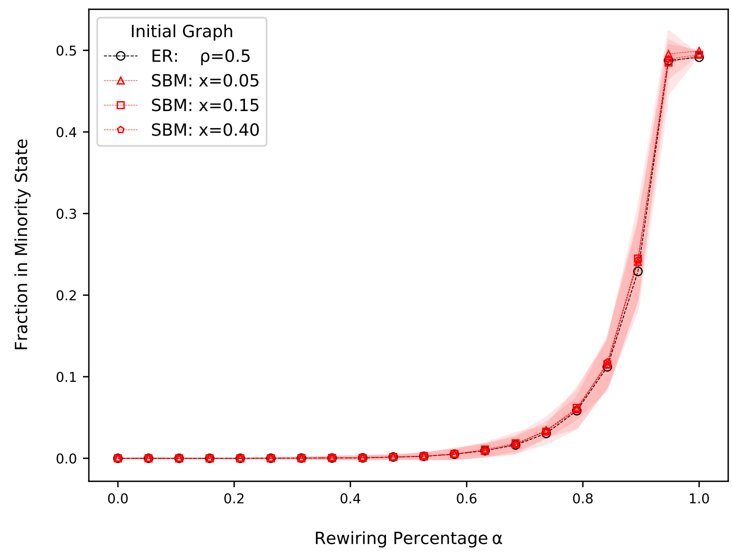
The results presented in the previous sections have been obtained staring from an Erdős–Rényi (ER) graph with a given average degree δ. It is of interest to explore the dynamics of the adaptive voter model starting from other initial graph configurations so, employing diverse generative models.

4.3.1 Stochastic Block Model

We decided to start by investigating the dynamics of the process on the stochastic block model (SBM) [32]. This model has already been presented in Chapter 3. The main property to be recalled is that SBM random graphs present a community structure. Each community can be considered as being an Erdős–Rényi graph. The nodes of each community are then connected with each other according to a matrix $\mathbf{M}$ which contains the probabilities of connection among the various communities. We limit ourselves to the study of stochastic block models with two communities weakly interconnected and with the same density of links within each community.

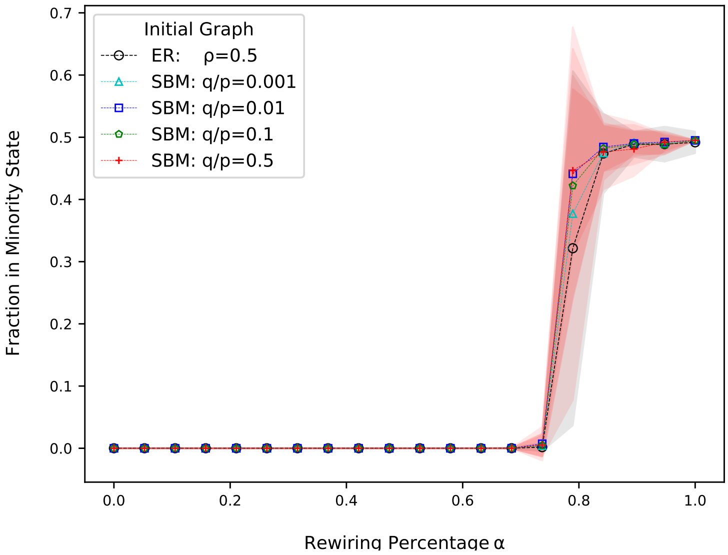


(a)

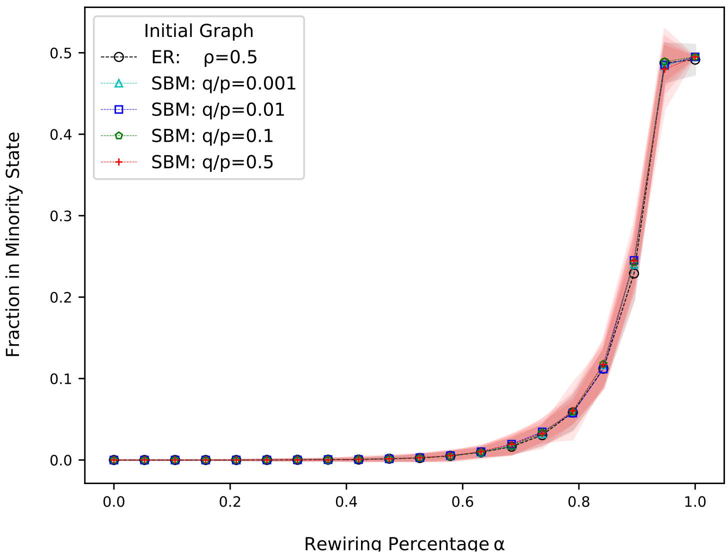


(b)

Figure 4.5: Fraction of nodes in the minority state as a function of the rewiring α and the initial bias x, starting from a stochastic block model (SBM) with N=2000 nodes, 2 communities with intra-edge probability equal to p and inter-edge probability to q and average degree $\delta = 12.0$. Each point is obtained by averaging over 10 different realizations of the process and the initial graph. The colored area around the points represents a confidence interval of $\pm 3\sigma$ around the mean value. The results of the SBM (in red) are compared to those of an Erdős–Rényi graph (in black) with the same number of nodes N and average degree δ. (a) Rewire-to-same and (b) rewire-to-random model.



(a)



(b)

Figure 4.6: In the exactly same setting as for Figure 4.5 it is shown the dependence with respect to the ratio of inter and intra edge probabilities $\frac{q}{p}$. Fluctuations around the mean appear to be slightly higher however, no meaningful influence on the system's dynamic is noticed by varying the parameter $\frac{q}{p}$. (a) Rewire-to-same and (b) rewire-to-random model.

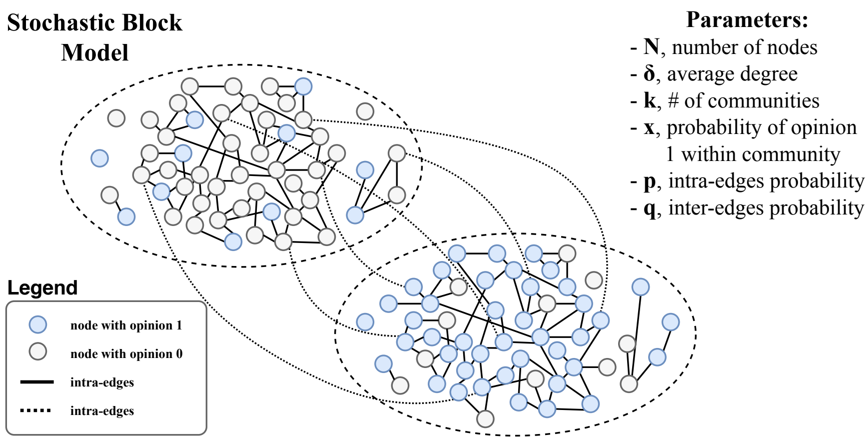


Figure 4.7: Stochastic Block Model. The parameters of the graph have been listed.

We simulated the adaptive voter dynamics on SBM graphs with N nodes and $k = 2$ equally-sized communities. Some of the degrees of freedom for the simulations are the same as the ER case namely, the average degree δ of the network, the percentage of rewiring α and the initial distribution of the opinions. Regarding the latter, we now have two distinct communities each one could be assigned with a different opinion distribution. It has been decided to maintain an overall balance of the opinions in the network, namely half of the N nodes will have opinion 1 and the other half opinion 0 initially. What will be changed is the internal bias of opinions within the community. A parameter x has been defined as the probability that a node of the first community has opinion equal to 1. The second community will have the converse bias, therefore the probability of a node of having opinion equal to 1 is $1 - x$. The other parameters which have to be discussed are the connection probabilities. We are considering graphs with two communities and we suppose that the communities interchangeable so, equally-size and equally-dense. So, the number of probabilities from the matrix $\mathbf{M}$ to be considered the values reduces to two. The intra-edge probability p, probability of existence of an edge within a community, and the inter-edge probability q, for links connecting the two communities. To keep the number of simulation parameters reasonable we considered the ratio between these two values $\frac{q}{p}$ and since we are interested in networks in which the communities are internally densely connected and only loosely connected with others, this ratio will always be smaller than 1. It is clear that, in the case of $\frac{q}{p} = 1$ there is no more distinction between the communities and a link in the network has probability of existence equal to p (it is p towards its own community and $q = p$ towards the other community so, overall it is p). Such a choice of the parameters would lead to an Erdős–Rényi graph.

In Figure 4.5 and 4.6 are presented the results for the fraction of nodes in the minority opinion for different values of initial bias x and probabilities ratio $\frac{q}{p}$. Surprisingly, the community structure of the initial network does not affect the fraction of nodes in the minority state. Before running the simulations, we expected that loosely coupled communities would have speed up fragmentation into two separate clusters however, this does not happen. The reason behind this behaviour might be found in the random choices taken during the rewiring phase. In fact, in both instances of the model, when a link needs to be rewired it then is connected to a random nodes that can be either belonging to the same community or to the other one. This helps to destroy the community structure and make the system behave as if the initial configuration was an ER graph.

4.3.2 Scale-free Chung-Lu Graphs

The results presented in the previous section show an insensitivity of the adaptive voter model with respect to the initial graph structure. This needs to be further investigated also because the stochastic block model and the Erdős–Rényi model produce very similar networks, all nodes have the same mean degree and the only difference is brought by the community structure of the SBM. Therefore, it has been decided to run the dynamics starting from a Chung-Lu random graph with scale-free degree distribution. This is a feature commonly observed in real-world complex networks. One of the consequences of it is the emergence of some nodes with very large degree, called *hubs* as shown in Figure 4.8.

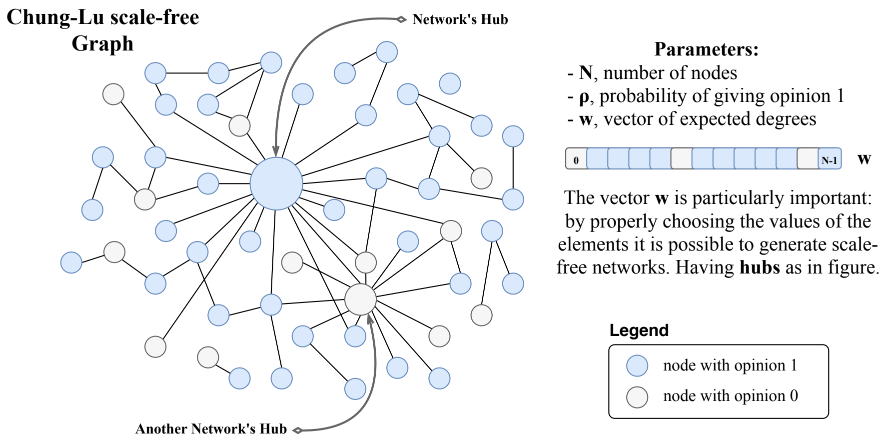


Figure 4.8: Chung-Lu graph together with its parameters. Hubs are highlighted in the network.

To generate the graph we used the procedure described in Chapter 3. It is of particular importance the method to generate the sequence of expected degrees **w** in order to obtain a scale-free distribution. First of all, the method [26] allows us to produce a sequence **w** which leads to a graph with a prescribed expected degree d and an expected maximum degree $M = n^c$ (degree of the largest *hub*). The method produces better results for γ (parameter of the scale-free distribution) greater than 3. However, we want to keep the exponent γ between 2 and 3 so, the parameter is chosen to be $\gamma = 2.9$. Then, choosing $c = 0.8$ we prescribe that the network will have a large hub. For a network of $N = 2000$ nodes, the prescribed maximum degree would be $N^{0.8} = 437.34$ and it can be empirically observed that the maximum degree of the generated Chung-Lu is $M_{sim} = 437$. The mean degree d is instead only approached from below, calculating it empirically it can be obtained a value around $d_{sim} = 11.8$,

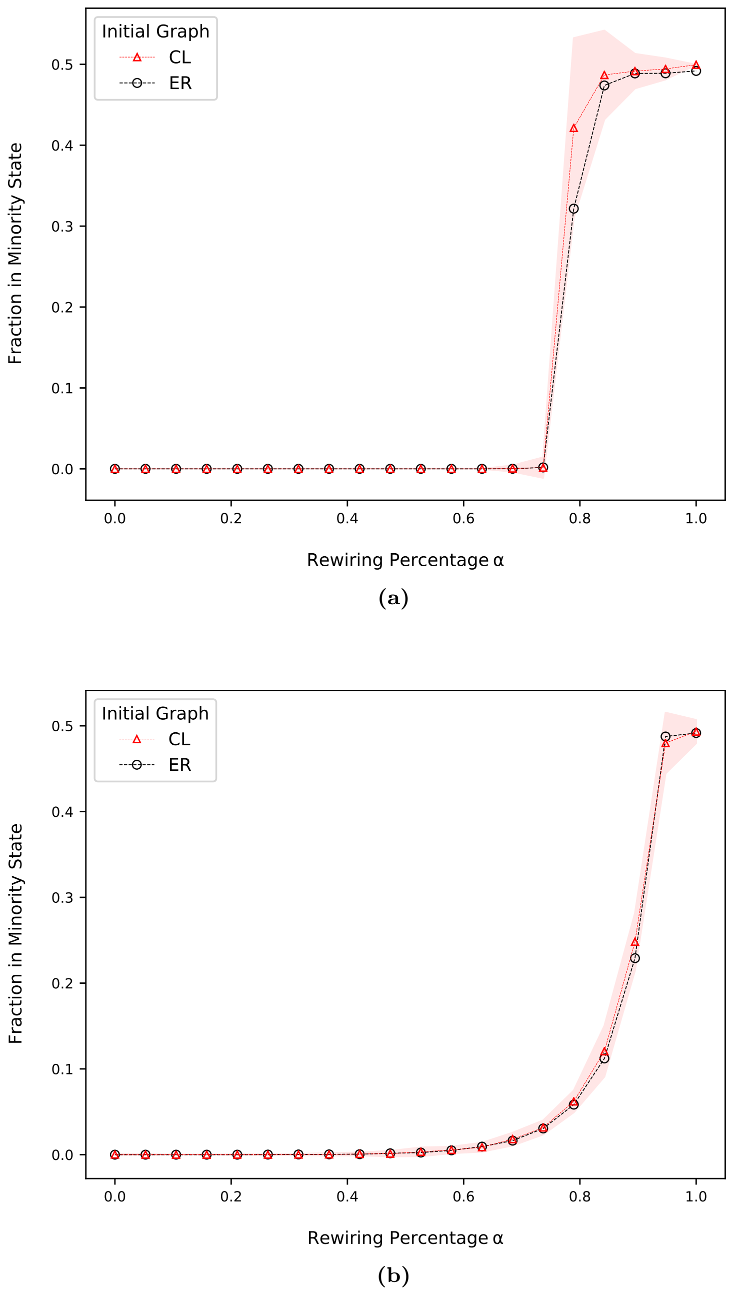


Figure 4.9: Fraction of nodes in the minority state as a function of the rewiring α starting from a Chung-Lu (CL) scale-free graph with N=2000 nodes, $\gamma = 2.9$ as the degree distribution exponent, average degree $\delta = 12.0$ and maximum expected degree equal to $N^0.8$. Each point is obtained by averaging over 10 different realizations of the process and the initial graph. The colored area around the points represents a confidence interval of $\pm 3\sigma$ around the mean value. The results of the CL (in red) are compared to those of an Erdős–Rényi graph (in black) with the same number of nodes N and average degree δ. (a) Rewire-to-same and (b) rewire-to-random model.

acceptable for our purpose. It is important to keep the average degree d close to the one of the ER and SBM case in order to make a comparison between them, since the threshold α_c is strongly influenced by the mean degree. What has been found comes again as a surprise. In fact, as can be seen from Figure 4.9 the behaviour of the system with a Chung-Lu graph is the same as the one with an Erdős–Rényi graph.

Algorithm 7: DFS

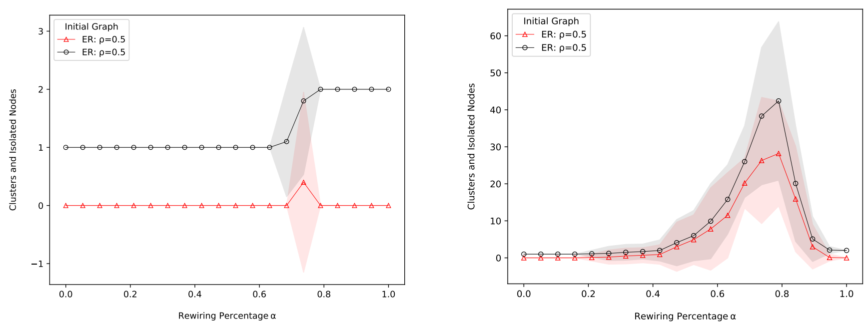
findClusters $(G(V,E))$
Input : graph $G(V,E)$
$V = \{0, 1, ..., N-1\}$
$E = \{(i,j)\}_{i,j \in V}$
Define ad empty set `IDvector`;
$id \longleftarrow -1$;
for $n_i = 0$ **to** $N-1$ **do**
  if n_i *is* ***not*** *visited* **then**
    $id \longleftarrow id + 1$;
    `DFS-Visit`(n_i, N_{n_i});

Algorithm 8: DFS-Visit

DFS-Visit (n_i, N_{n_i})
Input : node n_i
neighborhood N_{n_i}
`IDvector`$[n_i] \longleftarrow id$;
for n_j ***in*** N_{n_i} **do**
  if n_j *is* ***not*** *visited* **then**
    `DFS-Visit`(n_j, N_{n_j});

4.4 Fragmentation in Function of the Rewiring

The `c++` simulator collects various metrics of the process, not only the fraction of nodes in the minority opinion. For instance, the number of network's clusters when the absorbing state is reached. To do so, the graphs is explored using depth-first search and each time a connected component is identified it is assigned an integer label, the pseudo-code is provided above.



(a) (b)

Figure 4.10: Number of clusters at the end of the dynamics in function of the rewiring α for an Erdős–Rényi graph with $N = 2000$ nodes. Data points averaged over 10 realizations. (a) rewire-to-same version, the trend is as one might expect (b) rewire-to-random case, it can be seen a consistent spike in the number of clusters and for values of the rewiring close to $\alpha = 1$ the network results to be rather fragmented, with many isolated nodes.

On the one hand, for the rewire-to-same model the number of clusters is in line with what one would expect, with one cluster [2] for small values of the rewiring α and two clusters while approaching the pure rewiring $\alpha = 1$ as reported in Figure 4.10a. On the other hand, for the rewire-to-random model, a remarkable spike in the number of clusters (and isolate nodes) can be observed from Figure 4.10b. This means that the process profoundly changes the nature of the network. In fact, looking at social networks it is not customary to observe networks with an high fraction of isolate clusters (or nodes).

In Figure 4.11 are shown the curves regarding the number of clusters for all the generative graph models considered. In all the instances it is possible to observe a spike associated with a fragmentation of the underlying network. Again, also this result seems not to be influenced by the initial graph configuration. Therefore, it brings further evidence to the claim that the adaptive voter model is insensitive to the initial graph model.

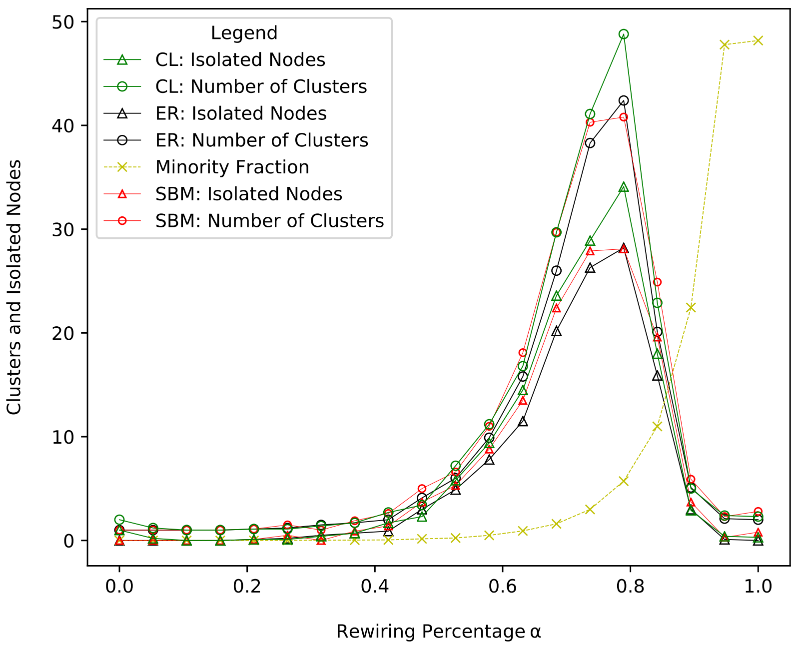


Figure 4.11: Number of clusters in function of the rewiring α for different initial graph configurations. All the graphs have $N = 2000$ nodes and average degree $\delta = 12.0$. On the graph it is superimposed the scaled version of the minority opinion fraction, to observe at which point happen the fragmentation. The figure is only concerned with the rewire-to-random version of the adaptive voter model.

[2] With our choice of the mean degree $\delta = 12.0$ we generate with high probability graphs that are connected.

Chapter 5

Convergence Time

When addressing the time to convergence of voter-like models on complex graphs it is important to carefully specify the fusion rule. In fact, minor variations which have no impact on regular graphs can have a dramatic impact on the time complexity of the process. In this chapter will be presented some literature results of the *classic* voter model at first. Then, the usual adaptive voter model is considered, showing the dependence of the consensus time as a function of the size of the network N and the rewiring probability α.

5.1 Complexity of the *classic* Voter Model

The voter model has been largely studied in the literature, firstly on regular lattices and then on heterogeneous graphs. Some results will be presented regarding both situations. These are in line with what we have found for the adaptive voter model and can be used to validate our findings.

5.1.1 Voter model on regular lattices

The *classic* voter model dictates to randomly select an agent A_i at each time step and then pick up one of its neighbours A_j, if any exists. Agent A_j influences agent A_i so that $x_i \longleftarrow x_j$. When starting from a disordered initial condition, as in usual coarsening process the dynamics tends to increase the order of the system [33]. Since no bulk noise is considered in the model, the states in which all agents hold the same opinion are absorbing for the process. Initially, the model has been investigated on regular lattices in order to understand whether full consensus can be reached on an infinite-sized system. The process on the one-dimensional lattice coincide with the zero-temperature Glauber dynamics. For the solution in any dimension d we report the results shown in the review done by Castellano et al. [33] who refer to the work of Frachebourg and Krapivsky [34]. The behaviour of the process is determined by the number of active interfaces namely, the number of links between agents who hold different opinions. If this number goes down to zero, it means that full consensus has been achieved in the network. Therefore, the temporal behaviour of this quantity $n_a(t)$ is of primary interest.

The asymptotic behaviour of the density of active interfaces $n_a(t)$ has been found to be described by the following by the equations:

$$n_a(t) \sim \begin{cases} t^{-\frac{2-d}{2}} & d < 2 \\ \frac{1}{\ln(t)} & d = 2 \\ a - bt^{-\frac{d}{2}} & d > 2 \end{cases} \tag{5.1}$$

These equations demonstrate that for $d \leq 2$ the model undergoes a coarsening process which leads to consensus in the graph. Whereas, for $d > 2$, there is a finite density of interfaces asymptotically meaning that an infinite system does not reach a consensus state. This is to what concerns infinite systems, in finite systems the voter model always reaches asymptotically a consensus since it represents an absorbing state for the dynamical process. Cox [35] demonstrated that the time to consensus depends on the system size N, the number of nodes in the network. He showed that:

$$t_{cons} \sim \begin{cases} N^2 & d = 1 \\ N \log N & d = 2 \\ N & d > 2 \end{cases} \tag{5.2}$$

In the previous chapters the time has been measured as the number of iterations (or updates) in the dynamical process. However, in the literature of the voter model, one time unit is considered as being the interval of time in which N agents (N being the size of the graph) have been updated. Pass from one formulation to the other it rather straightforward: for instance, considering the time complexities above, it is sufficient to multiply the values presented by N to have the values in terms of number of iterations. The results that have been stated just hold for static networks and more importantly, for regular networks. The voter model has been largely studied also on heterogeneous networks. This is crucial since real-world networks are far from being regular, the next section is devoted to the discussion of the voter model on these graphs.

5.1.2 Voter Model on Heterogeneous Networks

Complex networks, are better suited to describe social networks since they capture some key characteristics observed in populations of individuals. In such networks the nodes are profoundly different from one another. For instance, whereas in regular graphs all the node in the network have the same degree, in scale-free networks the degree distribution follows a power-law and there exist some large-degree nodes called *hubs*, which differentiate from the rest of the nodes in the network. This disparity in the degree of the nodes make the voter model behave differently on heterogeneous graphs compared to regular graphs. Moreover, slight modifications in the fusion rule, which have no effect on homogeneous graphs, have major effects on the dynamics on complex graphs. Therefore, it is important to distinguish between at least two classes update rules: *node-update* and *link-update*, summarized in Figure 5.1. Regarding the first class, we have already presented the two possible instances: the *direct* voter model and the *reverse* (also called *invasion process*) voter model. The adaptive model of Durrett et al. falls in the class of link-update voter models since at each time step a discordant link is picked up. In the classical formulation a random edge is chosen from the edge list E and only if the end nodes posses opposite opinions one of the two is forced to adopt the opinion of the other. Selecting discordant edges increases efficiency and leaves the stochastic process unchanged.

A motivation for the difference between the two mentioned categories can be found in Suchecki et al. [36]. It must be observed that now we are considering the dynamics on finite graphs, this implies that the system will always reach one of the absorbing states of the process (consensus), the time to reach consensus can also be referred to as survival time. Suchecki et al. argued that the survival time "scales linearly with the system size only when the updating rule respects the conservation law of the average magnetization", this conservation refers to an ensemble average and is not a elementary step conservation in the sense of the Kawasaki dynamics. For the voter model on regular lattices of general dimensionality the global magnetization (average spin) is conserved in the thermodynamic limit of large systems. However, on networks (e.g. scale-free graphs) in which nodes have strong heterogeneity in the degree distribution only link-selection voter models guarantee the conservation of the average magnetization whereas, with node-selection models this does not happen [36]. They reference [37] in which it has been studied the voter model on small-world graphs, finding a linear behaviour of the time to consensus as a function of the system size N and therefore agreeing with the behaviour on regular lattices.

However, Suchecki et al. found that the survival time of a node-selection voter model on Barabasi-Albert graphs (which are the simpler graphs exhibiting a power-law degree distribution) scales as $N^{0.88}$. To sum up, if the global magnetization is conserved, the voter model on an heterogeneous graph behaves as the voter model on a lattice whose dimension $d > 2$ and the survival time scales with N as can also be seen in the system of equations 5.4. In a later work [38], Suchecki et al. further analyzed the voter model on different classes of complex networks finding that the effective dimensionality, the network disorder and the degree heterogeneity determine if the voter model orders the system.

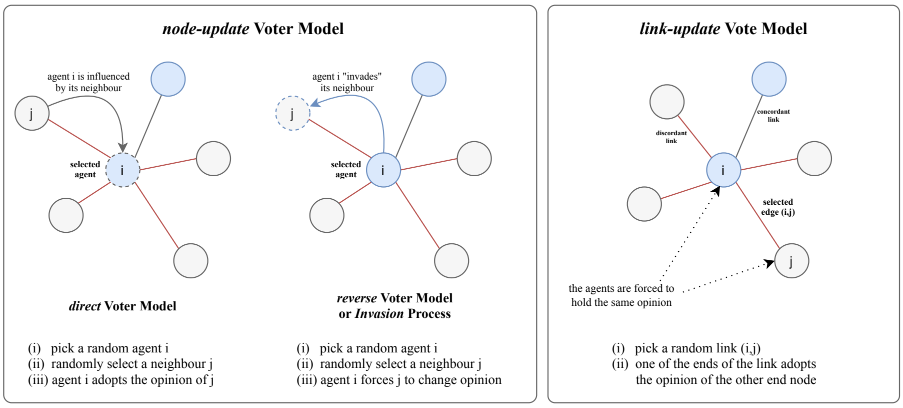


Figure 5.1: Representation of the three different update rules, one can differentiate between *node-update* and *link-update* rules. In red are shown the discordant links whereas, in grey the concordant one. An opinion change happens only when a discordant link is selected.

Sood et al. following the above work found conservation laws for all the three instances of the model, together with the exit probability (see [20] for the details). In addition, it has been provided a formula for the asymptotic behaviour of the time to consensus of the *direct* voter model on a graph with power-law degree distribution ($n_k \sim k^{-\nu}$):

$$t_{cons} \sim \begin{cases} N & \nu > 3 \\ \frac{N}{\ln N} & \nu = 3 \\ N^{\frac{2(\nu-2)}{\nu-1}} & 2 < \nu < 3 \\ (\ln N)^2 & \nu = 2 \\ \mathcal{O}(1) & \nu < 2 \end{cases} \tag{5.3}$$

And also for the asymptotic behaviour for the *reverse* voter model on heterogeneous graphs:

$$t_{cons} \sim \begin{cases} N & \nu > 2 \\ N \ln N & \nu = 2 \\ N^{3-\nu} & \nu < 2 \end{cases} \tag{5.4}$$

It is interesting to observe how for $\nu > 3$ (*direct* voter model) and for $\nu > 2$ (*reverse* voter model) the voter model on a scale-free network behaves asymptotically as a lattice whose dimension $d > 2$.

5.2 Effect of the Adaptivity on the Complexity

In this chapter we consider the adaptive voter model as presented by Durrett et al. and we demonstrate how the time complexity of the process changes in function of the rewiring α. Of course for $\alpha = 0$ we obtain the *classic* voter model with a link-update rule.

We start off by showing a temporal [1] plot of the fraction of nodes in minority opinion. The graph has been obtained by fixing the rewiring probability α and is in accordance with what has been found by Durrett et al. [9]. After an initial phase in which the values remains around 0.5 the value start fluctuating until it reaches a value close to 0 (here we are considering the rewire-to-same adaptive voter model under the critical threshold α_c). Durrett et al. demonstrated that the fraction of nodes in the minority state are correlated to the number of active links in the network. The shape of the curves is in agreement with the results regarding the *classic* voter model presented in [38]. As for the *classic* voter model a rapid transient occurs which leads the system to a meta-stable partially ordered state and then the curve oscillates around an average value until a finite size fluctuation takes the process in one of the two absorbing states [38]. From the figure it can be already seen that the time to equilibrium is highly variable and that realizations can strongly differ one from each other.

[1] From now on when talking about time we mean an elementary step of the voter dynamics in Figure 5.6. This corresponds to $\frac{1}{N}$ times unit as intended in the previous sections.

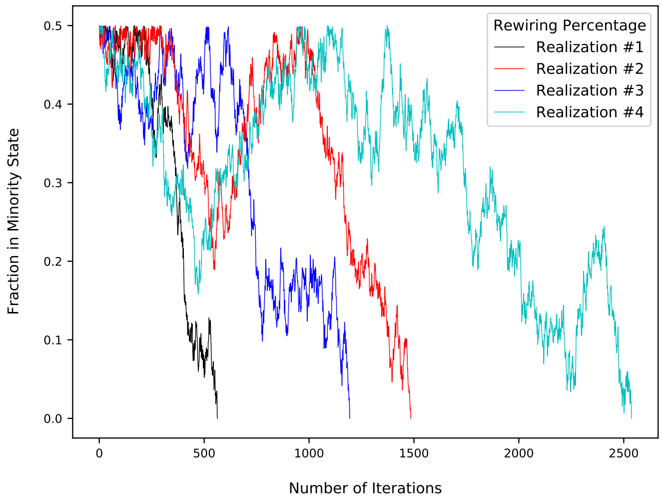
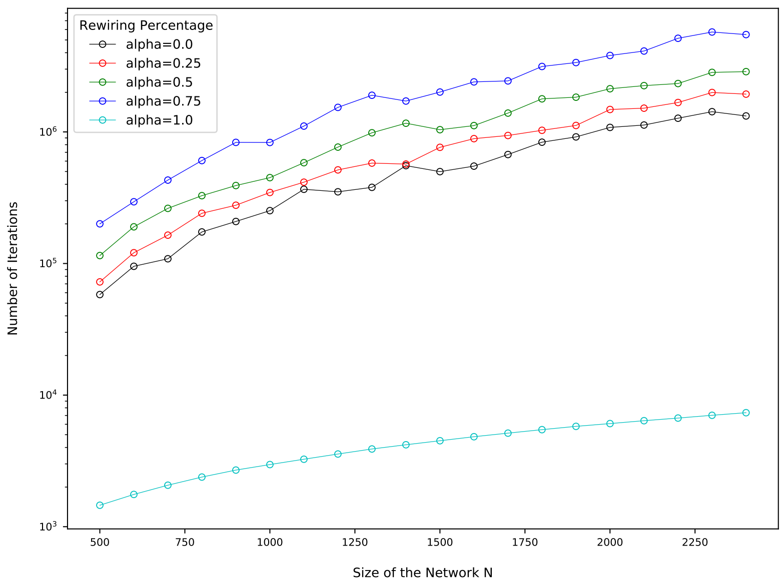
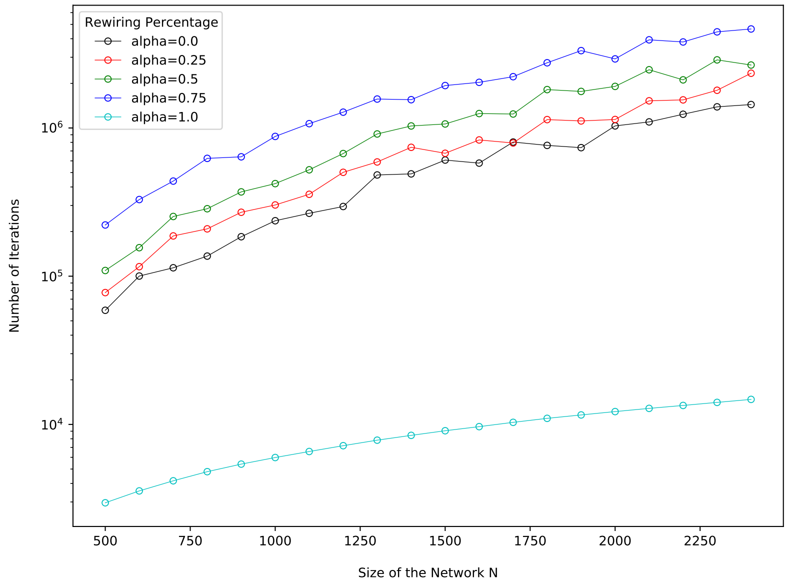


Figure 5.2: Time evolution of the fraction of nodes holding the minority opinion as a function of the time (a time unit is considered to be an elementary step, either a rewiring or an opinion-change). The initial graphs are Erdős-Rényi with $N = 2000$ nodes, average degree $\delta = 12.0$ and opinion density $\rho = 0.5$. These results belong to the rewire-to-same version of the model however, the rewire-to-random shows an analogous behaviour. Note: every time unit corresponds to 500 elementary steps, the process has been sampled every 500 iterations.



(a) Rewire-to-Same

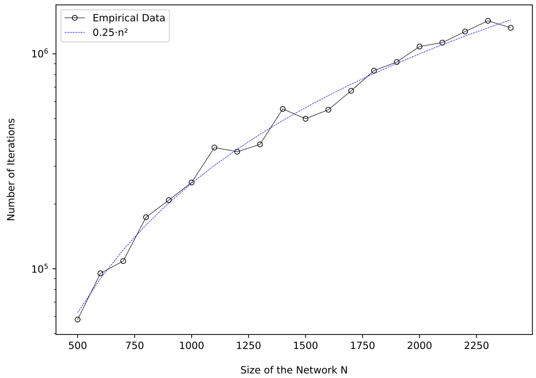


(b) Rewire-to-Random

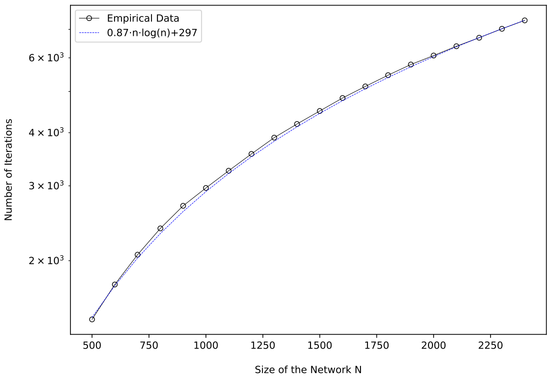
Figure 5.3: Number of iterations to reach an absorbing state in function of the network size N. Every point has been obtained averaging over 100 realizations. Curves for different values of the rewiring α are displayed, it can be seen that these follow the trend showed above: the complexity gradually increase with the α and then drops for values of the rewiring exceeding the critical value α_c. Both versions of the adaptive voter model have been investigated (a) rewire-to-same and (b) rewire-to-random, leading to similar results.

In Figure 5.3 the number of iterations to consensus is plotted in function of the size N of the network for different values of α. As it can be seen, the difference in complexity is substantial, from $t_{cons} \sim N^2$ for $\alpha = 0$ to $t_{cons} \sim N \log N$ for alpha approaching $\alpha = 1$. To support this claim, in Figure **??** we have provided a fit to a quadratic function for the case $\alpha = 0$ and to a $N \log N$ function for $\alpha = 1$. This agrees to what has been discussed in Chapter 4 regarding the behaviour of the model. Moreover, the results obtained for $\alpha = 0$ agree to what has been stated above for the *classic* voter model indeed, the quadratic scaling in the number of iterations coincides with the linear increase in the time as seen in the *classic* model.

The two versions of the adaptive model namely, the rewire-to-same and the rewire-to-random have a rather similar behaviour in terms of the time to reach consensus. Contrary to what we have seen regarding the phase transition in the fraction of nodes in the minority state, in this case the difference in the models does not play a crucial role. For this reason, we have reported curves interpolations only for the rewire-to-same case, for the rewire-to-random the results are completely analogous.



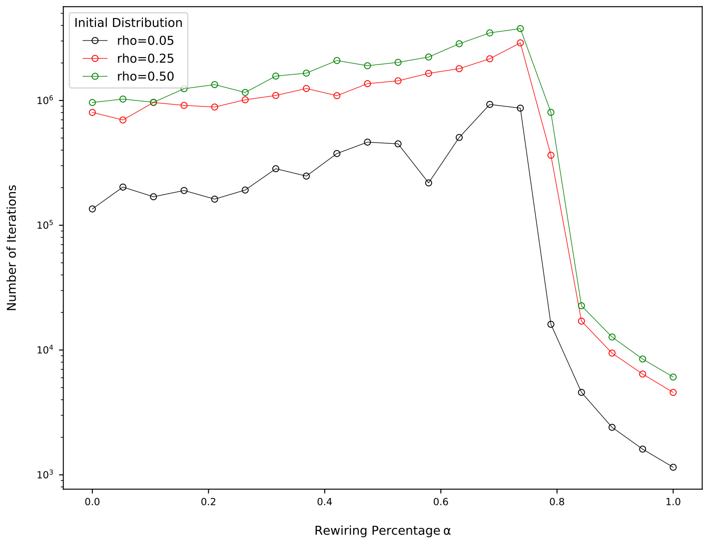
(a) $\alpha = 0$



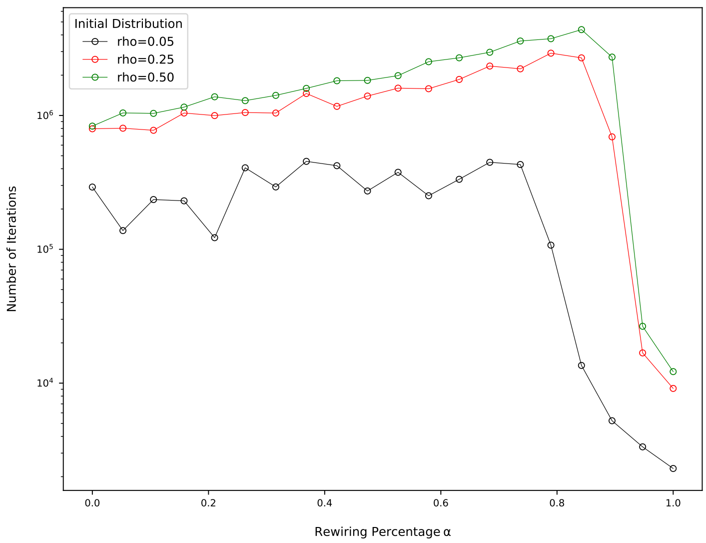
(b) $\alpha = 1$

Figure 5.4: Curve interpolations for the time complexity of the rewire-to-same adaptive voter model for the two extreme cases $\alpha = 0$ and $\alpha = 1$. In (a) the data have been interpolated with a N^2 function and in (b) with a $N \log N$ function. The data are those presented in Figure 5.3

It is interesting to observe the dependence of the time to consensus from the probability of rewiring α. Figure 5.5b shows data obtained from simulations averaging over 100 realizations. The number of iterations is rather consistent until a certain value around $\alpha = 0.8$ and then drops (abruptly in the rewire-to-same case) by a couple of order of magnitude. This graph needs to be compared to that of the fraction in minority state in Figure 4.5 looking at the Erdős-Rényi results. It is clear that the number of iterations drops in correspondence of the phase transition. An explanation to this phenomena can be provided. When the rewiring α exceeds the threshold value α_c then, the rewiring process is stronger than the opinion-changing process. As already discussed, the rewiring process can be seen as a coupon collector's problem which has complexity $\mathcal{O}(N \log N)$ and this motivates the behaviour above the threshold. On the other hand, when α is below α_c the pure voter process rules, it has already been observed several times that this process runs inherently in $\mathcal{O}(N^2)$.



(a) Rewire-to-same



(b) Rewire-to-random

Figure 5.5: Number of iterations to consensus as a function of the rewiring probability α. Results obtained starting from an Erdős-Rényi graph with $N = 2000$ nodes, average degree $\delta = 12.0$ and variable initial opinion density ρ. Every point in the graph has been obtained averaging over 100 realizations of the process. (a) Rewire-to-same and (b) rewire-to-random model.

5.3 Uniform Discordant Node Selection

In the previous section it has been shown how does the survival time of the process change as a function of the rewiring α. This dependence has already been observed for node-selection models in [21] by Nardini et al, who studied the *direct* and *reverse* voter model. We have investigated the link-update case by studying the adaptive voter model of Durrett et al. In their work, Nardini et al. found even a stronger change in the scaling as the one we observed, from N^2 for small values of α to $N \log N$ for large values of the parameter. In fact, for the *reverse* voter model the complexity grows exponentially with the system size increasing the value of α. Instead, the time convergence of the *direct* voter model is favored by the rewiring. Indeed, the scaling behaviour is analogous to the one we have observed for the model of Durrett et al., going from N^2 to $N \log N$ [2]. One of the main points made in [21] is that slight modifications in the fusion rule might lead to remarkable changes in the process' behaviour. And this is motivated by the fact that the effect of the rewiring on the direct and the reverse voter model is opposite. In the first case convergence is favoured instead, in the second case it is hindered. Also the studies [20] [36] of the classic voter model on heterogeneous graphs evidenced the sensitiveness to formulation of the updating rule.

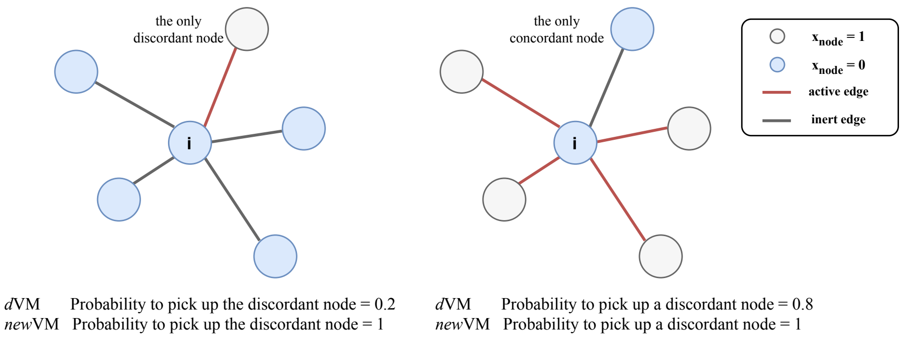


Figure 5.6: The difference between the proposed model and the classic voter model is explained through an example.

We have studied a modification of the direct voter model and further confirmed this characteristic. The proposed model shares the same setting as the direct voter model and just the neighbour's selection is modified. At each elementary step the process randomly selects a discordant node (a node with at least a disagreeing neighbour). Then, a neighbour is chosen between those who hold a different opinion, instead of any random neighbour as in the classic formulation of the model. The difference with respect to the dVM is that in the new model an opinion change will occur surely, irrespectively of the number of agreeing neighbours. Whereas, in the classic formulation, if a node has many agreeing neighbours and only a disagreeing one, the probability that the agent is forced to change its opinion is low. On the contrary, in our model it happens

[2] We refer to the number of iterations, whereas in [21] one unit time consists of N updates so $t_{cons} \sim N$ for $\alpha = 0$ and $t_{cons} \sim \log N$ for $\alpha = 1$. As already mentioned, the two formulations are equivalent.

with probability one. Intuitively, this affects the time to convergence. In the classic model nodes which have many inert connections have low probability of changing opinion whereas, nodes with many active links will change opinion with high probability. This favours the convergence towards a consensus since the fusion rule tends to an overall agreement. In the new model a discordant node is chosen uniformly and it will with probability one change its opinion at every time step.

Table 5.1: Comparison: Iterations needed to consensus.

	Time to Consensus	
N	*direct* voter model	*new* voter model
30	121.4	19813.4
40	155.2	189268.2
50	155.8	7742601.4
60	157.8	25157428.0

The growth of the survival time is so high that convergence cannot be seen if not in trivial cases, for networks with $N < 80$. One realization of the process on a network of eighty nodes takes up to several days of simulation making unfeasible the investigation on such networks. Obtaining data for this new version is far more difficult than for the classic model and to make a comparison between the two we could just run the dynamics on graphs of few nodes. In Table 5.1 the difference in the number of iterations to consensus is shown. As already mentioned the number of nodes is trivial and data have been obtained averaging over 10 realizations but, these numbers give a clear idea of the difference in convergence of the two processes.

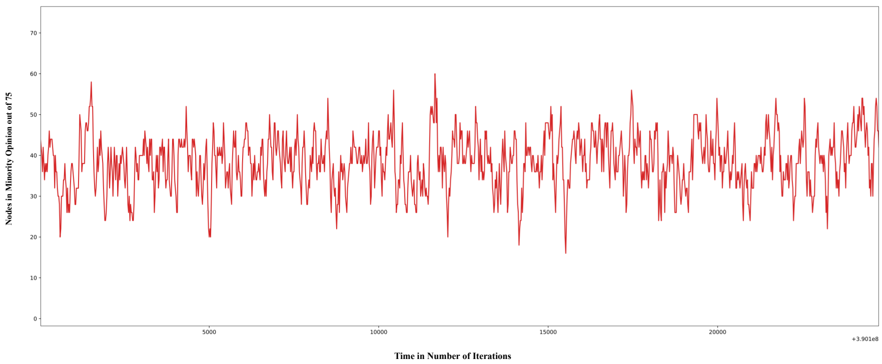


Figure 5.7: Only a portion of the behaviour in time is shown, an entire realization appears to be unreadable due to the extremely high number of data points. This sample has been produced running the modified version of the *direct* voter model on an Erdős-Rényi with $N = 75$ nodes.

As it can be seen from the plot showing the fraction of nodes in the minority state as a function of the number of iterations (Figure 5.7) the process keeps fluctuating around a mean value for an extremely long time. Since the stochastic process has two absorbing states (consensus) it will eventually reach one of them but, the fact that an update can take place also for nodes whose majority of connections are inert, hinders convergence. The model might be unrealistic because it does not take into consideration the bias of the neighborhood of a node towards a certain opinion and just triggers an update whenever there is a disagreement. However, it shows how model's modification which might seem minor can have a drastic effect on the system's behaviour.

Conclusions

In this work the main characteristics of opinion dynamics models have been presented providing a general framework valid for any model. In addition, the importance of complex networks in the field of social modelling has been many times underlined. We have discussed several generative models from the Erdős-Rényi to the Chung-Lu model, highlighting the aspects of real-world networks that each model is able to capture. Efficient generative algorithms have been presented together with the architecture of the `c++` simulator which has been developed. Since the processes we have considered are particularly demanding in terms of computational resources especially when considering large networks, the simulator needs to be thoughtfully designed, in order to be scalable. The main focus of this work has been the study of an adaptive voter model proposed by Durrett et al. in which the opinions of the agents of the network change together with the underlying graph. This phenomenon is referred to as *adaptivity* and leads to a rich dynamical behaviour. We have shown the phase transitions that the system undergoes in the two versions of the model: rewire-to-same and rewire-to-random. It is important to observe that the critical threshold of these phase transitions is strongly influenced by the average degree of the underlying graph. However, surprisingly it appears not to depend on the structure of the initial graph. Running the process on the stochastic block model (SBM) and on the Chung-Lu graph (CL) the results obtained are similar to the ones on the Erdős-Rényi graph. It has to be noted that these initial configurations strongly vary from one another, the Erdős-Rényi model produces graphs whose nodes have the same expected degree. On the contrary, the Chung-Lu model generates networks with a power-law degree distribution in which few nodes (*hubs*) have a very large degree.

In the last chapter we have discussed the behaviour of the time needed to reach consensus. The literature for the *classic* voter model is vast and has been shortly reviewed. While on homogeneous lattices, many interesting results have been formally proven, the investigation of opinion dynamics processes on heterogeneous graphs is still ongoing. Considering the adaptive voter model of Durrett et al. we have shown how the time complexity drastically changes for different values of the rewiring α, from N^2 for small values of the parameter, to $N \log N$ for values approaching 1. To demonstrate how minor changes in the module formulation greatly affect the behaviour of the dynamics, we have presented a variation of the *classic* voter model which does not reach consensus in a reasonable time, the update rule hinders the convergence and just random fluctuations lead the system to one of the absorbing states.

There is yet a conspicuous amount of work to be done in this field in order to get a better grip on the mechanisms behind one of the most important human processes: the opinion-formation. To start with, it would be of interest to carry out a mathematical explanation of the phenomena which have been observed in this work. Moreover, it can be investigated the effect of a different initial configuration on the time needed to reach consensus, since in Chapter 5 we have only considered Erdős-Rényi graphs as initial graphs.

Bibliography

[1] J. R. P. French, “A formal theory of social power.”, *Psychological Review*, vol. 63, no. 3, pp. 181–194, 1956. DOI: 10.1037/h0046123. [Online]. Available: https://doi.org/10.1037/h0046123.

[2] M. H. Degroot, “Reaching a consensus”, *Journal of the American Statistical Association*, vol. 69, no. 345, pp. 118–121, Mar. 1974. DOI: 10.1080/01621459.1974.10480137. [Online]. Available: https://doi.org/10.1080/01621459.1974.10480137.

[3] R. A. Holley and T. M. Liggett, “Ergodic theorems for weakly interacting infinite systems and the voter model”, *The Annals of Probability*, vol. 3, no. 4, pp. 643–663, Aug. 1975. DOI: 10.1214/aop/1176996306. [Online]. Available: https://doi.org/10.1214/aop/1176996306.

[4] K. Sznajd-Weron and J. Sznajd, “Opinion evolution in closed community”, *International Journal of Modern Physics C*, vol. 11, no. 06, pp. 1157–1165, Sep. 2000, ISSN: 1793-6586. DOI: 10.1142/s0129183100000936. [Online]. Available: http://dx.doi.org/10.1142/S0129183100000936.

[5] A. S. Elgazzar, “Application of the sznajd sociophysics model to small-word networks”, *International Journal of Modern Physics C*, vol. 12, no. 10, pp. 1537–1544, Dec. 2001. DOI: 10.1142/s0129183101002875. [Online]. Available: https://doi.org/10.1142/s0129183101002875.

[6] P. L. Krapivsky and S. Redner, “Dynamics of majority rule in two-state interacting spin systems”, *Physical Review Letters*, vol. 90, no. 23, Jun. 2003. DOI: 10.1103/physrevlett.90.238701. [Online]. Available: https://doi.org/10.1103/physrevlett.90.238701.

[7] T. Rogers and T. Gross, “Consensus time and conformity in the adaptive voter model”, *Physical Review E*, vol. 88, no. 3, Sep. 2013. DOI: 10.1103/physreve.88.030102. [Online]. Available: https://doi.org/10.1103/physreve.88.030102.

[8] A. Sirbu, V. Loreto, V. D. P. Servedio, and F. Tria, “Opinion dynamics: Models, extensions and external effects”, in *Understanding Complex Systems*, Springer International Publishing, May 2016, pp. 363–401. DOI: 10.1007/978-3-319-25658-0_17. [Online]. Available: https://doi.org/10.1007/978-3-319-25658-0_17.

[9] R. Durrett, J. P. Gleeson, A. L. Lloyd, P. J. Mucha, F. Shi, D. Sivakoff, J. E. S. Socolar, and C. Varghese, "Graph fission in an evolving voter model", *Proceedings of the National Academy of Sciences*, vol. 109, no. 10, pp. 3682–3687, Feb. 2012. DOI: 10.1073/pnas.1200709109. [Online]. Available: https://doi.org/10.1073/pnas.1200709109.

[10] Y. Dong, M. Zhan, G. Kou, Z. Ding, and H. Liang, "A survey on the fusion process in opinion dynamics", *Information Fusion*, vol. 43, pp. 57–65, Sep. 2018. DOI: 10.1016/j.inffus.2017.11.009. [Online]. Available: https://doi.org/10.1016/j.inffus.2017.11.009.

[11] P. Clifford and A. Sudbury, "A model for spatial conflict", *Biometrika*, vol. 60, no. 3, pp. 581–588, 1973. DOI: 10.1093/biomet/60.3.581. [Online]. Available: https://doi.org/10.1093/biomet/60.3.581.

[12] E. Yildiz, A. Ozdaglar, D. Acemoglu, A. Saberi, and A. Scaglione, "Binary opinion dynamics with stubborn agents", *ACM Transactions on Economics and Computation*, vol. 1, no. 4, pp. 1–30, Dec. 2013. DOI: 10.1145/2538508. [Online]. Available: https://doi.org/10.1145/2538508.

[13] R. Hegselmann and U. Krause, "Opinion Dynamics and Bounded Confidence Models, Analysis and Simulation", *Journal of Artificial Societies and Social Simulation*, vol. 5, no. 3, pp. 1–24, 2002. [Online]. Available: https://ideas.repec.org/a/jas/jasssj/2002-5-2.html.

[14] P. V. Mieghem, *Performance Analysis of Communications Networks and Systems*. Cambridge University Press, 2006. DOI: 10.1017/cbo9780511616488. [Online]. Available: https://doi.org/10.1017/cbo9780511616488.

[15] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks", *Nature*, vol. 393, no. 6684, pp. 440–442, Jun. 1998. DOI: 10.1038/30918. [Online]. Available: https://doi.org/10.1038/30918.

[16] N. E. Friedkin and E. C. Johnsen, "Social influence and opinions", *The Journal of Mathematical Sociology*, vol. 15, no. 3-4, pp. 193–206, Jan. 1990. DOI: 10.1080/0022250x.1990.9990069. [Online]. Available: https://doi.org/10.1080/0022250x.1990.9990069.

[17] ——, "Social positions in influence networks", *Social Networks*, vol. 19, no. 3, pp. 209–222, Aug. 1997. DOI: 10.1016/s0378-8733(96)00298-5. [Online]. Available: https://doi.org/10.1016/s0378-8733(96)00298-5.

[18] L. Festinger, "A theory of social comparison processes", *Human Relations*, vol. 7, no. 2, pp. 117–140, May 1954. DOI: 10.1177/001872675400700202. [Online]. Available: https://doi.org/10.1177/001872675400700202.

[19] G. Deffuant, D. Neau, F. Amblard, and G. Weisbuch, "Mixing beliefs among interacting agents", *Advances in Complex Systems*, vol. 03, no. 01n04, pp. 87–98, Jan. 2000. DOI: 10.1142/s0219525900000078. [Online]. Available: https://doi.org/10.1142/s0219525900000078.

[20] V. Sood, T. Antal, and S. Redner, "Voter models on heterogeneous networks", *Physical Review E*, vol. 77, no. 4, Apr. 2008, ISSN: 1550-2376. DOI: 10.1103/physreve.77.041121. [Online]. Available: http://dx.doi.org/10.1103/PhysRevE.77.041121.

[21] C. Nardini, B. Kozma, and A. Barrat, "Who's talking first? consensus or lack thereof in coevolving opinion formation models", *Physical Review Letters*, vol. 100, no. 15, Apr. 2008. DOI: 10.1103/physrevlett.100.158701. [Online]. Available: https://doi.org/10.1103/physrevlett.100.158701.

[22] D. Kimura and Y. Hayakawa, "Coevolutionary networks with homophily and heterophily", *Physical Review E*, vol. 78, no. 1, Jul. 2008. DOI: 10.1103/physreve.78.016103. [Online]. Available: https://doi.org/10.1103/physreve.78.016103.

[23] P. Holme and M. E. J. Newman, "Nonequilibrium phase transition in the coevolution of networks and opinions", *Physical Review E*, vol. 74, no. 5, Nov. 2006. DOI: 10.1103/physreve.74.056108. [Online]. Available: https://doi.org/10.1103/physreve.74.056108.

[24] E. Yildiz, A. Ozdaglar, D. Acemoglu, A. Saberi, and A. Scaglione, "Binary opinion dynamics with stubborn agents", *ACM Transactions on Economics and Computation*, vol. 1, no. 4, pp. 1–30, Dec. 2013. DOI: 10.1145/2538508. [Online]. Available: https://doi.org/10.1145/2538508.

[25] A. Basak, R. Durrett, and Y. Zhang, *The evolving voter model on thick graphs*, 2015. arXiv: 1512.07871 [math.PR].

[26] D. Fasino, A. Tonetto, and F. Tudisco, *Generating large scale-free networks with the chung-lu random graph model*, 2019. arXiv: 1910.11341 [physics.soc-ph].

[27] M. Alam, M. Khan, A. Vullikanti, and M. Marathe, "An efficient and scalable algorithmic method for generating large-scale random graphs", in *SC '16: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2016, pp. 372–383.

[28] V. Batagelj and U. Brandes, "Efficient generation of large random networks", *Phys. Rev. E*, vol. 71, p. 036 113, 3 Mar. 2005. DOI: 10.1103/PhysRevE.71.036113. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevE.71.036113.

[29] F. Chung and L. Lu, "Connected components in random graphs with given expected degree sequences", *Annals of Combinatorics*, vol. 6, no. 2, pp. 125–145, Nov. 2002. DOI: 10.1007/pl00012580. [Online]. Available: https://doi.org/10.1007/pl00012580.

[30] D. Fasino, A. Tonetto, and F. Tudisco, *Generating large scale-free networks with the chung-lu random graph model*, 2019. arXiv: 1910.11341 [physics.soc-ph].

[31] J. C. Miller and A. Hagberg, "Efficient generation of networks with given expected degrees", in *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2011, pp. 115–126. DOI: 10.1007/978-3-642-21286-4_10. [Online]. Available: https://doi.org/10.1007/978-3-642-21286-4_10.

[32] C. Lee and D. J. Wilkinson, "A review of stochastic block models and extensions for graph clustering", *Applied Network Science*, vol. 4, no. 1, Dec. 2019. DOI: 10.1007/s41109-019-0232-2. [Online]. Available: https://doi.org/10.1007/s41109-019-0232-2.

[33] C. Castellano, S. Fortunato, and V. Loreto, "Statistical physics of social dynamics", *Reviews of Modern Physics*, vol. 81, no. 2, pp. 591–646, May 2009, ISSN: 1539-0756. DOI: 10.1103/revmodphys.81.591. [Online]. Available: http://dx.doi.org/10.1103/RevModPhys.81.591.

[34] E. Ben-Naim, L. Frachebourg, and P. L. Krapivsky, "Coarsening and persistence in the voter model", *Phys. Rev. E*, vol. 53, pp. 3078–3087, 4 Apr. 1996. DOI: 10.1103/PhysRevE.53.3078. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevE.53.3078.

[35] J. T. Cox, "Coalescing random walks and voter model consensus times on the torus in zd", *Ann. Probab.*, vol. 17, no. 4, pp. 1333–1366, Oct. 1989. DOI: 10.1214/aop/1176991158. [Online]. Available: https://doi.org/10.1214/aop/1176991158.

[36] K. Suchecki, V. M. Eguíluz, and M. S. Miguel, "Conservation laws for the voter model in complex networks", *Europhysics Letters (EPL)*, vol. 69, no. 2, pp. 228–234, Jan. 2005, ISSN: 1286-4854. DOI: 10.1209/epl/i2004-10329-8. [Online]. Available: http://dx.doi.org/10.1209/epl/i2004-10329-8.

[37] C. Castellano, D. Vilone, and A. Vespignani, "Incomplete ordering of the voter model on small-world networks", *Europhysics Letters (EPL)*, vol. 63, no. 1, pp. 153–158, Jul. 2003. DOI: 10.1209/epl/i2003-00490-0. [Online]. Available: https://doi.org/10.1209%2Fepl%2Fi2003-00490-0.

[38] K. Suchecki, V. M. Eguíluz, and M. San Miguel, "Voter model dynamics in complex networks: Role of dimensionality, disorder, and degree distribution", *Physical Review E*, vol. 72, no. 3, Sep. 2005, ISSN: 1550-2376. DOI: 10.1103/physreve.72.036132. [Online]. Available: http://dx.doi.org/10.1103/PhysRevE.72.036132.