University of Bucharest

Virtute et Sapientia

Facultatea de Fizică Universitatea din București

FACULTY OF PHYSICS

Simulation of Potts Model on a Dynamically Rewired Network

Bachelor's Thesis

Author

Luca Mircea MIHĂILESCU

Scientific coordinator

Conf. Univ. Dr. Alexandru NICOLIN

Bucharest, 2020

Acknowledgements

I would like to acknowledge and thank the following important people who have supported me, not only during the course of this project, but throughout my Bachelor's degree.

Firstly, I would like to express my gratitude to my supervisor, Assistant Professor Dr. Alexandru Nicolin, for his support, guidance in the field of numerical methods and simulations, and insight throughout this research project.

I would also like to thank my colleague, Sebastian Micluță-Câmpeanu. Without his expertise in the Julia programming language, I would not been able to run my simulations at the scale I did.

And finally, I would like to thank all my close friends and family. You have all helped me to focus on what has been a hugely rewarding and enriching process.

Table of Contents

Introduction

The understanding of the laws which govern the behaviour of social masses is one of the outstanding challenges of modern research. It is now more important than ever as our democratic societies are threatened by the rise of *en masse* data mining and nontransparent social media algorithms[1]. Fortunately, physics can lend a hand in building such an understanding. The idea of a physical modeling of social phenomena is not at all a new one. In an 1825 essay, french philosopher Auguste Comte defines social physics as:

> ”that science which occupies itself with social phenomena, considered in the same light as astronomical, physical, chemical, and physiological phenomena, that is to say as being subject to natural and invariable Laws the discovery of which is the special object of its researches.” [2, 3]

Ten years later, Belgian statistician Adolphe Quetelet publishes his *Essay on Social Physics*[4], which proposes characterising social statistics using the concept of the ’average man’ which would be built on measured variables that follow a normal distribution[5]. After this essay, Comte would go on to refer to his new field as sociology, out of fear of being regarded as a follower.

The developments in the fields of social statistics were well known to Maxwell and Boltzmann and played a role in their embracing a statistical description of gases in favour of deriving the macroscopic laws of gases from the individual motions of particles, thus laying the foundations of modern statistical physics[6]. In a 1873 lecture to the British Association, Maxwell argues that physicists have started to employ the methods already used at the time by social statisticians[7]. Boltzmann, in the introduction to a scientific paper published by the Vienna Academy of Science a few years earlier, similarly states that the connection between the theory of heat and ’the principle of living forces’ has been ”known for a long time already”[8].

In the recent years, with the extended accessibility to computational resources and large databases (mostly thanks to the Internet), the field of social dynamics has made the transition from philosophical thought experiments to concrete research efforts worldwide[9]. However, to apply the tools and concepts of thermodynamics to the study of society, one needs to understand their meaning in a social context. One such interpretation of thermodynamics concepts into sociology has been produced by J. Mimkes in 1995 [10]. According to his study, a binary multicultural society can be understood using the model of regular solutions, which is applied to metal alloys. Members of two communities can manifest sympathy to members outside the community (attractive interaction), be indifferent to them (ideal solution), or manifest antipathy to them (repulsive interaction). In this interpretation Gibbs free energy G describes the general happiness of the society, and temperature T can be understood as tolerance, which can make society more united against the differences of the two communities. Table 1.1 shows all the conclusions drawn regarding the equivalence between thermodynamics and social science.

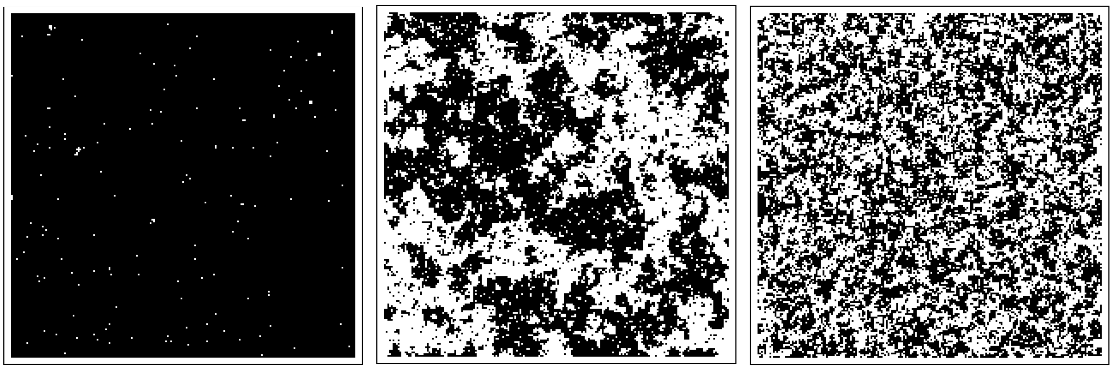
Table 1.1: *Equivalence of thermodynamics terms to social science according to Mimkes' model of regular mixtures*

Abbreviation	Natural Science	Social scienceă
A-B	Alloys	Societies
x	atomic percentage	size of minority (%)
	Functions	Feelings
G	free enthalpy	general happiness
T	temperature	tolerance
E_{AA}	cohesive energy	tradition, heritage
$E_{AB} > 0$	cohesive energy	curiosity, love
$E_{AB} < 0$	repelling energy	distrust, hate
$E = 0$	no cohesion	apathy
$\epsilon > 0$	attractive interaction	sympathy
$\epsilon = 0$	ideal solution	indifference
$\epsilon < 0$	repelling interaction	antipathy
	State of alloys	State of Society
	disorder, solubility	integration
	solubility limit	segregation
	phase diagram	intermarriage diagram

1.1 The Ising paradigm

Whether we're focusing on opinions, social status, cultural and linguistic features, or human kinematics, models can be devised in terms of small sets of variables. Of course, they would be oversimplifications, but qualitative properties of large scale phenomena do not necessarily depend on the microscopic details of the process. As such, simplified models can offer valuable information about macroscopic features such as symmetries, dimensionality, conservation laws etc. One of the most relevant models in physics to this kind of analysis is the Ising model for ferromagnets[11]. Beyond its physical significance, the Ising ferromagnet can also serve as a simple model for opinion dynamics: spins can be seen agents under the influence of the majority of their interacting partners.

Let us consider a collection of N points (i.e. agents) with a spin (i.e. opinion) $s_i = \pm 1$. For any two neighbouring points i, j there is an interaction J_{ij}. Energetically, this interaction determines each spin to be aligned with its nearest neighbours. When no external magnetic field is present, the total energy of the system is equal to the Hamiltonian function



(a) $T < T_c$ (b) $T \sim T_c$ (c) $T > T_c$

Figure 1.1: *Spatial configurations in the Ising model. Black squares represent spins with $\sigma = +1$ and white one correspond to $\sigma = -1$. (Reprinted from "Behavior of Early Warnings near the Critical Temperature in the Two-Dimensional Ising Model," by Morales I.O., Landa E., Angeles C.C., Toledo J.C., Rivera A.L., et al., 2015, PLOS ONE, 10(6), doi:10.1371/journal.pone.0130751. Copyright 2015 by Morales et al. Distributed under terms of the Creative Commons Attribution License.)*

$$H = -\frac{1}{2}\sum_{(i,j)} J_{ij} s_i s_j \tag{1.1}$$

where the sum runs on the pairs of neighbours. The most common implementation of the Ising dynamics is the Metropolis algorithm[12]. In it, each step of the simulation a spin is flipped with a probability $\exp(-\Delta E/k_B T)$, where ΔE is the change in energy, k_B is the Boltzmann constant, and T is the temperature. The interactions driven by (1.1) should lead to a completely homogeneous state: either all spins are positive, or all are negative. However, this holds only for small temperatures. At temperatures above a critical temperature T_c, thermal noise injected fluctuations destroy order. By definition, the average magnetization is

$$M = \frac{1}{N}\sum_{j}\langle s_j \rangle \tag{1.2}$$

where the brackets denote the average over different iterations. For $T < T_c$ the magnetization will be $M(T) > 0$, while for $T > T_c$, $M(T) = 0$. Worth mentioning is also the Potts model[13], which will be relevant later on for the model I decided to employ. The difference in its case is that spins can take one out of q values. Identical neighbouring spins are energetically favored. The Potts has found such uses as the simulation of sorting in a mixture of biological cells[14], or computer vision and image restoration[15]. The Ising model corresponds to the Potts special case $q = 2$.

Yet another variation to the Ising model can be found in Axelrod's model of dissemination of culture[16]. From the point of view of statistical physics it is a vectorial generalization of the Ising-like models (here culture refers to a vector of variables denoting a set of inextricable 'cultural' characteristics). The model consists of individuals located on a network (or lattice) endowed with a vector of F integer variables $(\sigma_1, ..., \sigma_F)$. Each variable, or 'cultural features', can assume values $\sigma_f = 0, 1, ..., q$. These cultural features are supposed to model different “beliefs, attitudes, and behavior”. Each step, an individual i together with a neighbor j are selected and the similarity between them is calculated:

$$\omega_{i,j} = \frac{1}{F} \sum_{f=1}^{F} \delta_{\sigma_f(i), \sigma_f(j)} \tag{1.3}$$

here $\delta_{i,j}$ being Kronecker's delta. Then, with probability $\omega_{i,j}$ one of the features for which traits are different is set equal to the neighbor's $(\sigma_f(i) = \sigma_f(j))$. The phenomenology these dynamics determine is not trivial, however, and predicts the emergence of polarization despite the tendency of interacting people to become more alike.

The Ising model has been applied to describe business confidence, segregation, and language change[17]. In the last two cases[18, 19], the authors were not aware of the Ising model, and designed more complex simulations that were less flexible. In the case of business confidence[20] good news or bad news can lead to a uniform optimist or pessimist overlook in the population, if the news follow in too quick succession (i.e. the field oscillates too much), people will start ignoring them and adopting random opinions [21]. In the study of segregation, temperature T emerges as a measure of tolerance, with individual agents possibly having their own T which might change over time. In the case of language change, it seems that the rate of change for certain characteristics decays as the population gets larger. If the agents only exchange characteristics with their neighbours, this influence is weak.

1.2 Scale-free distributions

A very important role in the theory of critical phase transitions (such as the one reviewed earlier in the case of the Ising model) is played by critical point exponents[22]. To arrive to critical exponents, we need to define, for convenience, a measure of the deviation in temperature from the critical temperature T_c:

$$\tau = \frac{T - T_c}{T_c} \tag{1.4}$$

The critical exponent associated with a function will then be:

$$\lambda = \lim_{\tau \to 0} \frac{\ln |F(\tau)|}{\ln |\tau|} \tag{1.5}$$

More usually, we will encounter the following, equivalent, relation:

$$F(\tau) \sim |\tau|^{-\lambda} \tag{1.6}$$

Here, the $\sim$ sign is used instead of the two sides being equal because (1.6) it represents the asymptotic behaviour of the function $F(\tau)$ as $\tau \to 0$. In magnetic systems, critical exponents are for several functions, some of the most common of which are listed in table 1.2.

Table 1.2: *Commonly used critical exponents for a magnetic and fluid systems*

Magnetic system	
Zero-field specific heat	$C_H \sim \|\tau\|^{-\alpha}$
Zero-field magnetization	$M \sim (-\tau)^{\beta}$
Zero-field isothermal susceptibility	$\chi_T \sim \|\tau\|^{-\gamma}$
Critical isotherm ($t = 0$)	$H \sim \|M\|^{\delta} \operatorname{sgn}(M)$
Correlation length	$\xi \sim \|\tau\|^{-\nu}$
Pair correlation function at T_C	$G(\vec{r}) \sim 1/r^{d-2+\eta}$
Fluid system	
Specific heat at constant volume V_C	$T \sim \|t\|^{-\alpha}$
Liquid-gas density difference	$(\rho_l - \rho_g) \sim (-t)^{\beta}$
Isothermal compressibility	$k_T \sim \|t\|^{-\gamma}$
Critical isotherm ($t = 0$)	$P - P_c \sim \|\rho_l - \rho_g\|^{\delta} \operatorname{sgn}(\rho_l - \rho_g)$
Correlation length	$\xi \sim \|\tau\|^{-\nu}$
Pair correlation function at T_C	$G(\vec{r}) \sim 1/r^{d-2+\eta}$

We call a distribution scale-free when it follows a power law like the one at (1.6). Essentially, when measuring something, if the distribution of results follows a power-law it means that the measured phenomenon exhibits features at all scales. This occurrence is not at all limited to critical phase transitions, however[23]. For example, the distribution of earthquake magnitudes seems to obey a $k^{-\gamma}$ function with $\gamma \approx 3.04$, number of hits on web sites, $\gamma \approx 2.40$, intensity of wars, $\gamma \approx 1.80$, and intensity of solar flares, $\gamma \approx 1.83$ [24]. Studies have also been done on the rank-frequency distribution of words in various languages, of which I will just mention Romanian, with $\gamma \approx 1$ [25], and English, with $\gamma \approx 2.20$ [24].

Scale-free distributions do not need to follow strictly a $k^{-\gamma}$ form. Take, for instance, figure 1.2, which comes from a study calculating the distribution of income in the United States[26]. The distribution in this case is a double power law. More complex distributions exist, but they are beyond the scope of this thesis.

1.3 Scale-free networks

Often, natural and man-made systems (e.g. the Internet, citation networks, social networks) have the tendency to exhibit a structure closely resembling scale-free networks[27].

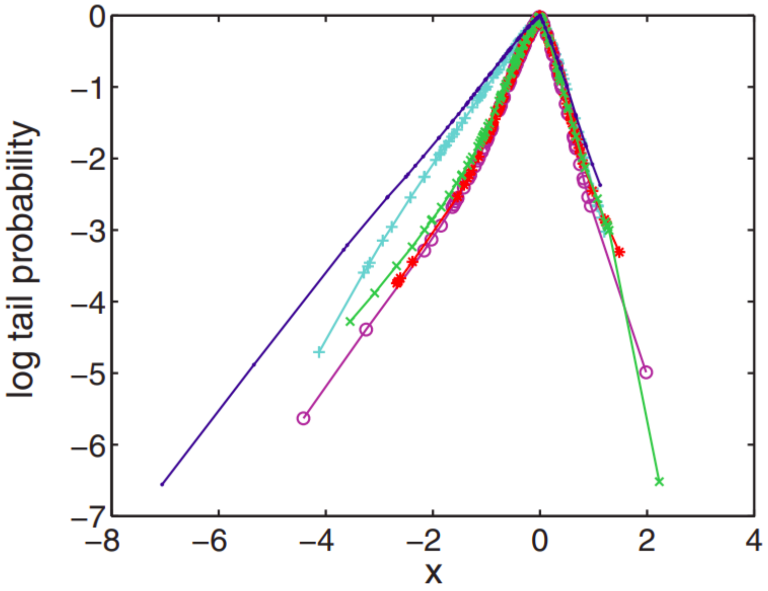


Figure 1.2: *Laplace plot by age group. Cyan plus (+): under 30, magenta circle (o): 30s, red asterisk (*): 40s, green cross (x): 50s, blue dot (·): over 60. Data from the U.S. Census Bureau Current Population Survey 2000-2009. Reprinted from Journal of Economic Behavior & Organization, 84(1), Alexis Akira Toda, "The double power law in income distribution: Explanations and evidence," pages 364-381, Copyright 2012, with permission from Elsevier.*

These are networks with a scale-free degree distribution (i.e. the probability that a randomly chosen node will have k connections). This means that the fraction $P(k)$ of nodes of degree k has the following form:

$$P(k) \sim k^{-\gamma} \tag{1.7}$$

In a famous study about the World Wide Web, University of Notre-Dame researchers mapped all incoming and outgoing links in the university's *nd.edu* domain[28]. Sure enough, the distribution they found was a power-law like the one at (1.7), meaning it was a scale-free network. The resulting exponent for outgoing links was then checked against the ones obtained by independently mapping *whitehouse.gov*, *yahoo.com*, and *snu.ac.kr*. All the three domains exhibited a distribution with the same exponent as with *nd.edu*, $\gamma \approx 2.45$.

In an attempt to explain the occurrence of scale-free networks in real life, Albert-László Barabási and Réka Albert have elaborated an algorithm for generating random scale-free networks (see figure 1.3) using a preferential attachment mechanism[29]. In the Barabási-Albert model, the network starts with m_0 initial connected nodes. Then, new nodes are added to the network one by one. Each new node is then attached to $m \leq m_0$ of the already existing nodes with the following probability:

$$p_i = \frac{k_i}{\sum_j k_j} \tag{1.8}$$

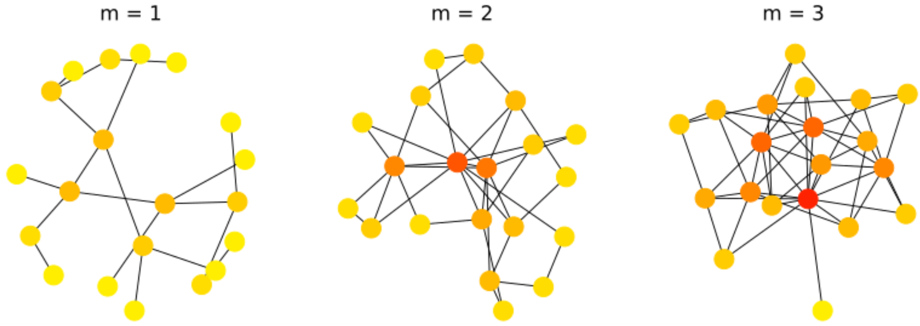


Figure 1.3: *Display of three graphs generated with the Barabasi-Albert (BA) model. Each has 20 nodes and a parameter of attachment m as specified. The color of each node is dependant upon its degree (same scale for each graph). (Wikipedia user HeMath / Creative Commons Attribution-Share Alike 4.0 International license (`https://commons.wikimedia.org/wiki/File:Barabasi_albert_graph.svg`))*

where p_i is the probability that the new node is attached to node i, k_i is the degree of node i, and the sum is calculated over all pre-existing nodes j.

An evolving Barabási-Albert network can be mapped to a Bose gas, with nodes corresponding to energy levels, and links to particles[30]. For each new node, $2m$ particles are added: m particles on the energy level corresponding to the node's fitness, and m particles distributed to the other energy levels, corresponding to the outgoing links. There are 3 possible behaviours:

1. Scale-free phase: occurs when all nodes have the fitness. The fraction of links of the oldest node decay to zero in the thermodynamic limit.
2. Fit-get-rich: occurs when nodes have different fitnesses and $I(\beta, \mu) = 1$ has a solution. Eventually, the system evolves to a configuration of a few very connected nodes along with many less connected ones.
3. Bose-Einstein condensation: occurs under a critical temperature where $I(\beta, \mu) = 1$ has no solution. Under this circumstance a winner-takes-all scenario occurs (the biggest hub also maintains a finite share of the links throughout the expansion of the system).

1.4 Social networks

At the most basic level, a social network is a structure made up of a set of social actors (be it individuals or organizations), links between pairs of people (dyadic ties), and other social interactions[31, 32]. Besides dyads, which are links between two people and simplest possible feature in a social network, we can also encounter repeating structures made up of three or more people. All these are called network motifs: recurrent and statistically

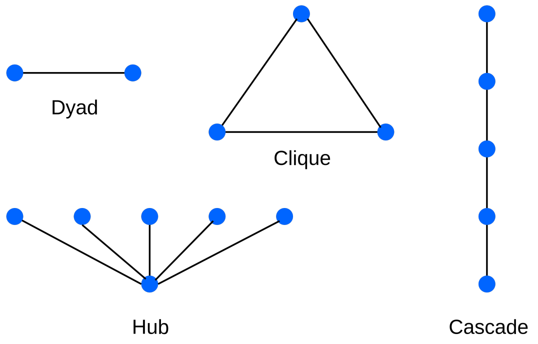


Figure 1.4: *Types of network motifs.*

significant patterns in a network.

Heider's Balance Theory[33] is one of the most famous theories concerned with the analysis of triads (i.e. structures emerging between tree agents). According to it, balance state over a dyad occurs if the two like each other or dislike each other. Meanwhile, should the two have different sentiments regarding each other (positive and negative) the dyad is in imbalance. In the case of triads, balance state between the three can be found if the algebraic multiplication of signs in the triad relations has a positive sign. As it is immediately apparent, this theory holds many similarities with the aforementioned Ising dynamics, in which the energetic tendency is towards what Heider would call balance.

Another aspect that has been studied in social networks is the degree of separation, or average path length[34]. Arguably the most famous experiment in this area is Milgram's small-world experiment[35]. Long before the appearance of internet social networks, this experiment aimed to find what is the average path length between any two persons by making use of the postal system. The procedure was as follows:

1. Individuals in the U.S. city of Boston, Massachusetts, were chosen to be end points of the experiment.

2. Information packets containing the instructions for participating in the experiment were initially sent to randomly selected individuals in Omaha or Wichita.

3. Recipients were asked to sign the roster included in the packet and resend the packet to the target person if they knew the target on a first name basis. If not, they were asked to send it to the person they deemed most likely to be able to reach them in this manner.

4. Upon arrival, the target would inspect the roster and count how many times the package had been forwarded.

Using this methodology, Milgram was able to calculate that the median number of intermediate persons was only 5.5 [36].

1.5 Justification for a two-layered model

All the social simulation models reviewed thus far concerned themselves with reviewing the evolution of one observable in the system, be it opinion, cultural vector, or connectivity. Some of them equip the agents with a certain ’fitness’ value, which determines the probability the agent will have influence on another agent or not. But this too is arbitrary. In real life scenarios such as elections[37], there is one one rapidly evolving question of opinion (e.g. candidate to vote for) while the structure of the social network is given by other, more long-term, features of the individuals (e.g. political values). In other words, one will be more likely to copy the opinion of people sharing the same values as oneself. This effect has been especially observed in the case of social media, where the followed-follower dynamic makes it easier to select the people sharing similar values[38].

In terms of a simulation, this considerations will translate into a two-layered model, with two different dynamics: one layer is concerned with rewiring the network so that agents with similar values connect with each other, and another layer for the evolution of the opinion of interest on said network.

Model Setup

In this thesis, I model opinion dynamics by using a Potts-like agent-based model, and the evolution of the network structure is performed by using a slightly modified version of Axelrod's model of dissemination of culture. This chapter provides the reader with a detailed description of the proposed model.

2.1 Agent characteristics

A directional network consisting of N agents is used. Each agent is represented by a vertex in the network, accompanied by a set of three features. These features are:

1. **Vote:** a variable in the range of $v = 0..n$, representing the agent's opinion. *Votes* can be understood literally, as voting intention in an election, or, more generally, as an opinion subject to quick change in a social network.
2. **Cultural vector:** a vector $\sigma = (\sigma_1, ..., \sigma_F)$, with $\sigma_f = 0, 1, ..., q$, representing an immutable set of cultural characteristics. Here culture is understood as, in Axelrod's words, "*the set of individual attributes that are subject to social influence*".
3. **Energy:** defined as $\epsilon_i = -J \sum_j \delta v_i, v_j$, where $j \in$ inneighbors, and δ is the Kronecker delta symbol. Note that this definition is akin to the Potts model Hamiltonian, with the difference that in this case spins are replaced by *votes*.

2.2 Network rewiring

Each step, an agent is selected for which, with a certain probability, an inneighbour will be removed and another added. Here, I referred to the transition probability defined by Axelrod for his model of dissemination of culture:

$$\omega_{i,j} = \frac{1}{F} \sum_{f=1}^{F} \delta_{\sigma_f(i),\sigma_f(j)} \tag{2.1}$$

However, this probability only grows when to cultural characteristics are *identical.* In reality, beliefs are usually on a spectrum of intensity. For instance, a person with a belief

$\sigma_f = 3$ will find themselves more likely to interact with another with $\sigma_f = 4$ rather than one with $\sigma_f = 9$. Taking this into consideration, a new function η can be devised:

$$\eta_{i,j} = 1 - \frac{1}{F} \cdot \frac{1}{q} \sum_{f=1}^{F} |\sigma_f(i) - \sigma_f(j)| \tag{2.2}$$

Using this revised probability, this stage in the time-step is defined by the following activities:

- One agent i is selected at random.
- Another agent j that is not an inneighbour is selected randomly.
- With a probability $\eta_{i,j}$ an edge from j to i is added, and i's energy ϵ_i is reevaluated.
- Now an agent k is selected randomly from i's inneigbours.
- With a probability $1 - \eta_{i,j}$ the edge from k to i is removed, and i's energy ϵ_i is reevaluated.

It is immediately apparent that this rewiring procedure will lead to similar agents becoming more connected, and eventually forming hubs. This behaviour mirrors the echo-chamber effect observed in social media.

2.3 Opinion dynamics

Having rewired the network the selected agent's connections, it will now reconsider its *vote*. This happens by attributing a new random vote to the agent, and reevaluating agent energy with the new vote. If the energy is lower, i.e. the agent's opinion is more in line with his influencers', then the new vote is kept. Otherwise, it will keep its new opinion with a certain probability, which is dependent on temperature (tolerance) T and difference in energy $\Delta\epsilon = \epsilon_{new} - \epsilon_{old}$:

$$p = \exp\left(-\frac{\Delta\epsilon}{T}\right) \tag{2.3}$$

2.4 Overall execution procedure

Putting the network rewiring part and the opinion dynamics parts together, the algorithm will go through the following procedure:

Step 1: Generating initial population. A data vector containing N data structures is created, where each data structure contains the features described at section 2.1, with random votes and cultural vectors.

Step 2: Network initialization. A Barabási–Albert graph is initialized, growing by adding new vertices to N_0 initial vertices. Each new vertex is attached to k different vertices already present in the system by preferential attachment.

Step 3: Compute initial energy and vote distribution. Compute energy for each agent, count vote distribution, then store the data in the log.

Step 4: Select random agent.

Step 5: Network rewiring. Perform the network rewiring procedure on the selected agent, recalculating ϵ and E afterwards.

Step 6: Opinion dynamics. Perform the opinion dynamics procedure on the selected agent, recalculating ϵ and E afterwards.

Step 7: Advance to next step. Advance the time step and go back to Step 4 until desired number of steps is reached.

Step 8: Data export. The generated data containing the E time series, vote distribution over time and the final network form is exported.

Results

To examine the general characteristics exhibited by this model, I ran a series of simulations, varying, one at a time, population, temperature, initial network, and the way probability $\eta_{i,j}$ is calculated. In what follows I will present each variation together with its result.

The main simulation run was done over a population of 1000 agents initialized in a Barabási–Albert network, for a number of steps $t = 2 \cdot 10^7$, at temperature $T = 1$. The full specifications can be seen in table 3.1.

Table 3.1: *Specifications for main simulation run*

Initial network	Barabási–Albert network
Opinion dynamics	Potts model
Rewiring probability	$\eta_{i,j} = 1 - \frac{1}{F \cdot q} \sum_{f=1}^{F} \lvert \sigma_f(i) - \sigma_f(j) \rvert$
Run time	$t = 2 \cdot 10^7$
Temperature	$T = 1$
Coupling constant	$J = 1$
Population	$N = 1000$

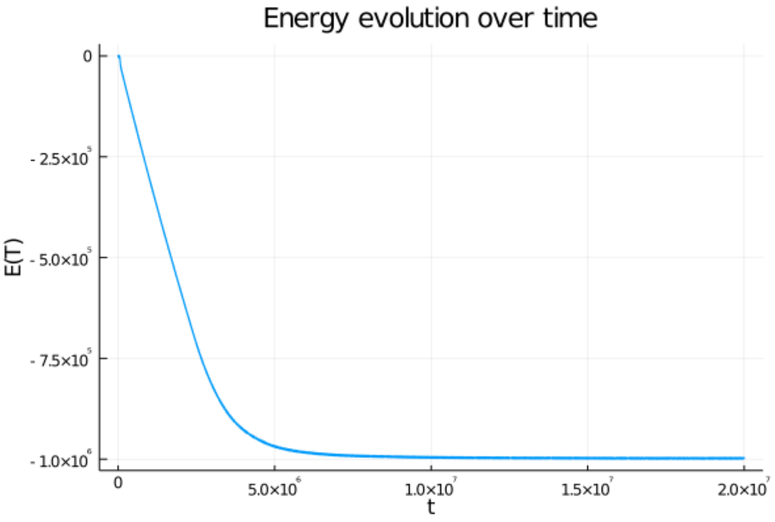


Figure 3.1: *Total system energy evolution over time for a simulation of final model ran for* $t = 2 \cdot 10^7$ *over a population of 1000*

Figure 3.1 shows the evolution of the total system energy for this simulation. Energetically, the system reaches the ground state at $t \approx 5 \cdot 10^6$. However, consensus is reached much earlier ($t \approx 10^5$), as figure 3.2 reveals. This indicates that the network rewiring dynamics bring their own contribution to the total energy. The issue of different rewiring frequencies will be analysed in section 3.2. Refer to figure B.2 for all the plots for this simulation.

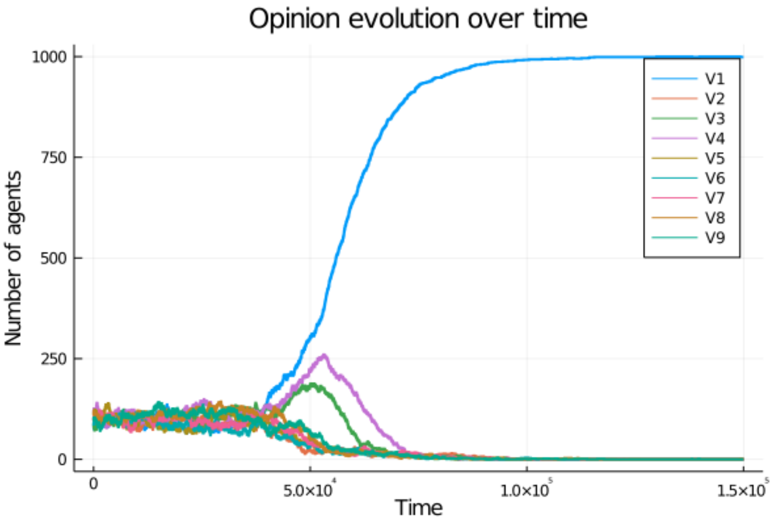


Figure 3.2: *Opinion partition over time,* $t \in \{1..1.5 \cdot 10^5\}$

3.1 Fixed rewiring probability

When the rewiring probability η is fixed, consensus is reached more slowly, even if the population itself is smaller, as revealed by figure 3.3, where consensus wasn't yet reached by the end of the simulation at $t = 1.8 \cdot 10^6$.

Table 3.2: *Specifications for fixed rewiring probability simulation run*

Initial network	Barabási–Albert network
Opinion dynamics	Potts model
Rewiring probability	$\eta_{i,j} = 0.5$
Run time	$t = 1.8 \cdot 10^6$
Temperature	$T = 1$
Coupling constant	$J = 1$
Population	$N = 100$

A possible explanation for the slower opinion dynamics is that the randomness of the rewiring caused by the fixed η makes it harder for the agents to have the same neighbors for too long, thus adding more noise to the dynamics.

Refer to figure B.3 for all the plots for this simulation.

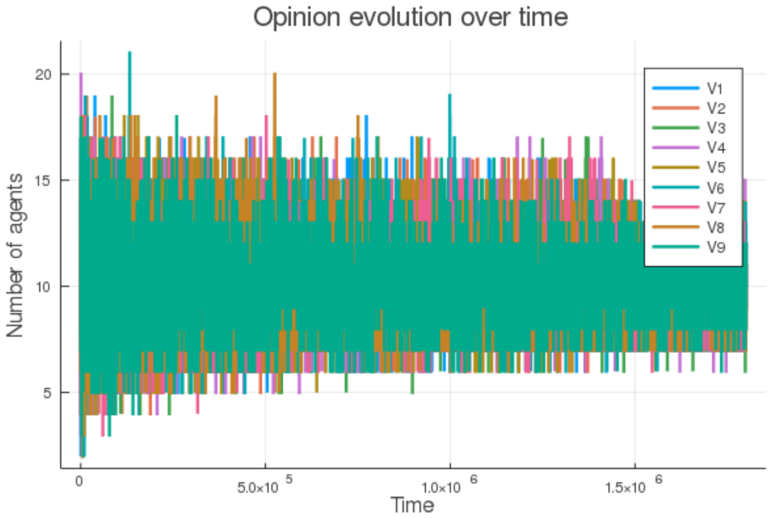


Figure 3.3: *Opinion partition over time for simulation ran for* $t = 1.8 \cdot 10^6$ *over a population of 100*

3.1.1 Majority rule

A variant of the fixed rewiring probability simulation is the majority rule simulation. In this scenario, the opinion dynamics step behaves as follows:

1. An agent is selected at random.
2. The opinions of its inneighbors are counted.
3. The agent copies the opinion held by the majority of its inneighbors.

Table 3.3: *Specifications for majority rule run*

Initial network	Barabási–Albert network
Opinion dynamics	Majority rule
Rewiring probability	$\eta_{i,j} = 0.5$
Run time	$t = 10^6$
Temperature	$T = 1$
Coupling constant	$J = 1$
Population	$N = 100$

Unsurprisingly, in this case consensus is reached almost immediately, despite the noise generated by network rewiring (see figure 3.4). This opinion dynamics rule is even stronger than a Potts rule with $T = 0$, since in that case there are still chances that the new opinion will cause an increase in system energy, and as such it is dropped.

Refer to figure B.4 for all the plots for this simulation.

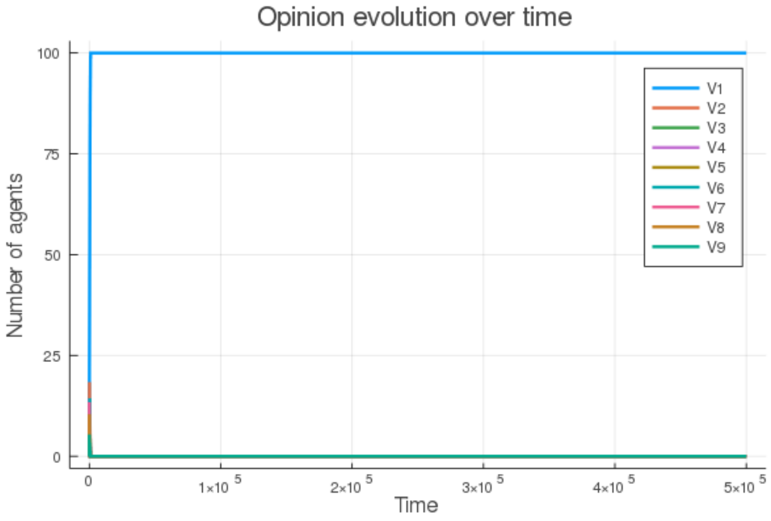


Figure 3.4: *Opinion partition over time for majority rule simulation ran for $t = 10^6$ over a population of 100*

3.1.2 Random initial network

Another tested variant to the fixed rewiring probability simulation is one where the network is initialized using a Erdős–Rényi random network[39].

Table 3.4: *Specifications for Erdős–Rényi initial network run*

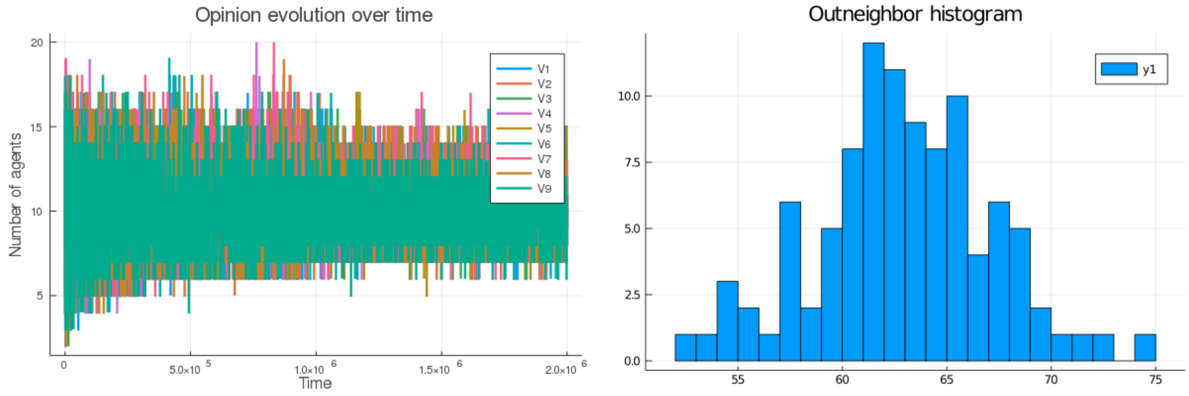
Initial network	Erdős–Rényi network
Opinion dynamics	Potts model
Rewiring probability	$\eta_{i,j} = 0.5$
Run time	$t = 2 \cdot 10^6$
Temperature	$T = 1$
Coupling constant	$J = 1$
Population	$N = 100$

In this case, the results (figure 3.5a) are rather similar to the results obtained from using a Barabási–Albert. This confirms that the network rewiring dynamics are strong enough so as to be independent of the topology of the initial network.

Refer to figure B.5 for all the plots for this simulation.

3.2 Varying rewiring frequency

Keeping the same fixed rewiring probability $\eta = 0.5$, four simulations were experimenting with different rewiring frequencies. Instead of running the rewiring procedure every step of the simulation, the procedure was run at every 10, 50, 100, and 1000 steps, respectively (see table 3.5).



(a) *Opinion partition over time*

(b) *Outneighbor histogram*

Figure 3.5: *Opinion partition and outneighbor histogram for random initial network simulation ran for $t = 2 \cdot 10^6$ over a population of 100*

Table 3.5: *Specifications for different rewiring frequencies simulation runs*

Initial network	Barabási–Albert network
Opinion dynamics	Potts model
Rewiring probability	$\eta_{i,j} = 0.5$
Run time	$t = 1.4 \cdot 10^5, 2 \cdot 10^6, 4.2 \cdot 10^6, 2 \cdot 10^6, 6.5 \cdot 10^6$
Temperature	$T = 1$
Coupling constant	$J = 1$
Population	$N = 100$
Rewiring frequency	$1/10, 1/50, 1/100, 1/1000, 0$

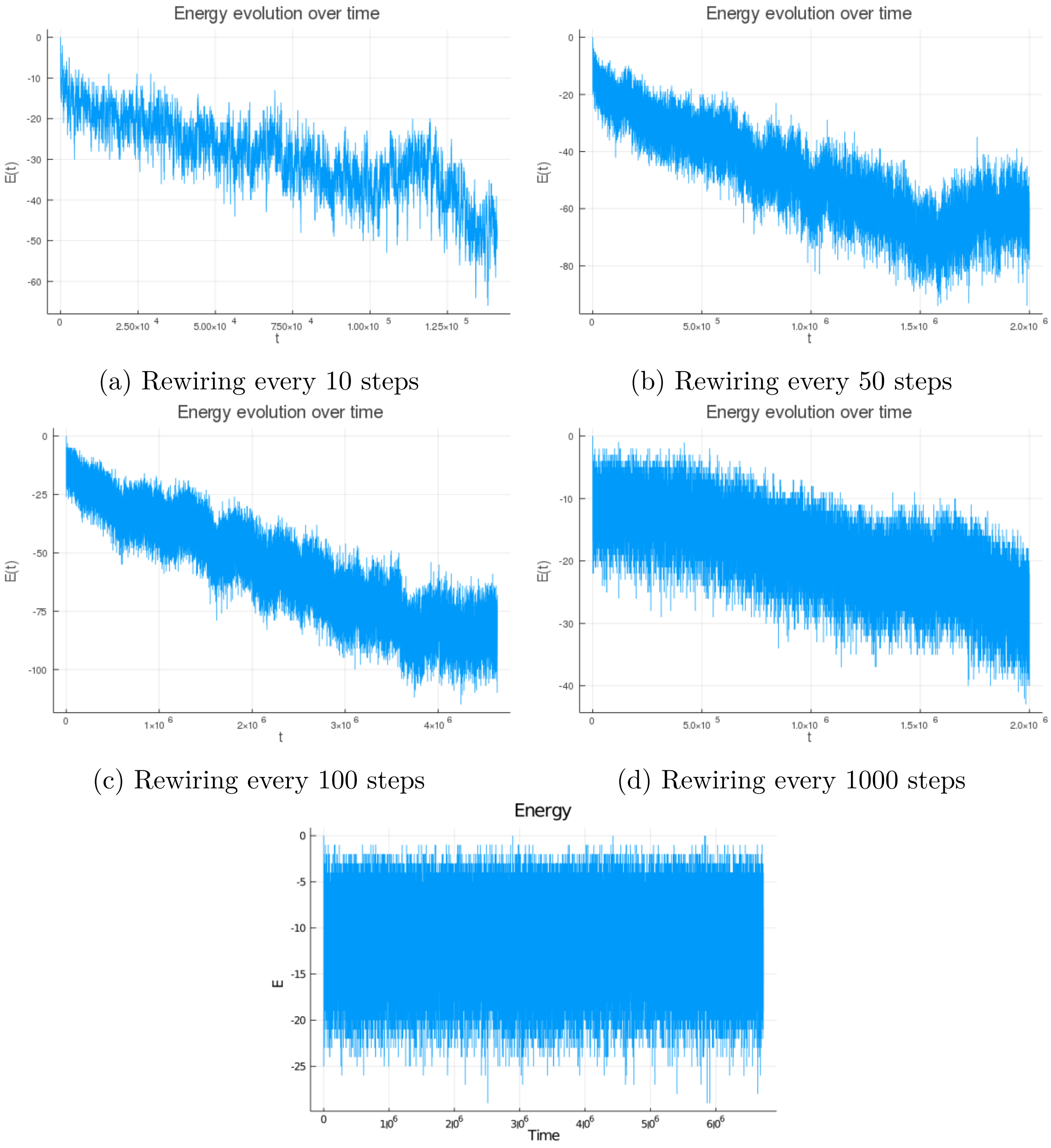
Analysing the results (figure 3.6), it appears, however, that even with a fixed η, the network rewiring is involved in reaching the energy ground state. The lower the rewiring frequency is, the higher the noise and the higher the minimal system energy.

3.3 Varying temperature

Similar to the previous the section, simulations were also done with varying temperatures T. Again, η was fixed and the population was 100 (see table 3.6).

Figure 3.7 shows the energy evolution and the opinion partition over time for a system at $T = 1$. Figures 3.8 and 3.9 show the same information for simulations ran at $T = 10$, and $T = 100$, respectively. Additional plots, such as energy distribution plots, can be found in the appendix figures B.6, B.7, and B.8.

Here, three trends are apparent. First, higher temperature is correlated with higher noise. This should come as no surprise, especially when considering the expression of opinion switch probability p (2.3). Indeed, this mirrors the classic Ising behaviour presented in section 1.1. It is still an important confirmation of the fact temperature behaves as expected in this model, and thus can be easily related to tolerance.



(a) Rewiring every 10 steps

(b) Rewiring every 50 steps

(c) Rewiring every 100 steps

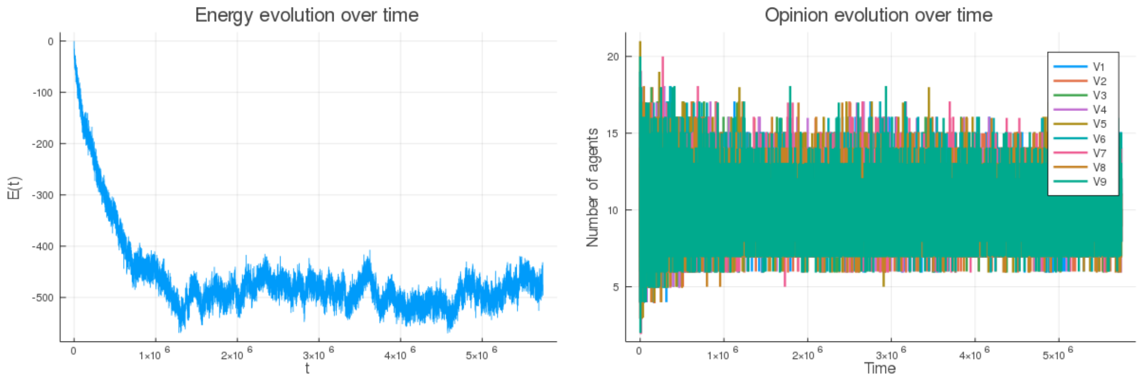
(d) Rewiring every 1000 steps

(e) No rewiring

Figure 3.6: *Simulations ran for different rewiring frequencies over a population of 100*

Table 3.6: *Specifications for different temperatures simulation runs*

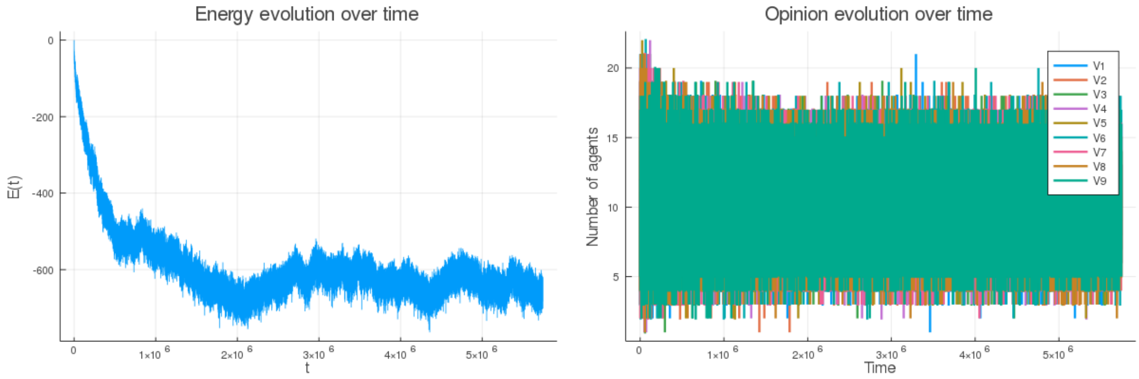
Initial network	Barabási–Albert network
Opinion dynamics	Potts model
Rewiring probability	$\eta_{i,j} = 0.5$
Run time	$t = 6 \cdot 10^6$
Temperature	$T = 1, 10, 100$
Coupling constant	$J = 1$
Population	$N = 100$



(a) Total system energy evolution over time (b) Opinion partition over time

Figure 3.7: *Simulation with fixed rewiring probability, ran for* $t = 6 \cdot 10^6$ *at* $T = 1$, *over a population of 100*

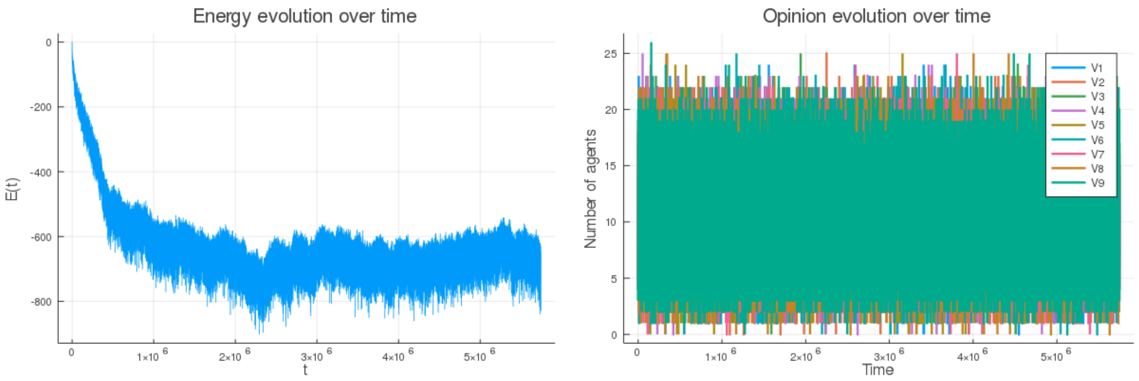
Secondly, the evolution of opinions over time exhibits a chaotic behaviour for $T = 10$ and $T = 100$, while for $T = 1$ it seems like it may be convergent, though a longer run would have been necessary to confirm this beyond all doubts. However, it can still be observed that for $T = 1$ the share of opinions in the network is more evenly divided, hence the smaller values in the opinion partition plot (figure 3.7).



(a) Total system energy evolution over time (b) Opinion partition over time

Figure 3.8: *Simulation with fixed rewiring probability, ran for* $t = 6 \cdot 10^6$ *at* $T = 10$, *over a population of 100*

Thirdly, and most interestingly, the higher the temperature, the lower the system energy ground state seems to be. If for $T = 1$ it is $\langle E_{ground} \rangle \approx 490$, for $T = 10$, it is $\langle E_{ground} \rangle \approx 610$, and for $T = 100$, it is $\langle E_{ground} \rangle \approx 650$. This has interesting implications, as higher tolerance could mean, in this context, a more stable configuration of the social network.



(a) Total system energy evolution over time

(b) Opinion partition over time

Figure 3.9: *Simulation with fixed rewiring probability, ran for* $t = 6 \cdot 10^6$ *at* $T = 100$, *over a population of 100*

Conclusions

The most important contribution of this thesis is to provide an alternative and — in my opinion — more realistic approach to model opinion dynamics compared to single-layered models. The new approach is based on the postulate that people are influenced by others with already compatible views. The process of selecting the persons one is willing to listen to can ultimately be the factor that leads to consensus, polarization, or otherwise fragmentation. In the two-layered model, agents change their opinions in order to be more in line with their neighbors, while at the same time the network is constantly changing in order to link agents with similar values.

I show that the outcome depends strongly on the fashion and frequency to which the network is updated. A random rewiring dynamic will cause high noise and prolonged fragmentation, while a dynamic based on similarity between agents ultimately leads to consensus. A lower network update ratio will lead to a slower evolution towards system stability and to higher noise in the evolution of opinions. Furthermore, temperature variations evoke the expected behaviour from an Ising-like simulation.

Admittedly, the model is still a long way from maturity, with the analytical descriptions of its behaviour still needing to be done. The exact relationship between the time required to reach the system energy ground state and rewiring frequency is sill unknown, as is the value of critical temperature T_C under default conditions. Besides a better description of the model phenomenology, future steps can include experimenting with different cultural values distributions. While it might be interesting to study models in themselves, ultimately they have to work for the reasons they were created, in this regard it would be a good idea to test the model on networks extracted from empirical data.

References

[1] Cathy O'Neil. *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy.* Crown Books, 2016.

[2] Georg G. Iggers. Further remarks about early uses of the term ”social science”. *Journal of the History of Ideas*, 20(3):433, 1959.

[3] Auguste Comte. Considérations philosophiques sur les sciences et les savants. *Le Producteur, journal philosophique de l'industrie, des sciences et des beaux arts*, I(465), 1825.

[4] Adolphe Quetelet. *Sur lhomme et le développement de ses facultés, ou, Essai de physique sociale.* Bachelier, 1835.

[5] Gustav Jahoda. Quetelet and the emergence of the behavioral sciences. *SpringerPlus*, 4(1), 2015.

[6] Theodore M. Porter. *From Quetelet to Maxwell: Social Statistics and the Origins of Statistical Physics*, page 345–362. Boston Studies in the Philosophy of Science 150. Springer Netherlands.

[7] James Clerk Maxwell. Molecules. *Nature*, Sep 1873.

[8] Ludwig Boltzmann. Über die mechanische bedeutung des zweiten hauptsatzes der wärmetheorie. *Wissenschaftliche Abhandlungen*, 1:4–30, 1866.

[9] Claudio Castellano, Santo Fortunato, and Vittorio Loreto. Statistical physics of social dynamics. *Reviews of Modern Physics*, 81(2):591–646, 2009.

[10] J. Mimkes. Binary alloys as a model for the multicultural society. *Journal of Thermal Analysis*, 43(2):521–537, 1995.

[11] James J. Binney. *The theory of critical phenomena: an introduction to the renormalization group.* Oxford Univ. Press, 2002.

[12] David P. Landau and K. Binder. *A guide to Monte Carlo simulations in statistical physics.* Cambridge University Press, 2015.

[13] F. Y. Wu. The potts model. *Reviews of Modern Physics*, 54(1):235–268, 1982.

[14] François Graner and James A. Glazier. Simulation of biological cell sorting using a two-dimensional extended potts model. *Physical Review Letters*, 69(13), Sep 1992.

[15] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11), Nov 2001.

[16] Robert Axelrod. The dissemination of culture - a model with local convergence and global polarization. *The Journal of Conflict Resolution*, 41(2):203–226, Apr 1997.

[17] D. Stauffer. Social applications of two-dimensional ising models. *American Journal of Physics*, 76(4):470–473, 2008.

[18] Thomas C. Schelling. Dynamic models of segregation†. *The Journal of Mathematical Sociology*, 1(2):143–186, 1971.

[19] Daniel Nettle. Is the rate of linguistic change constant? *Lingua*, 108(2-3):119–136, 1999.

[20] M Hohnisch, S Pittnauer, S Solomon, and D Stauffer. Socioeconomic interaction and swings in business confidence indicators. *Physica A: Statistical Mechanics and its Applications*, 345(3-4):646–656, 2005.

[21] Martin Hohnisch, Dietrich Stauffer, and Sabine Pittnauer. The impact of external events on the emergence of social herding of economic sentiment, Feb 2008.

[22] J. M. Yeomans. *Statistical Mechanics of Phase Transitions*, page 25–27. Clarendon Press, 1992.

[23] Carla M.a. Pinto, A. Mendes Lopes, and J.a. Tenreiro Machado. A review of power laws in real life phenomena. *Communications in Nonlinear Science and Numerical Simulation*, 17(9):3558–3578, 2012.

[24] Mej Newman. Power laws, pareto distributions and zipfs law. *Contemporary Physics*, 46(5):323–351, 2005.

[25] Adrian Cocioceanu, Carina Mihaela Raportaru, Alexandru I. Nicolin, and Dragan Jakimovski. Rank-frequency distributions of romanian words. *AIP Conference Proceedings*, 2017.

[26] Alexis Akira Toda. The double power law in income distribution: Explanations and evidence. *Journal of Economic Behavior & Organization*, 84(1):364–381, 2012.

[27] Albert-Laszló Barabási. *Linked: The New Science of Networks.* Perseus, 2002.

[28] Albert-László Barabási, Réka Albert, and Hawoong Jeong. Scale-free characteristics of random networks: the topology of the world-wide web. *Physica A: Statistical Mechanics and its Applications*, 281(1-4):69–77, 2000.

[29] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[30] Ginestra Bianconi and Albert-László Barabási. Bose-einstein condensation in complex networks. *Physical Review Letters*, 86(24):5632–5635, 2001.

[31] Marian-Gabriel Hâncean. *Rețelele sociale: teorie, metodologie și aplicații.* Polirom, 2014.

[32] Peter J. Carrington and John Scott. *The SAGE handbook of social network analysis.* SAGE Publications, 2014.

[33] Fritz Heider. *The Psychology of Interpersonal Relations.* John Wiley & Sons, 1958.

[34] M. E. J. Newman, Barabási Albert László, and Duncan J. Watts. *The structure and dynamics of networks.* Princeton University Press, 2006.

[35] Stanley Milgram. The small-world problem. *Psychology Today*, May 1967.

[36] Albert-Laszló Barabási. *Linked: The New Science of Networks*, page 27–30. Perseus, 2002.

[37] Rufin Zamfir, Ana Maria Luca, Alexandra Mihaela Ispas, Ana Maria Teaca, Vlad Iavita, and Raluca Andreescu. Monitoring facebook. presidential elections – romania, november 2019. *GlobalFocus Center.*

[38] Harald Holone. The filter bubble and its effect on online personal health information. *Croatian Medical Journal*, 57(3):298–301, 2016.

[39] Bollobás Béla. *Random graphs.* Cambridge University Press, 2004.

Appendices

Simulation Code

A.1 SocialSim.jl

```julia
module SocialSim

using Dates
using Random
using LightGraphs
using DataFrames
using CSV
using Dates
using GraphIO
using Plots
using StatsBase
using ProgressMeter

export run_sim

const J = 1

include("init.jl")
include("analysis.jl")
include("dynamics.jl")
include("storage.jl")

#Runs the simulation at a determined temperature for a determined number of steps
function run_sim(T,population,steps; exports_number=10)

    #Random.seed!(1234)

    #Get time of simulation and prepare folders
    exportTime = Dates.format(Dates.now(), "yyyy-mm-ddTHH-MM-SS")
    mkdir("Data/$exportTime")
    mkdir("Data/$exportTime/Network")


    #Initializing network and data frame
    data = createDataFrame()
    network, nodes = initNetwork(data, population)

    #Export network at each nx steps
    nx = div(steps, exports_number)

    #Executing steps
    @showprogress 5 "Computing..." for i in 1:steps
        ni = rand(1:population)
        procedure2(nodes, network, data, ID=ni, N=population, T=T)

        if mod(i, nx) == 0
            exportNetwork(exportTime, network, nodes, i)
        end
    end

    #Export data to a new folder
    exportData(exportTime, data, network, nodes)
    plotAnalysis(steps, exportTime, data, nodes, network)
end

#Runs the simulation at T for a determined time interval (in hours)
#=function set_run(T,population,duration; exports_number=10)

    #Get time of simulation and prepare folders
    exportTime = Dates.format(Dates.now(), "yyyy-mm-ddTHH-MM-SS")
    mkdir("Data/$exportTime")
    mkdir("Data/$exportTime/Network")

    #Initializing network and data frame
    data = createDataFrame()
    network, nodes = initNetwork(data, population)

    #Export network at each nx steps
    nx = div(steps, exports_number)

    timelimit = Dates.now() + Dates.Minute(duration)

    #Executing steps
    while Dates.now() < timelimit #&& length(Data.E) <= 2000000
        ni = rand(1:population)
        procedure2(nodes, network, data, ID=ni, N=population, T=T)

        if mod(i, nx) == 0
            exportNetwork(exportTime, network, nodes, i)
        end
    end

    #Export data to a new folder
    exportData(exportTime)
    plotAnalysis(length(data.E)-1, exportTime)
end=#

end
```

A.2 init.jl

```julia
struct Agent{T,E}
    id::Int
    values::Vector{T}
    vote::Ref{Int}
    energy::Ref{E}
end

function initNetwork(data, N)
    #Generate Barabasi graph with N nodes, 3 conntections each, 10 initial nodes
    network = barabasi_albert(N, 10, 3, #=seed=1,=# is_directed=true)
    #global Network = erdos_renyi(N, 4, is_directed=true)

    #Initialize agents
    nodes = Agent{Int,Int}[]
    for i in 1:N
        push!(nodes, Agent(i, rand(0:10, 5), Ref(rand(1:10)), Ref(0)))
    end

    #Initialize energy
    for i in 1:N
        dE!(nodes, i, network)
    end
    computeEnergy!(data, nodes)

    #Count initial preferences
    trackPreference!(data, nodes)

    return network, nodes
end
```

A.3 dynamics.jl

```julia
function rewire!(network, nodes, ID, options; remove=false)
    isempty(options) && return

    target = rand(options)

    # number of values
    val_range = axes(nodes[ID].values, 1)
    val_norm = 1 / (val_range[end] * 10)
    p = val_norm * sum(
        i->abs(nodes[ID].values[i]-nodes[target].values[i]),
        val_range)
    if !remove
        p = 1 - p
    end

    if rand() <= p
        #Add pr remove edge
        if remove
            rem_edge!(network, target, ID)
        else
            add_edge!(network, target, ID)
        end

        #Adjust energy
        if nodes[target].vote[] == nodes[ID].vote[]
            j = remove ? J : -J
            nodes[ID].energy[] += j
        end
    end
end

function procedure2(nodes, network, data; ID, N, T)

    #Add new row to data frame
    push!(data, zeros(10+1))

    #Select unconnected node, connect with probability w
    options = Int[]
    for i in 1:N
        if has_edge(network, i, ID) == false
            push!(options, i)
        end
    end

    rewire!(network, nodes, ID, options, remove=false)

    #Select edge, disconnect with probabilty w^-1
    options = inneighbors(network, ID)

    rewire!(network, nodes, ID, options, remove=true)

    #Select new random preference & track preference change
    oldVal = nodes[ID].vote[]
    newVal = rand(1:10)
    nodes[ID].vote[] = newVal

    #Substract current node energy from former node energy
    deltaE = dE(nodes, ID, network) - nodes[ID].energy[]

    p = -(deltaE)/T

    #If deltaE<0 apply it:
    if deltaE < 0
        nodes[ID].energy[] += deltaE
        for i in outneighbors(network, ID)
            nodes[i].energy[] = dE(nodes, i, oldVal, newVal)
        end
    #If deltaE>0 apply it with following probability:
```

```
    elseif rand() < exp(p)
        nodes[ID].energy[] += deltaE
        for i in outneighbors(network, ID)
            nodes[i].energy[] = dE(nodes, i, oldVal, newVal)
        end
    else
        nodes[ID].vote[] = oldVal
    end

    #Log preferences
    trackPreference!(data, oldVal, nodes[ID].vote[])

    #Compute system energy
    computeEnergy!(data, nodes)


    #Optional: change core opinions
end
```

A.4 analysis.jl

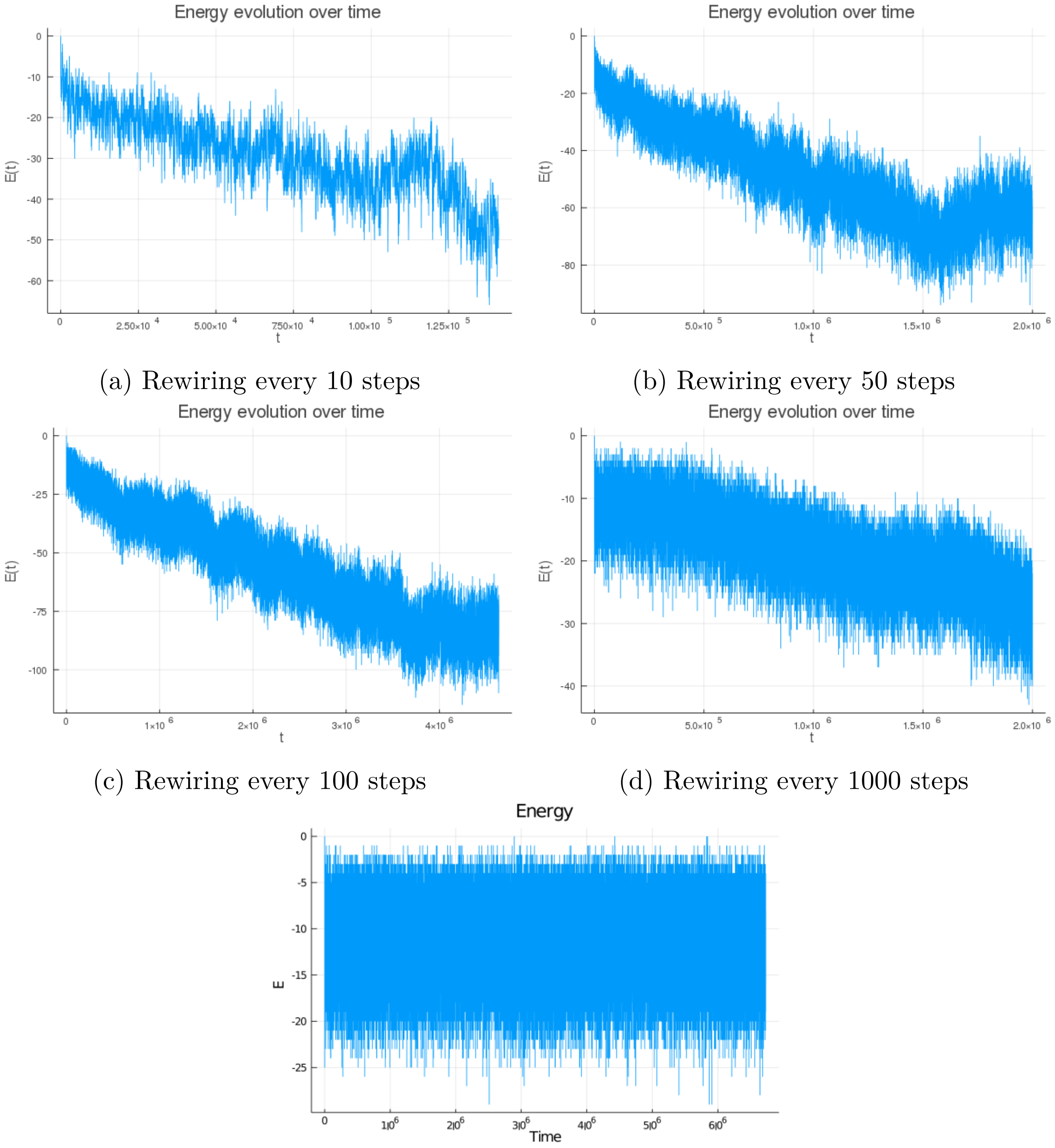
```
"""
    computeEnergy!(data, nodes)

Compute sytem energy
"""
function computeEnergy!(data, nodes)
    E = 0
    for i in 1:length(nodes)
        data.E[end] += nodes[i].energy[]
    end
end

"""
    dE(ID,oldVal,newVal)

Computes node energy by assessing the change in preference of an inneighbor
"""
function dE(nodes, ID, oldVal, newVal)

    if newVal == nodes[ID].vote[] && oldVal == nodes[ID].vote[]
        epsilon = nodes[ID].energy[]
    elseif newVal == nodes[ID].vote[] && oldVal != nodes[ID].vote[]
        epsilon = nodes[ID].energy[] - J
    elseif newVal != nodes[ID].vote[] && oldVal == nodes[ID].vote[]
        epsilon = nodes[ID].energy[] + J
    else
        epsilon = nodes[ID].energy[]
    end
```

```julia

    return epsilon
end

function dE(nodes, ID, network)

    #Goes through inneighbors and computes Potts node energy (epsilon)
    options = inneighbors(network,ID)
    epsilon = 0
    for i in 1:length(options)
        if nodes[ID].vote[] == nodes[options[i]].vote[]
            epsilon += -J
        end
    end

    return epsilon
end

#Slower: recomputes node energy thoroughly
function dE!(nodes, ID, network)

    epsilon = dE(nodes, ID, network)
    #Assign node energy
    nodes[ID].energy[] = epsilon
end

function datacol(data, i)
    getproperty(data, Symbol("c$i"))
end

"""
    trackPreference!(data, nodes)

Count popularity of each candidate
"""
function trackPreference!(data, nodes)
    for i in 1:length(nodes)
        v = nodes[i].vote[]

        # data.c$v[end] += 1
        datacol(data, v)[end] += 1
    end
end

function trackPreference!(data, old, new)
    #Copy previous distribution
    for i=1:10
        datacol(data, i)[end] = datacol(data, i)[end-1]
    end

    #Remove old value
    datacol(data, old)[end] -= 1
    #Add new value
    datacol(data, new)[end] += 1
end

#Plotting
function plotAnalysis(t, dir, data, nodes, network)

    #Energy evolution
    plot1=plot(1:t+1, data.E[1:t+1]#=/Data.E[1]=#, legend=false)
    xlabel!("Time")
    ylabel!("E/E_0")
    title!("Energy")
    png("Data/$dir/Energy")

    #Energy distribution
    energyDistribution=counts(-data.E)
    plot2=plot(1:length(energyDistribution), energyDistribution, legend=
        ↪ false)
    title!("Energy distribution")
    png("Data/$dir/EnergyDistribution")

    #Inneighbor histogram
    noInneighbors = []
    for i in 1:length(nodes)
        push!(noInneighbors, length(inneighbors(network, i)))
    end
    plot3=histogram(noInneighbors, bins = 15)
    title!("Inneighbor histogram")
    png("Data/$dir/InneighborHistogram")
end
```

A.5 storage.jl

```
function createDataFrame()
    DataFrame(E=[0], c1=[0], c2=[0], c3=[0], c4=[0], c5=[0], c6=[0], c7
        ↪ =[0], c8=[0], c9=[0], c10=[0])
end

function exportData(dir, data, network, nodes)
    #Save log
    CSV.write("Data/$dir/data.csv", data)

    #Save nodes table
    nodes_df = DataFrame(ID = Int[], Vote = Int[], sigma_1 = Int[],
        ↪ sigma_2 = Int[], sigma_3 = Int[],
        sigma_4 = Int[], sigma_5 = Int[], E = Int[])

    for i in 1:length(nodes)
        push!(nodes_df, (nodes[i].id, nodes[i].vote[], nodes[i].values[1],
            ↪ nodes[i].values[2],
            nodes[i].values[3], nodes[i].values[4], nodes[i].values[5],
                ↪ nodes[i].energy[]))
    end

    CSV.write("Data/$dir/nodes.csv", nodes_df)

    #Exporting the graph
    savegraph("Data/$dir/graph.net", network, "Network", GraphIO.NET.
        ↪ NETFormat())
end

function exportNetwork(dir, network, nodes, step)
    #Save nodes table
    nodes_df = DataFrame(ID = Int[], Vote = Int[], sigma_1 = Int[],
        ↪ sigma_2 = Int[], sigma_3 = Int[],
        sigma_4 = Int[], sigma_5 = Int[], E = Int[])

    for i in 1:length(nodes)
        push!(nodes_df, (nodes[i].id, nodes[i].vote[], nodes[i].values[1],
            ↪  nodes[i].values[2],
            nodes[i].values[3], nodes[i].values[4], nodes[i].values[5],
                ↪ nodes[i].energy[]))
    end

    CSV.write("Data/$dir/Network/nodes_$step.csv", nodes_df)

    #Exporting the graph
    savegraph("Data/$dir/Network/graph_$step.net", network, "Network",
        ↪ GraphIO.NET.NETFormat())
end
```

Plots



(a) Rewiring every 10 steps

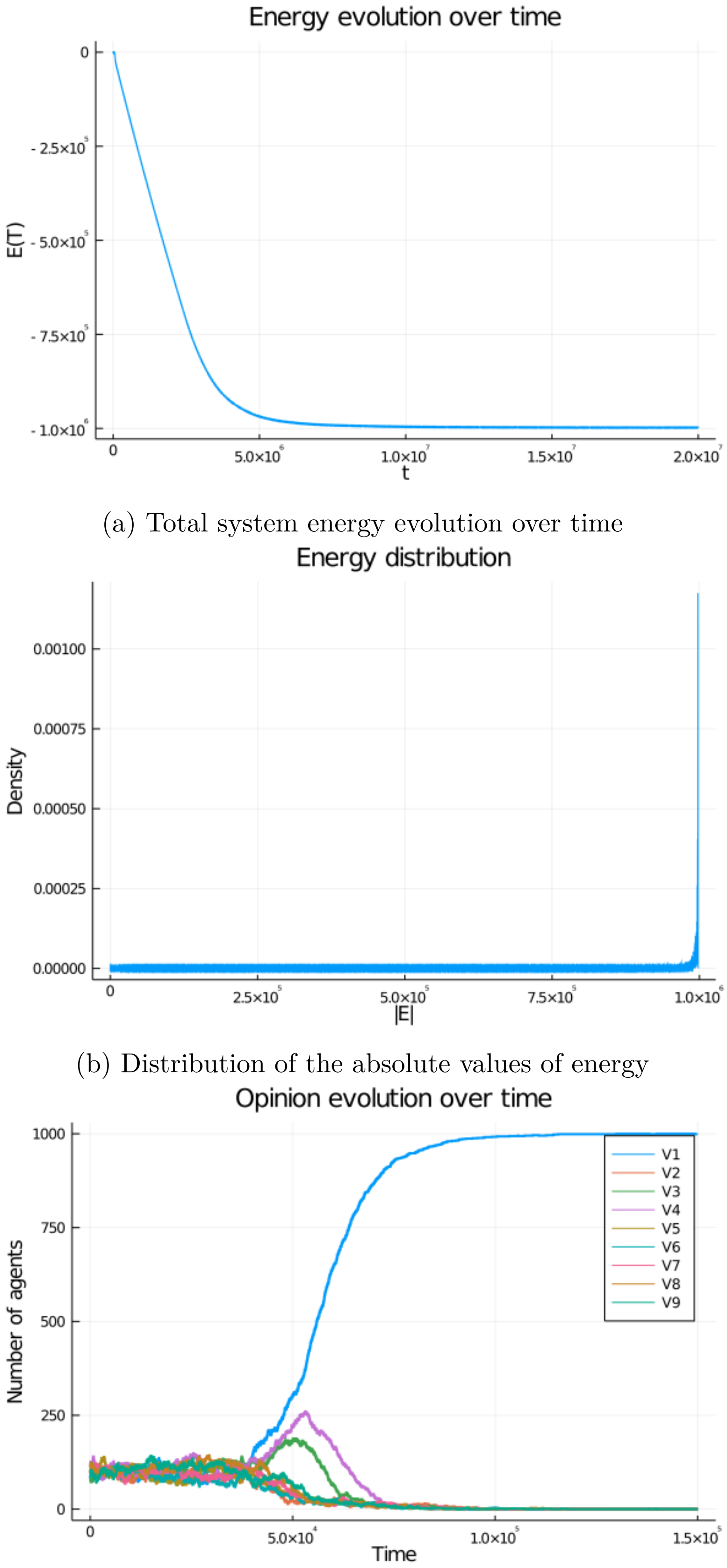
(b) Rewiring every 50 steps

(c) Rewiring every 100 steps

(d) Rewiring every 1000 steps

(e) No rewiring

Figure B.1: *Simulations ran for different rewiring frequencies over a population of 100*

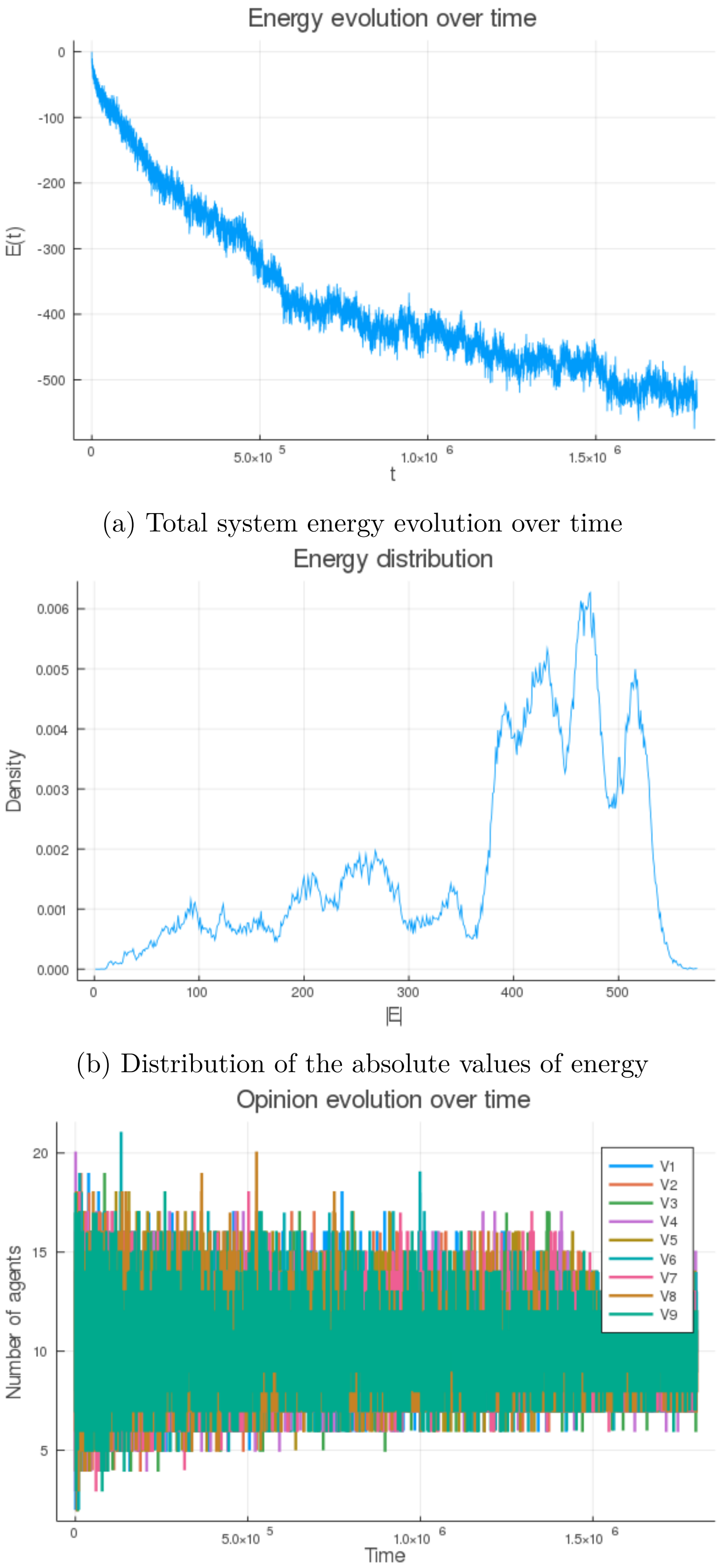


(a) Total system energy evolution over time

(b) Distribution of the absolute values of energy

(c) Opinion partition over time, $t \in \{1..1.5 \cdot 10^5\}$

Figure B.2: *Simulation of final model ran for* $t = 2 \cdot 10^7$ *over a population of 1000*

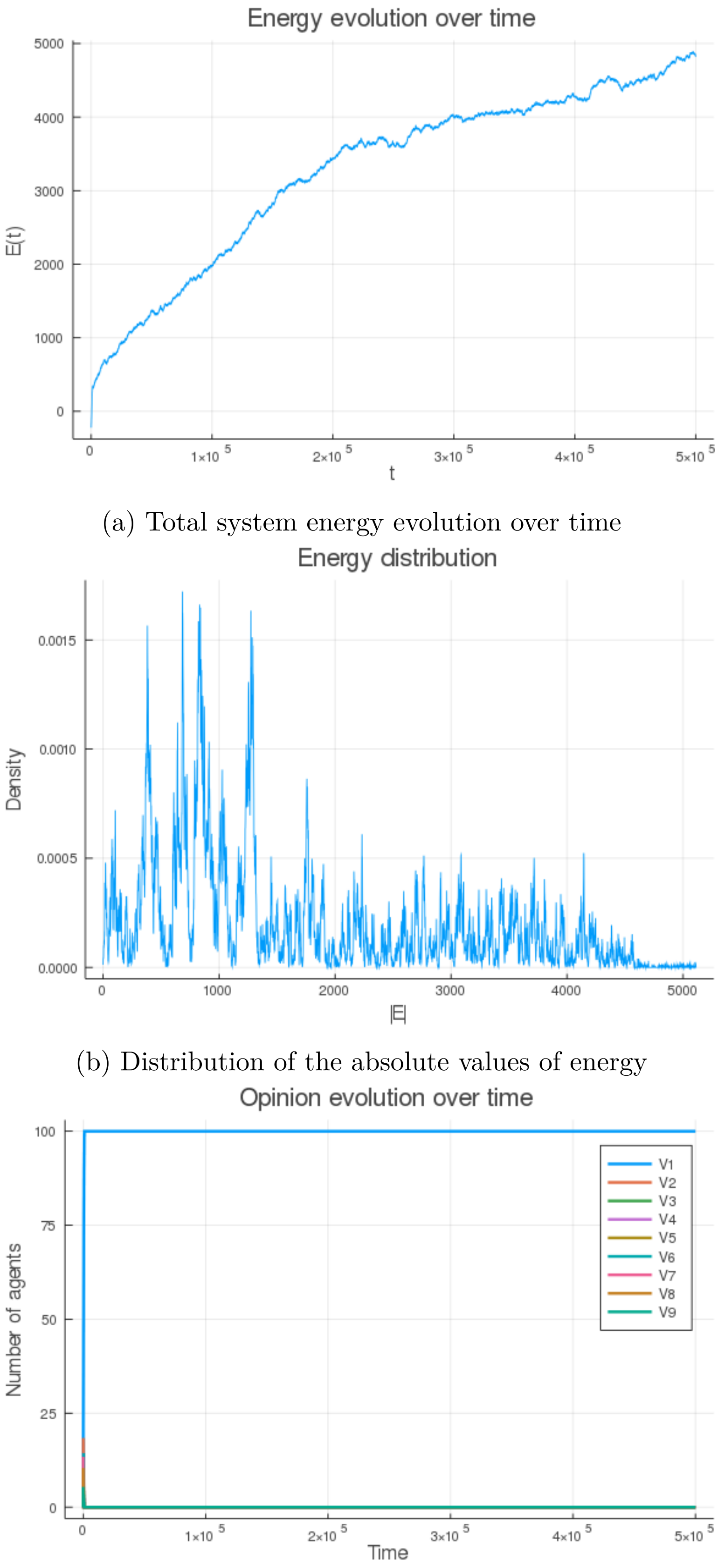


(a) Total system energy evolution over time

(b) Distribution of the absolute values of energy

(c) Opinion partition over time

Figure B.3: *Fixed rewiring probability simulation ran for* $t = 1.8 \cdot 10^6$ *over a population of 100*

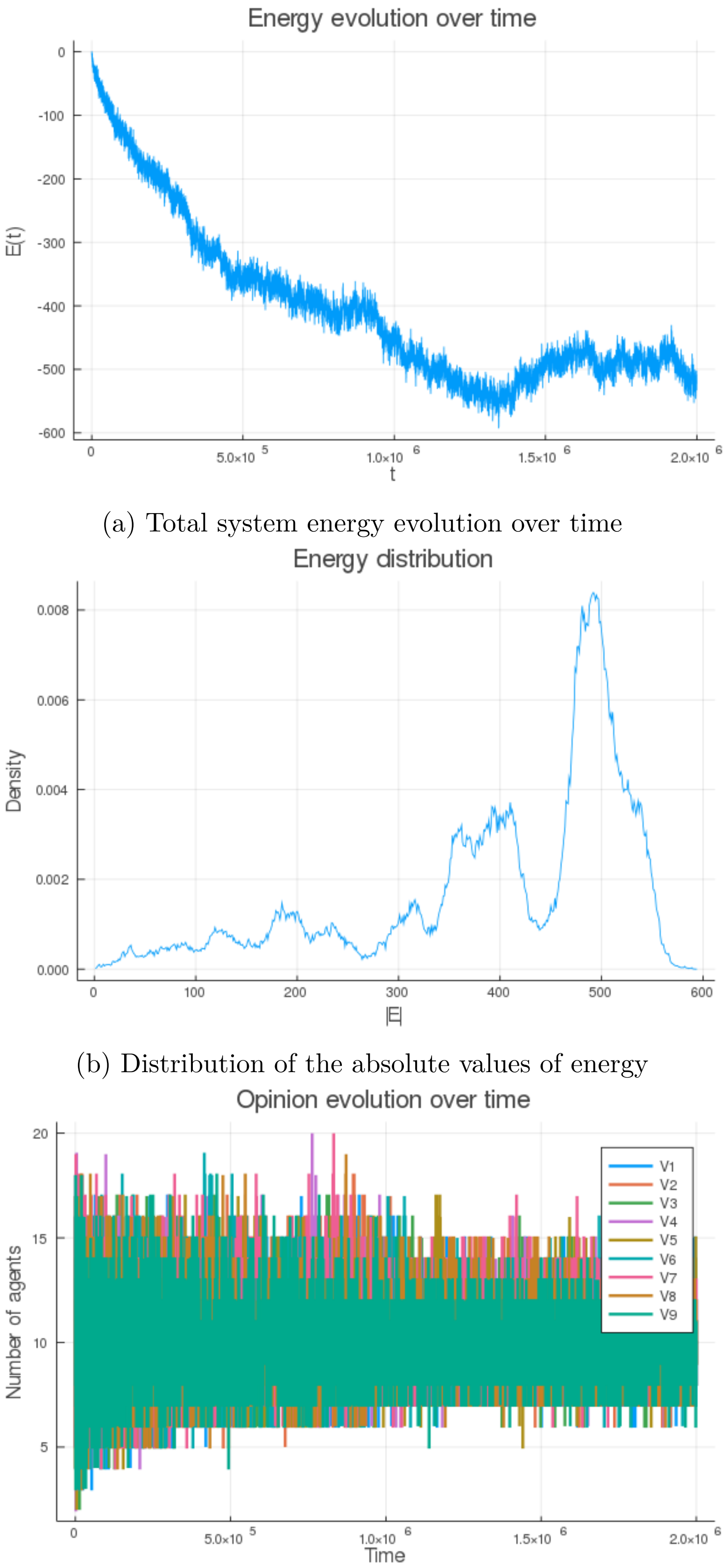


(a) Total system energy evolution over time

(b) Distribution of the absolute values of energy

(c) Opinion partition over time

Figure B.4: *Majority rule simulation ran for* $t = 10^6$ *over a population of 100*

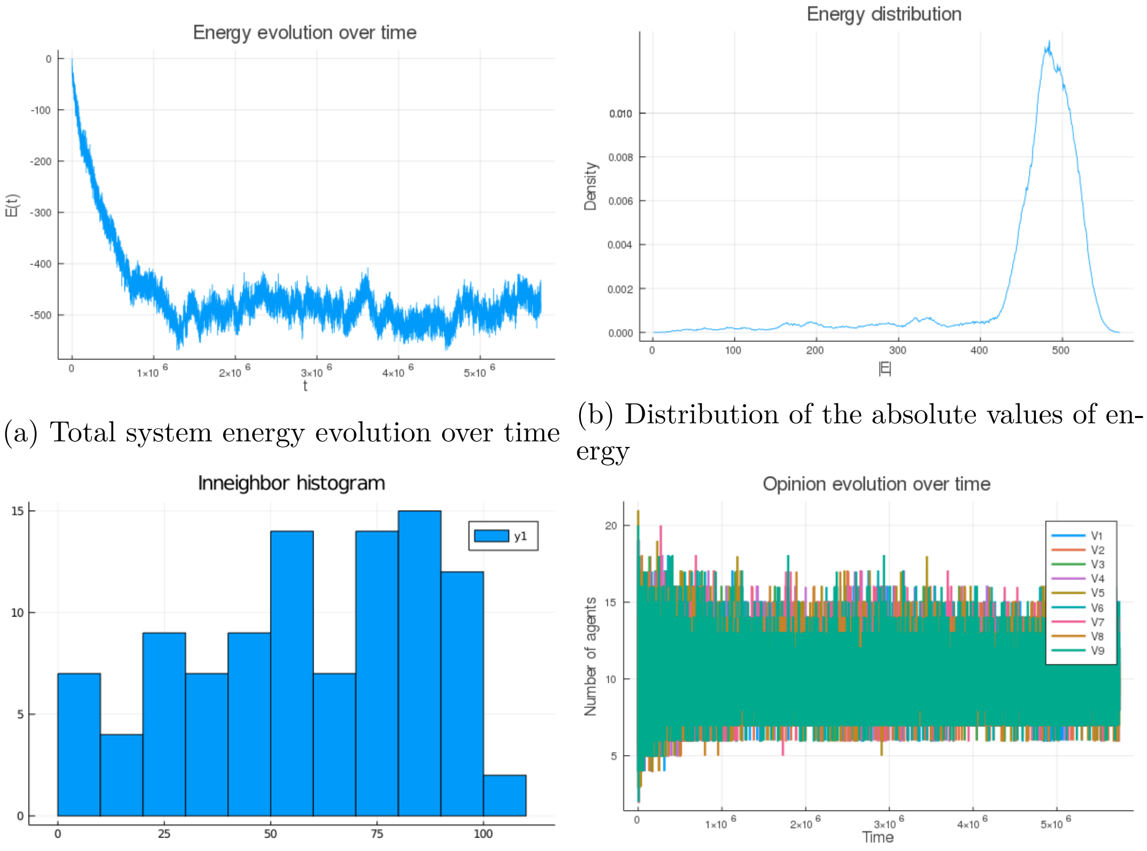


(a) Total system energy evolution over time

(b) Distribution of the absolute values of energy

(c) Opinion partition over time

Figure B.5: *Random initial network simulation ran for $t = 2 \cdot 10^6$ over a population of 100*



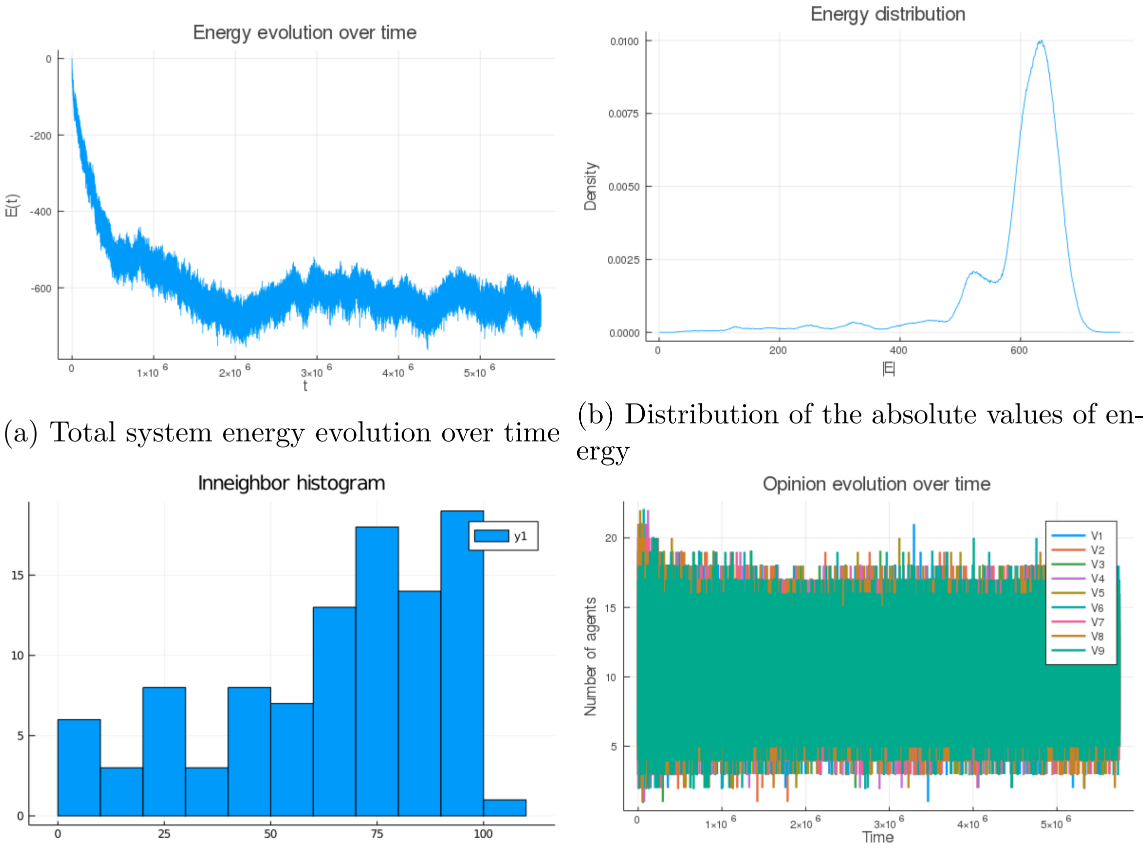
(a) Total system energy evolution over time

(b) Distribution of the absolute values of energy

(c) Inneighbor histogram

(d) Opinion partition over time

Figure B.6: *Simulation with fixed rewiring probability, ran for* $t = 6 \cdot 10^6$ *at* $T = 1$*, over a population of 100*



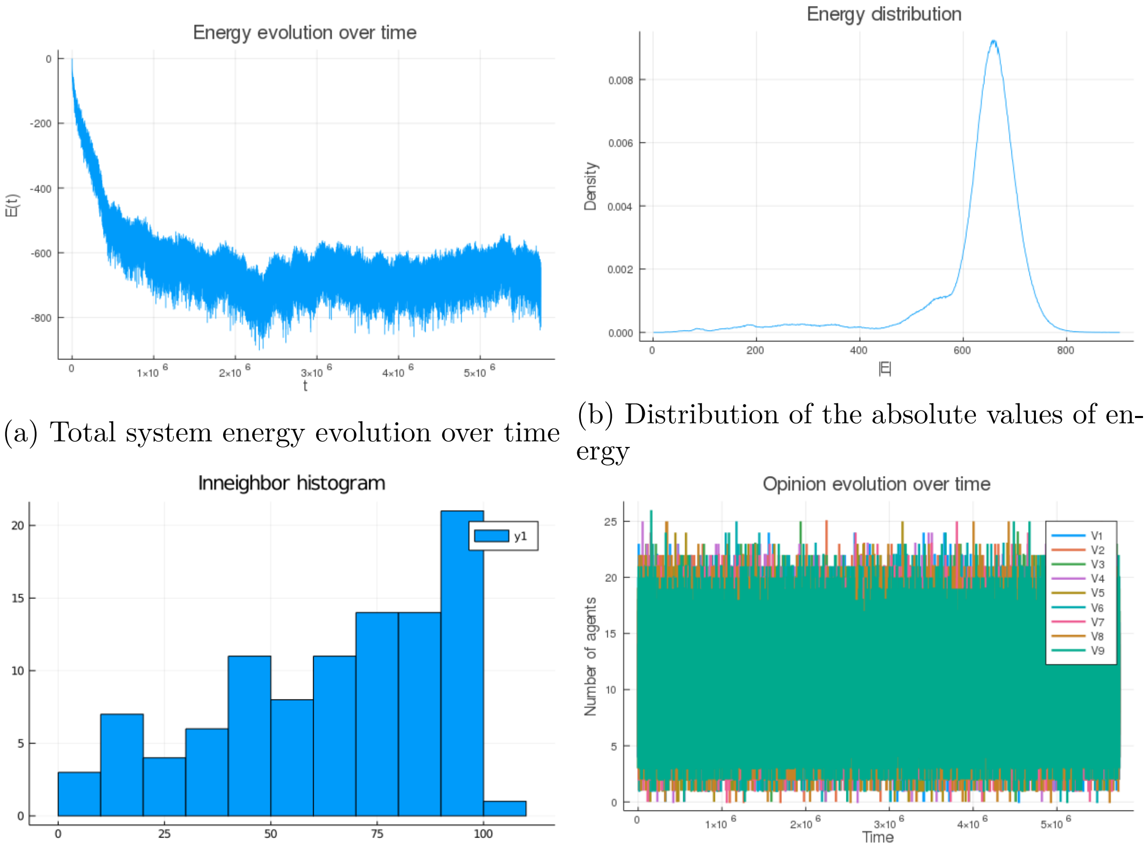
(a) Total system energy evolution over time

(b) Distribution of the absolute values of energy

(c) Inneighbor histogram

(d) Opinion partition over time

Figure B.7: *Simulation with fixed rewiring probability, ran for $t = 6 \cdot 10^6$ at $T = 10$, over a population of 100*



(a) Total system energy evolution over time

(b) Distribution of the absolute values of energy

(c) Inneighbor histogram

(d) Opinion partition over time

Figure B.8: *Simulation with fixed rewiring probability, ran for* $t = 6 \cdot 10^6$ *at* $T = 100$, *over a population of 100*