Network Dynamics and Learning, Homework 3

Giovanni Cadau *Politecnico di Torino*
Turin, Italy
s304861@studenti.polito.it

January 15, 2023

1 Influenza H1N1 2009 Pandemic in Sweden

In this section it is requested to simulate an epidemic with the aim of learning the network-structure characteristics and disease-dynamics parameters of a specific pandemic.

The disease propagation model used to simulate the epidemic is a discrete-time simplified version of the *SIR* epidemic model. Given a finite populations of agents, at any time $t \in N$, agent $i \in \mathcal{V}$ is in a state $X_i(t) \in \mathcal{A} = \{S, I, R\}$. The population of agents is identified with the node set of a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. In particular, agents represent individuals of a certain population and their state can be: susceptible (denoted as S in the state vector $\mathcal{A}$), infected (denoted as I) or recovered (denoted as R). An evolution process for the state vector $X(t)$ is considered, caused by pairwise interactions between neighbouring nodes in $\mathcal{G}$ and by spontaneous mutations, subjected to the following disease-dynamics parameters:

- $\beta \in [0, 1]$: probability that the infection is spread from an infected individual to a susceptible one (given that they are connected by a link) during one time step
- $\rho \in [0, 1]$: probability that an infected individual will recover during one time step.

In addition to this, the disease propagation model used can be enriched by considering the possibility to administer vaccination to (part of) the population of individuals by means of a vector $Vacc(t)$, where $Vacc_t$ represents the total fraction of population that has received vaccination by time unit t. The individuals to vaccinate are randomly selected from the population that has not yet received vaccination. According to this possibility, the state vector $\mathcal{A}$ is such that $\mathcal{A} = \{S, I, R, SV, IV, RV\}$, where SV stands for 'susceptible but vaccinated', IV for 'infected but vaccinated' and RV for 'recovered but vaccinated'. Regardless of the state of an individual prior to the vaccination, once vaccinated the individual will not be able to become infected nor infect any other individuals.

1.1 Epidemic simulation on a known graph, without vaccination

Firstly, the epidemic is simulated on a known graph $\mathcal{G}_{known}$.

In particular, $\mathcal{G}_{known}$ is chosen as a symmetric k-regular undirected graph with node set $\mathcal{V} = \{1, 2, .., n\}$, where every node is directly connected to the k nodes whose index is closest to their own modulo n.

The following are the parameters used for the simulation:

- $n = 500$
- $k = 4$
- $\beta = 0.3$
- $\rho = 0.7$

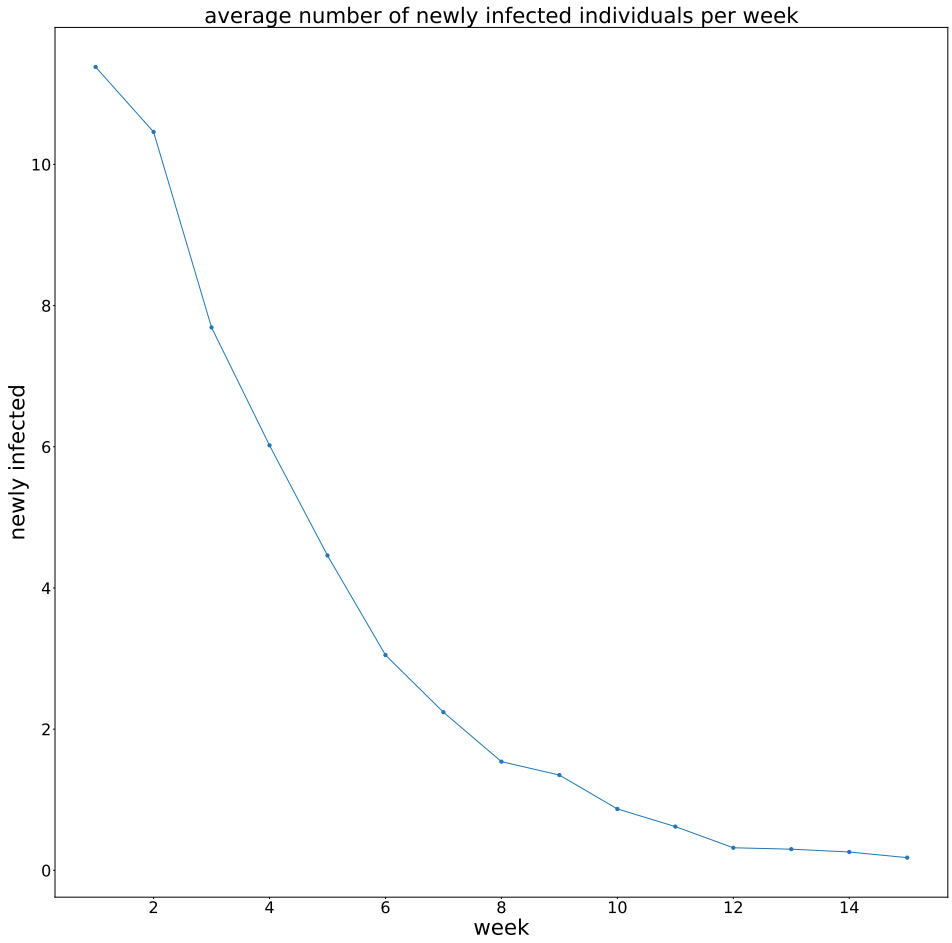

Figure 1: average number of newly infected individuals per week

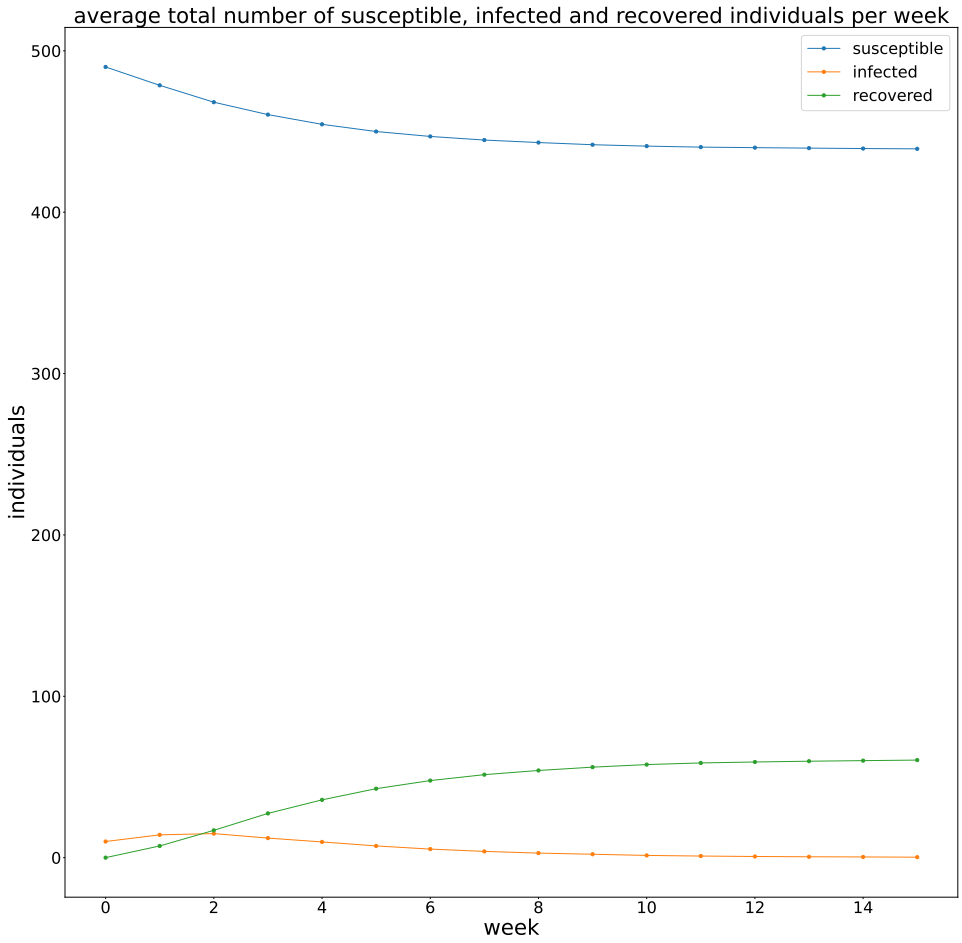

Figure 2: average total number of susceptible, infected and recovered individuals at each week

With 1 week as unit of time, one single simulation is run for 15 weeks.

As initial configuration is chosen a configuration with 10 infected nodes selected at random from the node set.

100 different simulation has been run to evaluate the state of agents at different weeks.

Fig 1 shows the average number of newly infected individuals each week and Fig 2 the average total number of susceptible, infected and recovered individuals at each week.

1.1.1 Test of different initial configurations

To evaluate how the choice of the initial configuration influence the epidemic simulation, different simulations with the same parameter set of the simulations described in Section 1 but with different initial configurations have been run. In particular, for each initial configuration 100 simulations have been run and the average values are evaluated.

The following are the different initial configurations tested. They all involve 10 infected nodes selected in different ways:

- infected nodes randomly chosen
- infected nodes randomly chosen such that they are equally spaced in the graph (i.e., their index numbers are randomly chosen such that they are equally spaced in the space $\{1, 2, ..., n\}$)
- infected nodes randomly chosen all grouped in the graph (i.e., their index numbers are randomly chosen such that they are consecutive numbers in the space $\{1, 2, ..., n\}$)
- infected nodes randomly chosen such that they are grouped in 2 groups equally spaced in the graph
- infected nodes randomly chosen such that they are grouped in 3 groups equally spaced in the graph
- infected nodes randomly chosen such that they are grouped in 4 groups equally spaced in the graph
- infected nodes randomly chosen such that they are grouped in 5 groups equally spaced in the graph

Fig 3 shows the average number of newly infected individuals each week and Fig 4, 5, 6 the average total number of susceptible, infected and recovered individuals at each week for different configurations, respectively.

As can be seen from the plots, in the first weeks, the number of newly infected individuals is higher when the initial infected nodes are not grouped at all, since there is a higher number of nodes that can be infected with a higher probability (this case can also be seen as 10 different groups equally spaced, each with a single node). The number of newly infected individuals in the first weeks decreases as the number of groups decreases, reaching the minimum value when there is only 1 group (i.e., all nodes are grouped). However, in the last weeks, the number of newly infected individuals converges to a similar value between all configurations. The number of susceptible and recovered individuals at each weeks is different between different configuration, in accordance with the variation of the number of newly infected individuals between different configurations described above. The number of susceptible individuals increases as the number of groups decreases and the opposite happens for the number of recovered individuals.

1.2 Epidemic simulation on a random graph, without vaccination

As a second step, the epidemic is simulated on a random graph $\mathcal{G}_{random}$.

In particular, $\mathcal{G}_{random}$ is generated according to the *preferential attachment* model such that average degree is close to k, in the following way. At time $t = 1$ is generated an initial graph $\mathcal{G}_1$, that is complete with $k+1$ nodes. Then at every time $t \geq 2$, a new graph $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t)$ is created by adding a new node to $\mathcal{G}_{t-1}$ and connecting it to $\frac{k}{2}$ existing nodes $\mathcal{V}_{t-1}$ of $\mathcal{G}_{t-1}$ randomly chosen according to a probability proportional to the current degree of the node it is connecting to. The procedure is repeated until the cardinality of the set $\mathcal{V}$ reaches n.

The following are the parameters used for the simulation:

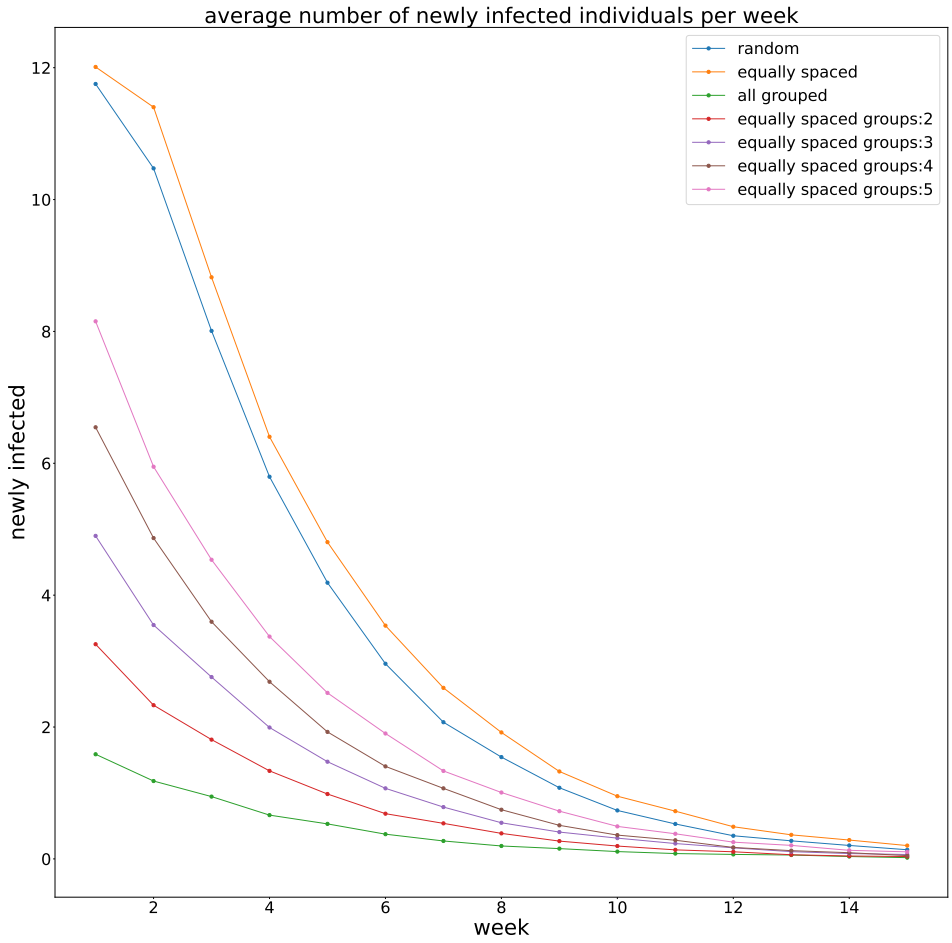

Figure 3: average number of newly infected individuals per week for different initial configurations

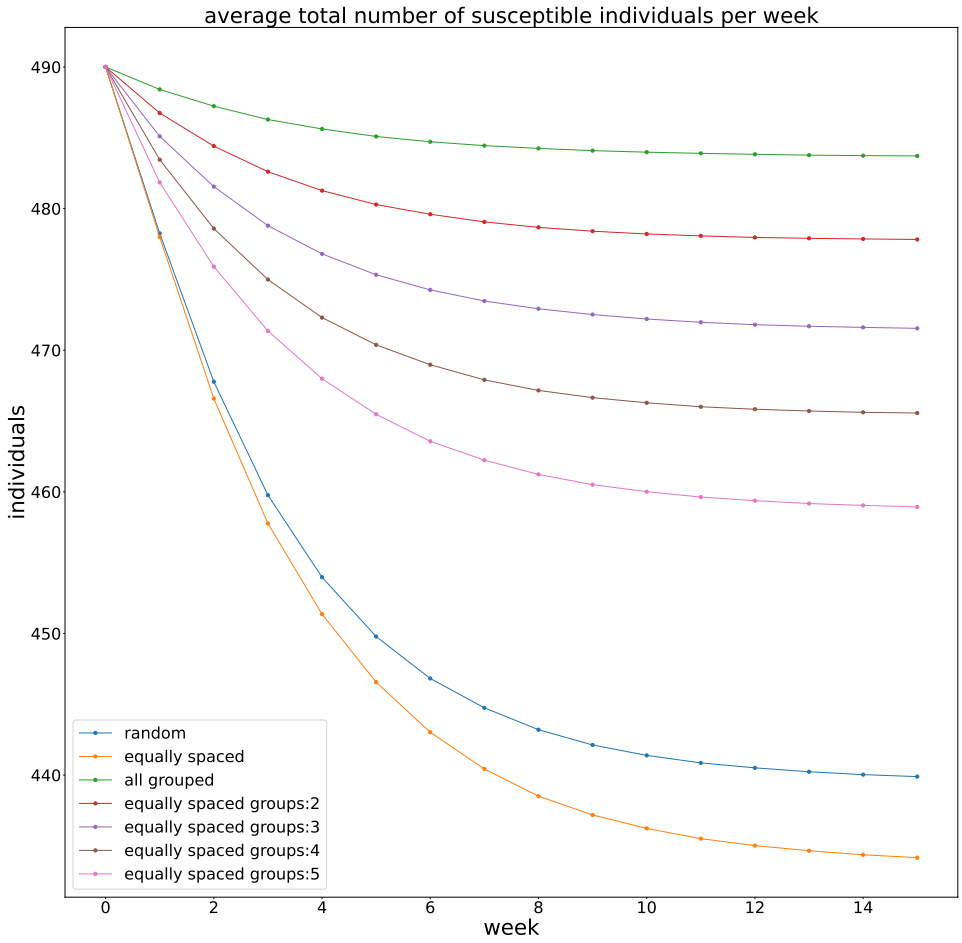

Figure 4: average total number of susceptible individuals at each week for different initial configurations

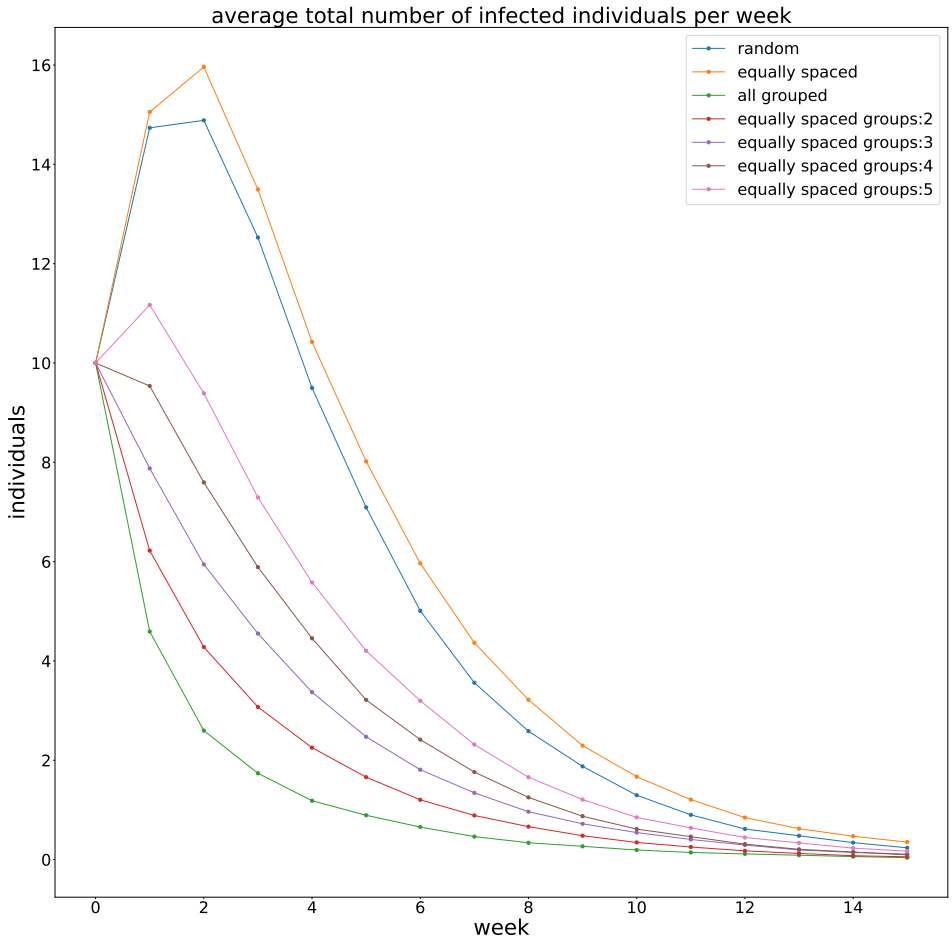

Figure 5: average total number of infected individuals at each week for different initial configurations

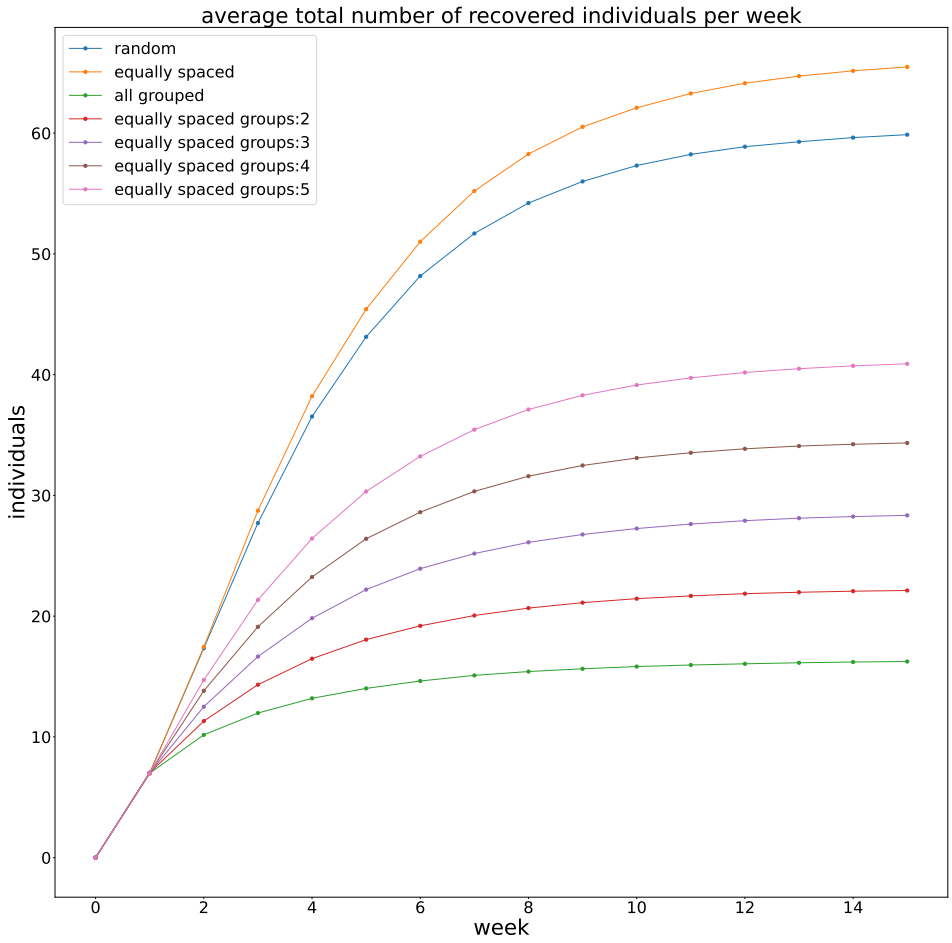

Figure 6: average total number of recovered individuals at each week for different initial configurations

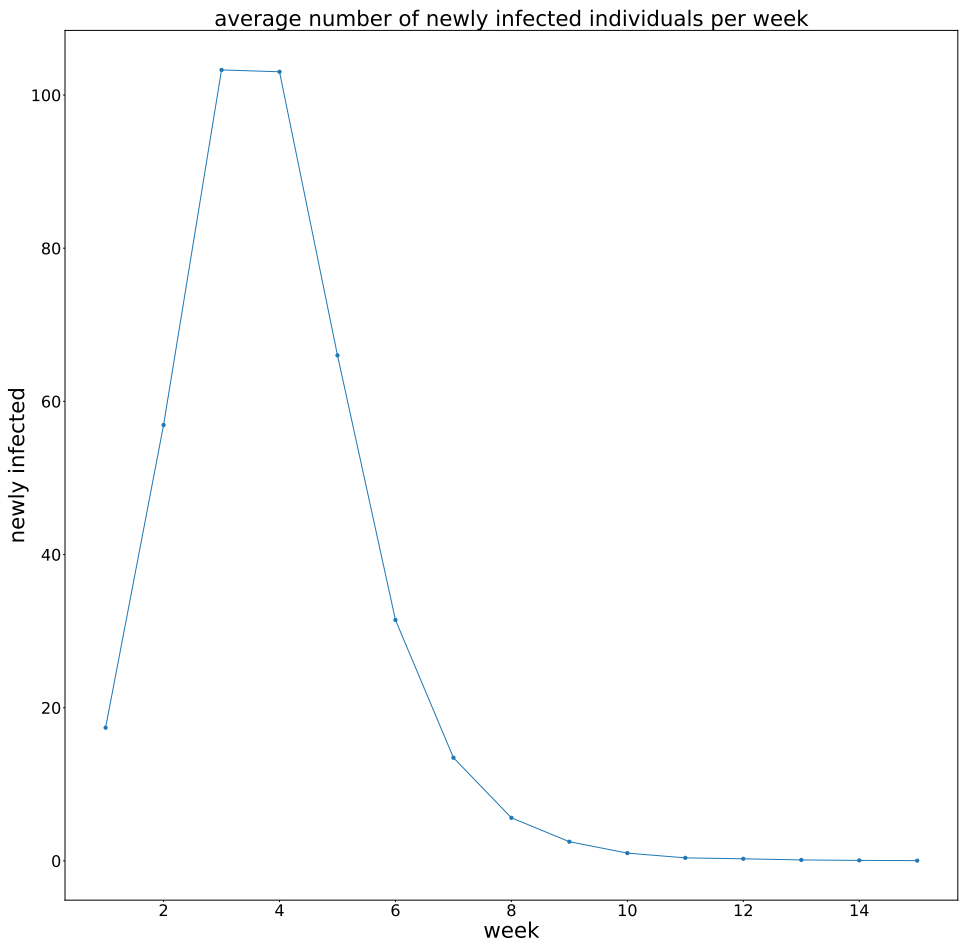

Figure 7: average number of newly infected individuals per week

- $n = 500$
- $k = 6$
- $\beta = 0.3$
- $\rho = 0.7$

With 1 week as unit of time, one single simulation is run for 15 weeks.

As initial configuration is chosen a configuration with 10 infected nodes selected at random from the node set.

100 different simulation has been run to evaluate the state of agents at different weeks.

Fig 7 shows the average number of newly infected individuals each week and Fig 8 the average total number of susceptible, infected and recovered individuals at each week.

1.2.1 Test of different initial configurations

The same initial configurations described in Subsection 1.1.1 are tested, with the same set of parameters. In particular, for each initial configuration 100 simulations have been run on $\mathcal{G}_{random}$ and the

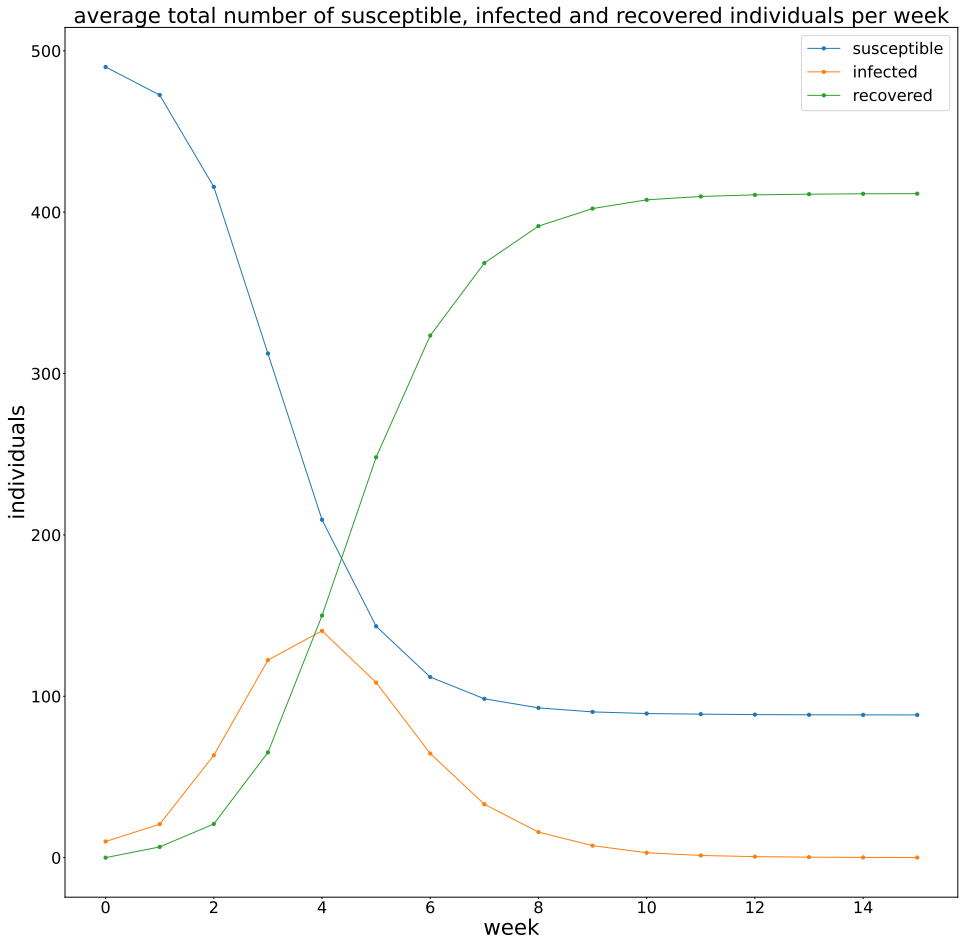

Figure 8: average total number of susceptible, infected and recovered individuals at each week

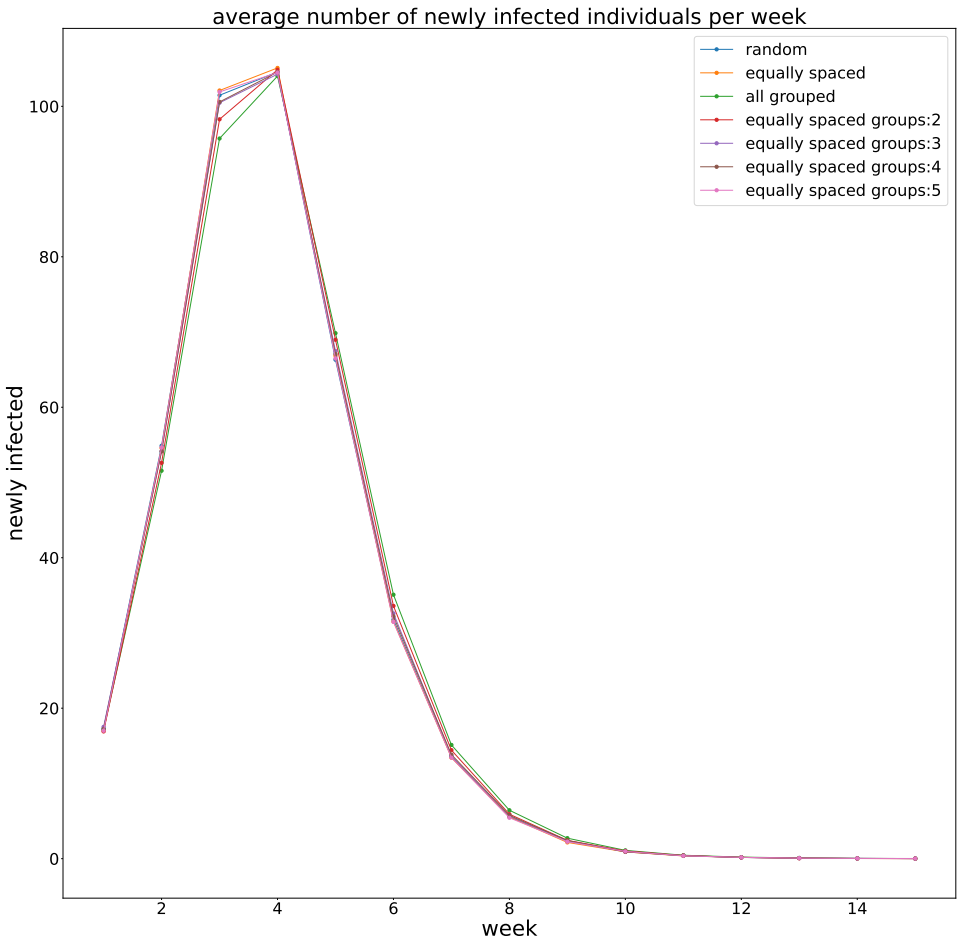

Figure 9: average number of newly infected individuals per week for different initial configurations

average values are evaluated. Since on $\mathcal{G}_{random}$ the position of a node is not related to which nodes it is connected to, in this case different configurations don't produce significant changes in the states of the nodes, as can be seen in Fig 9, showing the average number of newly infected individuals each week for different configurations. Plots showing the average total number of susceptible, infected and recovered individuals at each week for different configurations, respectively, can be found in the attached code.

1.3 Epidemic simulation on a random graph, with vaccination

The simulation is run on the same graph used in Section 1.2, with the same parameters n, k, β, ρ but taking into account the vaccination received by the population, according to the vector $Vacc(t)$:

$$\begin{array}{rccccccccccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\ Vacc = & (0 & 5 & 15 & 25 & 35 & 45 & 55 & 60 & 60 & 60 & 60 & 60 & 60 & 60 & 60) \end{array}$$

With 1 week as unit of time, one single simulation is run for 15 weeks.

As initial configuration is chosen a configuration with 10 infected nodes selected at random from the node set.

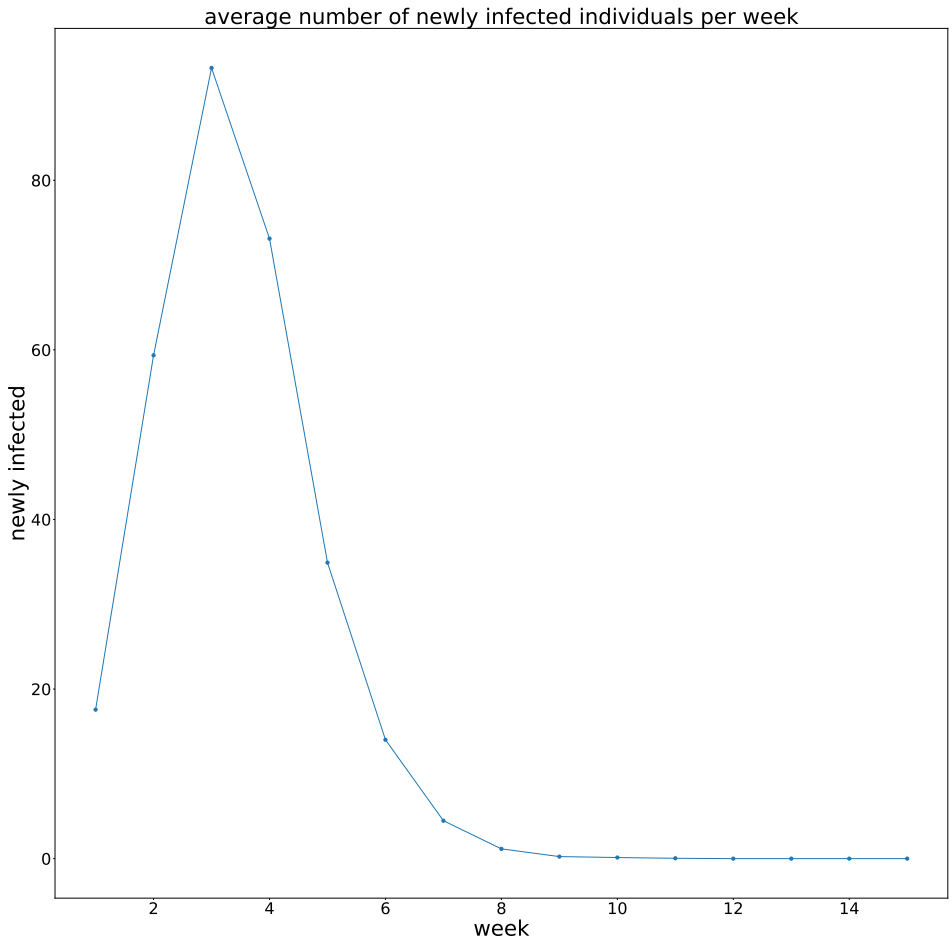

Figure 10: average number of newly infected individuals per week

100 different simulation has been run to evaluate the state of agents at different weeks.

Fig 10 shows the average number of newly infected individuals each week, Fig 11 the average number of newly vaccinated individuals each week and Fig 12 the average total number of susceptible, infected, recovered and vaccinated individuals at each week.

1.3.1 Test of different initial configurations

The same initial configurations described in Subsection 1.1.1 are tested, with the same set of parameters. In particular, for each initial configuration 100 simulations have been run on $\mathcal{G}_{random}$ and the average values are evaluated. According to the considerations stated in Subsection 1.2.1, also in this case different configurations don't produce significant changes in the states of the nodes, as can be seen from the plots, available in the attached code, showing the average number of newly infected and newly vaccinated individuals each week for different configurations and the average total number of susceptible, infected, recovered and vaccinated individuals at each week for different configurations, respectively.

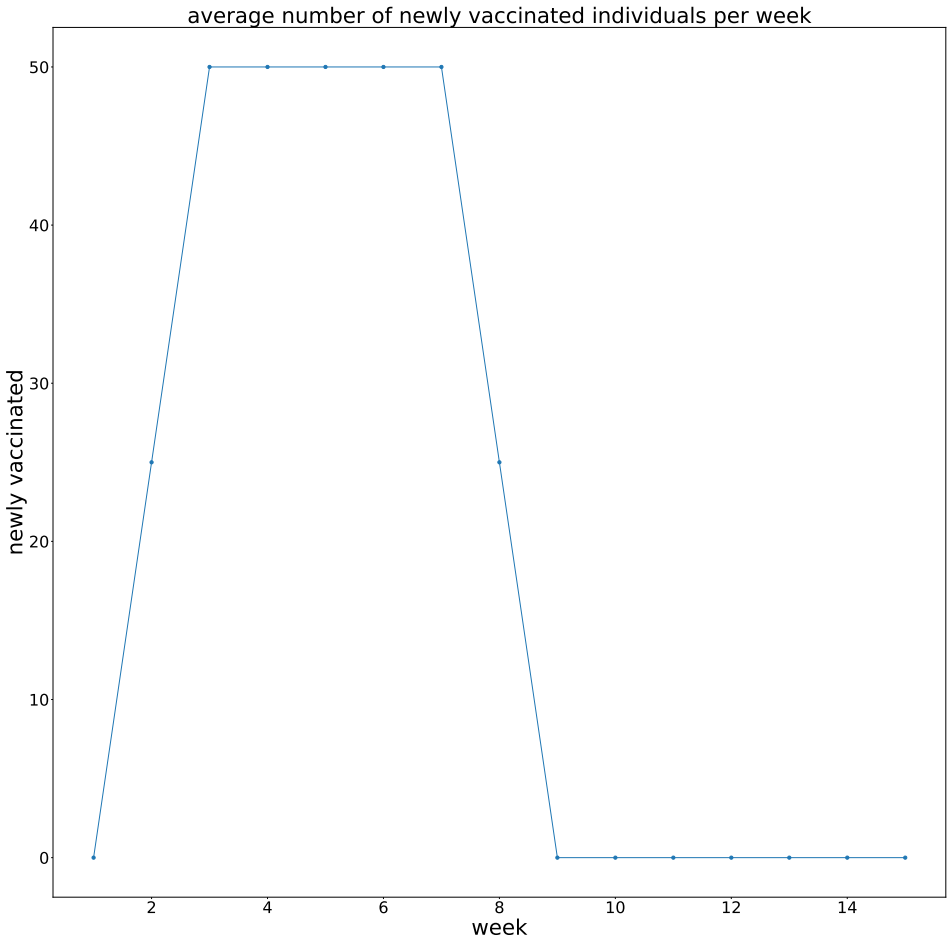

Figure 11: average number of newly vaccinated individuals per week

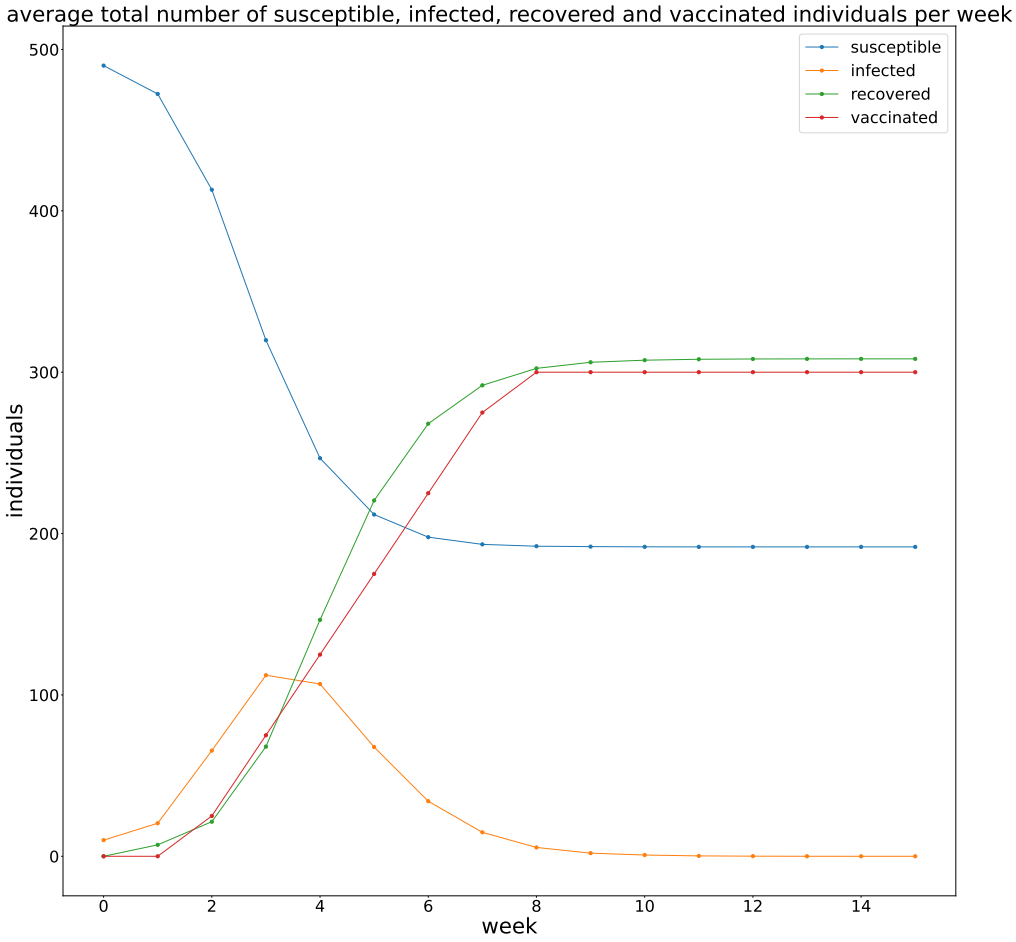

Figure 12: average total number of susceptible, infected, recovered and vaccinated individuals at each week

1.4 Population social structure and disease-spread parameters estimation

With respect to the H1N1 pandemic in Sweden during the fall of 2009[1], the pandemic between week 42, 2009 and week 5, 2010 is simulated[2] on a random graph $\mathcal{G}_{random}$ generated according to the *preferential attachment* model, with $n = 934$, given the vector $Vacc(t)$.

An algorithm exploiting a gradient-based search over the parameter space of k, β, ρ is used in order to find the set of parameters that best matches the real pandemic.

Starting from k_0, β_0, ρ_0 along with same Δk, $\Delta\beta$, $\Delta\rho$, for each set of parameters (k, β, ρ) in the parameter space $(k \in \{k_0 - \Delta k, k_0, k_0 + \Delta k\}, \beta \in \{\beta_0 - \Delta\beta, \beta_0, \beta_0 + \Delta\beta\}, \rho \in \{\rho_0 - \Delta\rho, \rho_0, \rho_0 + \Delta\rho\})$, $N = 10$ simulations have been run in order to evaluate the root-mean-square error (RMSE) between the simulation and the real pandemic:

$$RMSE = \sqrt{\frac{1}{n_t}\sum_{t=1}^{n_t}(I(t) - I_0(t))^2},$$

where n_t is the number of time units, $I(t)$ is the average number of newly infected individuals each week in the simulation[3] and $I_0(t)$ is the true value of newly infected individuals each week.

k_0, β_0, ρ_0 are updated to the set of parameters yielding the lowest RMSE, stopping the algorithm when the same set of parameters is found.

1.4.1 Gradient-based search algorithm implementation

The first thing that has been noticed is related to the choice of N. With $N = 10$, as described in Section 1.4 the value of the *RMSE* between different simulations for the same configuration of parameters (k, β, ρ) is not stable, with variations of up to 500%. To overcome this issue, N has been set to 100, leading to much less significant variations.

The algorithm implementation has been run with:

$$k_0 = 10, \beta_0 = 0.3, \rho_0 = 0.6; \Delta k = 1, \Delta\beta = 0.1, \Delta\rho = 0.1$$

and led to the following results:

$$optimum = \begin{matrix} \text{k} & \beta & \rho \\ (8 & 0.2 & 0.6) \end{matrix}$$

corresponding to $RMSE = 5.21180$.

1.4.2 Other algorithms implementation

In the following, other algorithms are proposed in order to try to better estimate the parameters.

1.4.2.1 Different values for $\Delta k, \Delta\beta, \Delta\rho$

The algorithm described in Subsection 1.4.1 is iterated M times, each one with different values of $\Delta k, \Delta\beta, \Delta\rho$. In particular, in each iteration their values are reduced by a factor *step*.

With:

$$M = 8, steps = 0.75;$$

$$k_0 = 20, \beta_0 = 0.5, \rho_0 = 0.5;$$

and, in the 1^{st} iteration:

[1] whose data describing the newly infected and vaccinated individuals each week have been taken from a report by the Swedish Civil Contingencies Agency (Myndigheten for samhallsskyddoch beredskap, MSB) and the Swedish Institute for Communicable Disease Control (Smittskyddsinstitutet, SMI)

[2] the population of Sweden is scaled down by a factor of 10^4

[3] the average number between the N simulations

$$\Delta k = 20, \Delta\beta = 0.5, \Delta\rho = 0.5$$

the following results have been obtained:

$$optimum = \begin{matrix} \text{k} & \beta & \rho \\ (16 & 0.09375 & 0.498) \end{matrix}$$

corresponding to $RMSE = 5.45668$.

1.4.2.2 Search for different local minima

The algorithm described in Subsection 1.4.1 is able to find a single local minimum, dependent of the initial choice of (k_0, β_0, ρ_0). To overcome this limit, the algorithm is repeated M times, in order to reach different local minima, hoping that among them there is the global minimum.

Each iteration, starting from a configuration (k_0, β_0, ρ_0), leads to a local minimum corresponding to configuration (k_i, β_i, ρ_i). To reach different local minima, a perturbation is applied to the configuration (k_i, β_i, ρ_i) in order to get a new configuration $(k_0^*, \beta_0^*, \rho_0^*)$ to which re-apply (i.e., starting from it) the algorithm in the following iteration. The perturbation is obtained by generating 6 different neighbours of the current local minimum (they are inserted in a FIFO queue and extracted in the following iterations, one neighbour per iteration) in the following way:

$$(k_0^*, \beta_0^*, \rho_0^*)_1 = (k_i + I\Delta k, \beta_i, \rho_i)$$

$$(k_0^*, \beta_0^*, \rho_0^*)_2 = (k_i - I\Delta k, \beta_i, \rho_i)$$

$$(k_0^*, \beta_0^*, \rho_0^*)_3 = (k_i, \beta_i + I\Delta\beta, \rho_i)$$

$$(k_0^*, \beta_0^*, \rho_0^*)_4 = (k_i, \beta_i - I\Delta\beta, \rho_i)$$

$$(k_0^*, \beta_0^*, \rho_0^*)_5 = (k_i, \beta_i, \rho_i + I\Delta\rho)$$

$$(k_0^*, \beta_0^*, \rho_0^*)_5 = (k_i, \beta_i, \rho_i - I\Delta\rho)$$

where I represents the increment.
With:

$$M = 50, I = 3;$$

$$k_0 = 10, \beta_0 = 0.3, \rho_0 = 0.6;$$

$$\Delta k = 1, \Delta\beta = 0.1, \Delta\rho = 0.1$$

the following (potential) global minimum[4] has been obtained:

$$optimum = \begin{matrix} \text{k} & \beta & \rho \\ (15 & 0.1 & 0.6) \end{matrix}$$

corresponding to $RMSE = 4.71775$.

[4] minimum value between all local minima

1.4.2.3 CMA-ES optimization algorithm

In order to minimize the RMSE, computed as above, the *Covariance matrix adaptation evolution strategy* (CMA-ES)[5] numerical optimization algorithm has been used.

The algorithm[6] tries to optimize the parameters (k, β, ρ) based on an objective function to be minimized. The latter is built by computing the RMSE as explained in Section 1.4.

The algorithm has been run for 1000 iterations (i.e., number of evaluations of the objective function) with the following hyperparameters:

- search space for parameter k: $k \in [1, 20]$
- search space for parameter β: $\beta \in [0, 1]$
- search space for parameter ρ: $\rho \in [0, 1]$
- initial value for parameter k, β, ρ: half of their search space
- normalization = *True*: parameters are normalized to space $[a, b]$ before being passed to the optimizer and denormalized back to their original space when the objective function is evaluated. a and b chosen as:
  - $a = 0$
  - $b = 4$

The following is the optimum obtained:

$$optimum = \begin{matrix} \text{k} & \beta & \rho \\ (4 & 0.39562 & 0.40448) \end{matrix}$$

corresponding to $RMSE = 4.87677$.

1.4.3 Other graphs generation

In the following, other graphs are proposed in order to try to better estimate the parameters.

In particular, it has been tried to increase the cluster coefficient of a graph. Since it describes *how many neighbours of a given node are neighbours to each other* and the graph used in this context represents the interaction between individuals, could be useful to improve the cluster coefficient so that the random graph is much similar to a real network graph.

The cluster coefficient has been changed in the following way.

In the generation of a random graph according to the preferential attachment model, $c = \frac{k}{2}$ nodes are extracted each iteration and connected to the new node to be added to the graph.

A portion of them, corresponding to cp are now extracted such that $\frac{cp}{2}$ triangles are added in the graph (1^{st} extraction) and the remaining $c(1-p)$ nodes are extracted with a probability proportional to their degree (2^{nd} extraction). $p \in [0, 1]$ is a real number which represents the fraction of nodes selected in the 1^{st} extraction.

In particular, if two nodes are already connected by a link, then connecting the new node to both nodes will generate a new triangle in the graph. The 1^{st} extraction is therefore realized by randomly selecting (with a uniform probability) $\frac{cp}{2}$ edges from the list of all edges between 2 nodes a and b, such that, by connecting the new node to a and b, a new triangle in the graph is generated. The corresponding cp nodes are connected to the new node. The 2^{nd} extraction is realized among nodes not extracted yet.

Fig 13 shows the comparison of the cluster coefficient of two different families graphs: the first generated according to the standard model, described in section 1.2, the second according to this procedure, with 'cluster parameter' $p = 0.5$.

$N = 1000$ simulations have been run on a random graph generated with the algorithm described in this Subsection for each value of $p \in \{0.25, 0.5, 0.75\}$, with parameters (k, β, ρ) such that $k = 15, \beta = 0.1, \rho = 0.6$.

The following are the $RMSE$ between the simulation and the real pandemic for each value of p:

[5] Covariance matrix adaptation evolution strategy (CMA-ES)

[6] whose is implementation is provided by Nevergrad

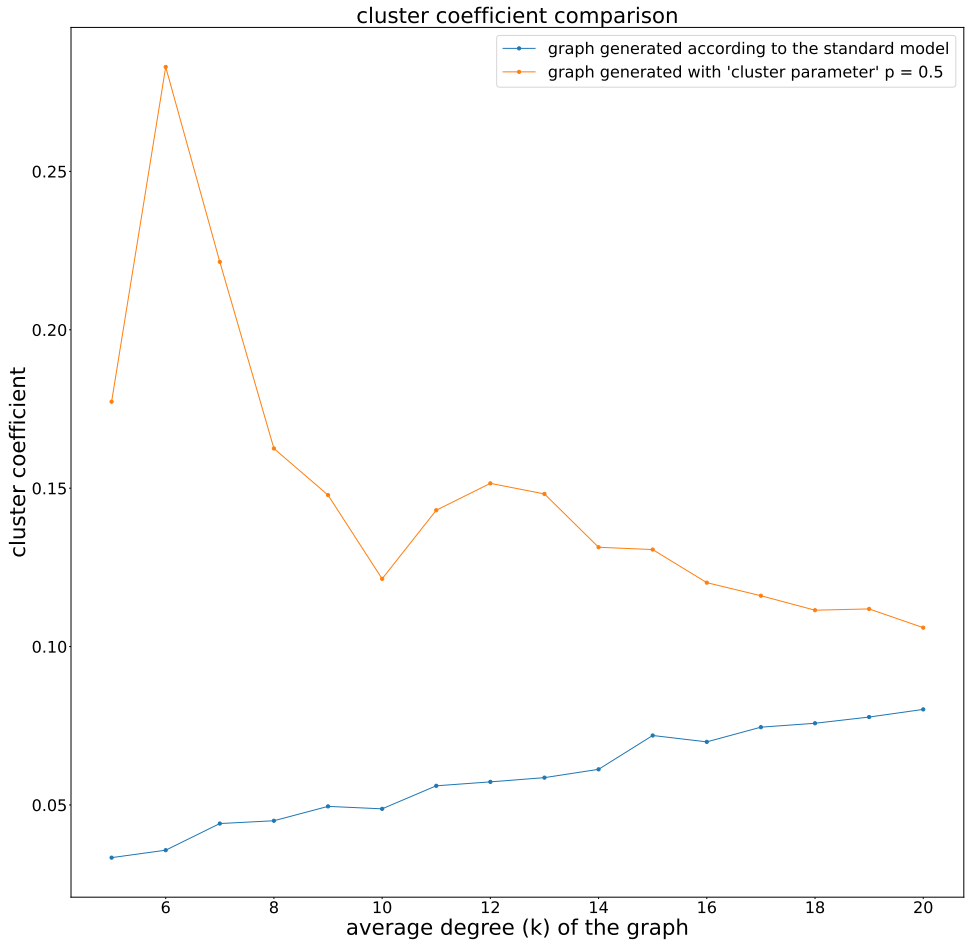

Figure 13: cluster coefficient comparison

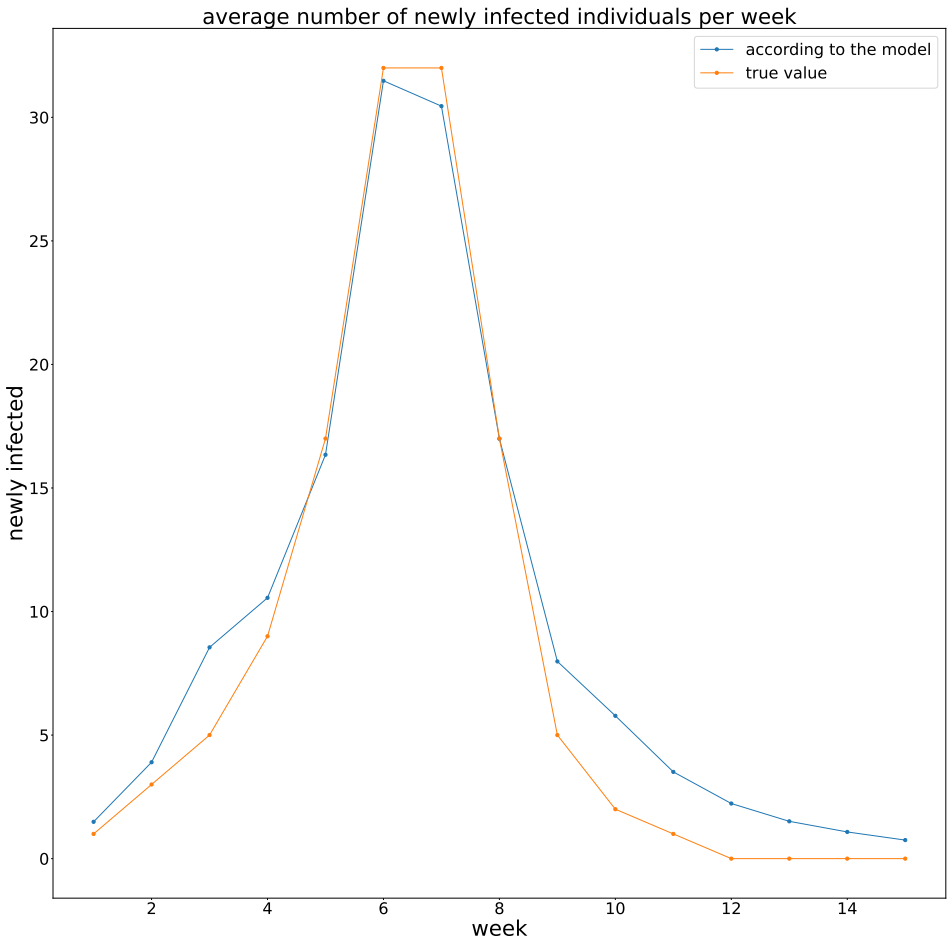

Figure 14: average number of newly infected individuals per week

- $p = 0.25, RMSE = 4.13350$
- $p = 0.5, RMSE = \mathbf{3.85957}$
- $p = 0.75, RMSE = 4.34858$

1.4.4 Epidemic simulation with parameters found

$N = 100$ simulations have been run on a random graph generated according to the standard preferential attachment model, with parameters (k, β, ρ) such that $k = 15, \beta = 0.1, \rho = 0.6$.

Fig 14 shows the average number of newly infected individuals each week according to the model compared to the true value of newly infected individuals each week and Fig 15 the average total number of susceptible, infected and recovered individuals at each week according to the model.

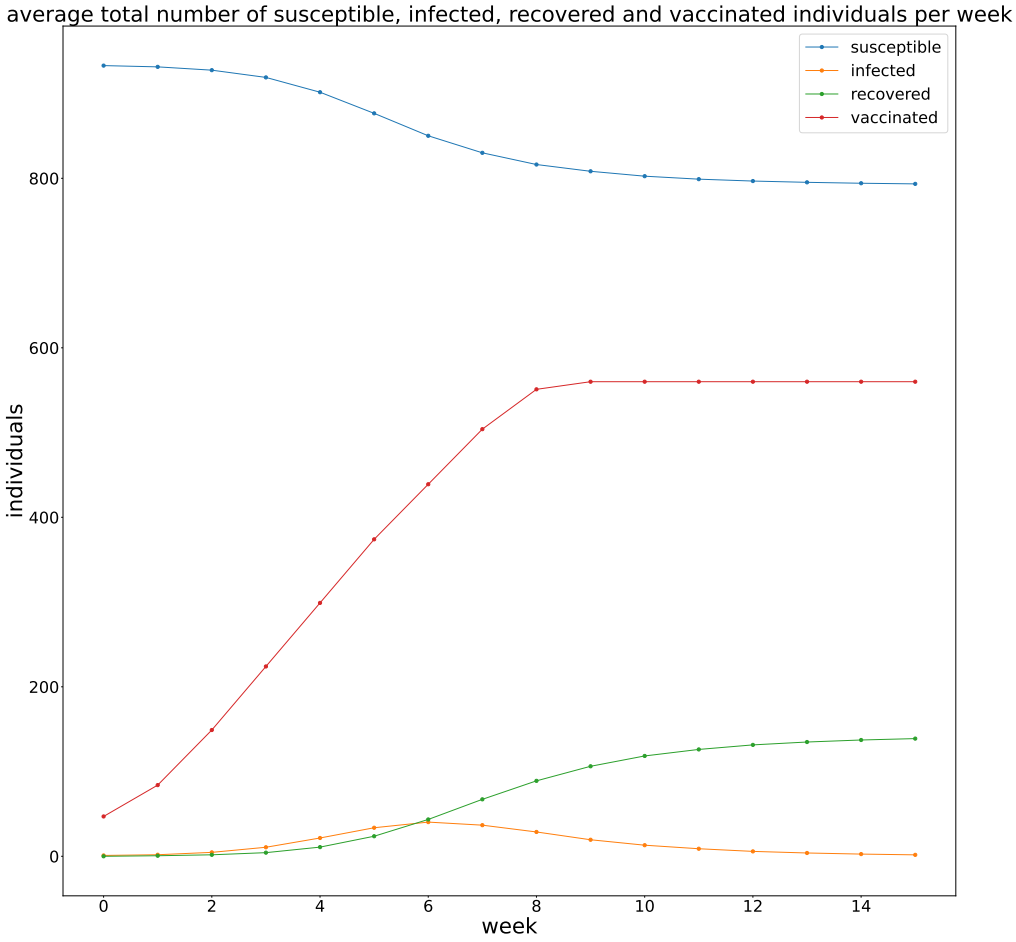

Figure 15: average total number of susceptible, infected, recovered individuals and vaccinated at each week

2 Coloring

In this section it is requested to simulate a learning dynamics in potential games with the aim of studying graph coloring problem to assign WiFi-channels to routers.

The purpose of graph coloring is to assign a color to each node in a given undirected graph, such that none of the neighbors of a node have the same color as that node.

The dynamics is described as follows.

Given an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the state of the node $i \in \mathcal{V}$ is denoted by $X_i(t)$, with state vector $\mathcal{C}$.

Every discrete time instant t, one node $I(t)$, chosen uniformly at random, wakes up and updates its color. The new color (resulting from a node's update), is chosen from a probability distribution given by:

$$P(X_i(t+1) = a \mid X(t), I(t) = i) = \frac{e^{-\eta(t)\sum_j W_{ij} c(a, X_j(t))}}{\sum_{s \in \mathcal{C}} e^{-\eta(t)\sum_j W_{ij} c(s, X_j(t))}}$$

where c represents the cost function and $\eta(t)$ the inverse noise parameter, defined as a function of time t:

$$\eta(t) = \frac{t}{100}$$

The potential function of a configuration at a given time t is defined as:

$$U(t) = \frac{1}{2} \sum_{i,j \in \mathcal{V}} W_{ij} c(X_i(t), X_j(t))$$

2.1 Learning dynamics simulation on a Line Graph

In the following, a line graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with $\mathcal{V} = \{1, 2, ..10\}$, state vector $\mathcal{C} = \{red, green\}$ and an initial configuration $X_i(t) = red, \forall i \in \mathcal{V}$ are considered.

The cost function c is defined as:

$$c(s, X_j(t)) = \begin{cases} 1, & \text{if } X_j(t) = s \\ 0, & \text{otherwise} \end{cases}$$

Fig 16 shows the potential function as a function of each possible configuration $X \in \mathcal{X} = \mathcal{A}^{\mathcal{V}}$, where $\mathcal{X}$ is the configuration vector. As can be seen from the plot, the minima of the potential function correspond to configurations in which each node is in a state such that none of its neighbors are in the same state, i.e.: this configurations are solutions of the coloring problem. The maxima of the potential function correspond to configurations in each node is in the same state. Configurations close to a solution of the coloring problem correspond to a low value of the potential function.

The learning dynamics has been simulated for 600 time units. Fig 17 shows the potential function as a function of time. As can be seen from the plot, as the time increases, the inverse noise increases (i.e., the noise decreases) and with a higher probability the following configuration is a lower-potential configuration. In the limit of $t \to +\infty$, the noise tends to 0: for high values of t, when the dynamics reach a 0-potential configuration, it very unlikely to leave it.

2.2 Learning dynamics simulation on a Graph representing a network of routers

In the following, a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with $\mathcal{V} = \{0, 1, 2, ..99\}$, state vector $\mathcal{C} = \{red, green, blue, yellow, magenta, cyan, olive, black\}$[7] and an initial configuration $X_i(t) = red, \forall i \in \mathcal{V}$ are considered.

In the graph $\mathcal{G}$, nodes represent routers, a link between two nodes means that the two routers are able to interfere with each other and colors (i.e., states) represent frequency bands.

The cost function c is defined as:

[7] With respect to the text of the assignment, the color *white* has been changed to *olive*, due to visualization purposes

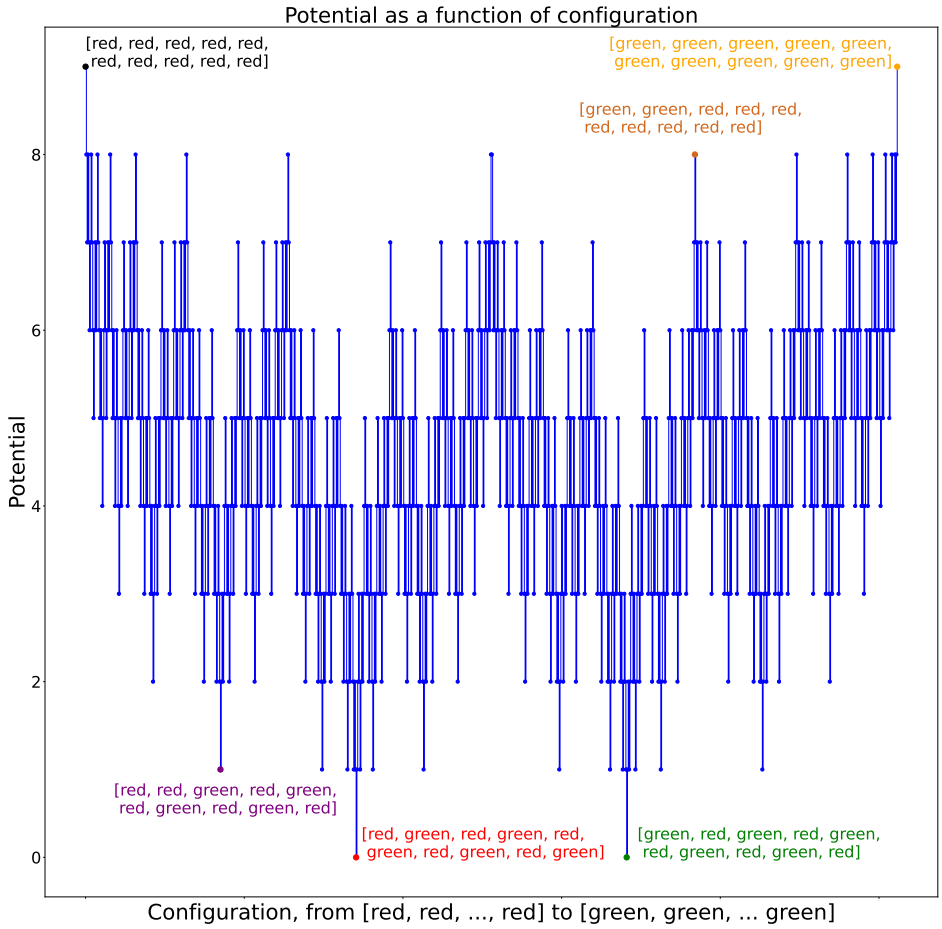

Figure 16: potential as a function of configurations. Configurations are represented on axis x, from [red,..., red] to [green, ..., green], complementing a color each step, starting from the last element of the vector

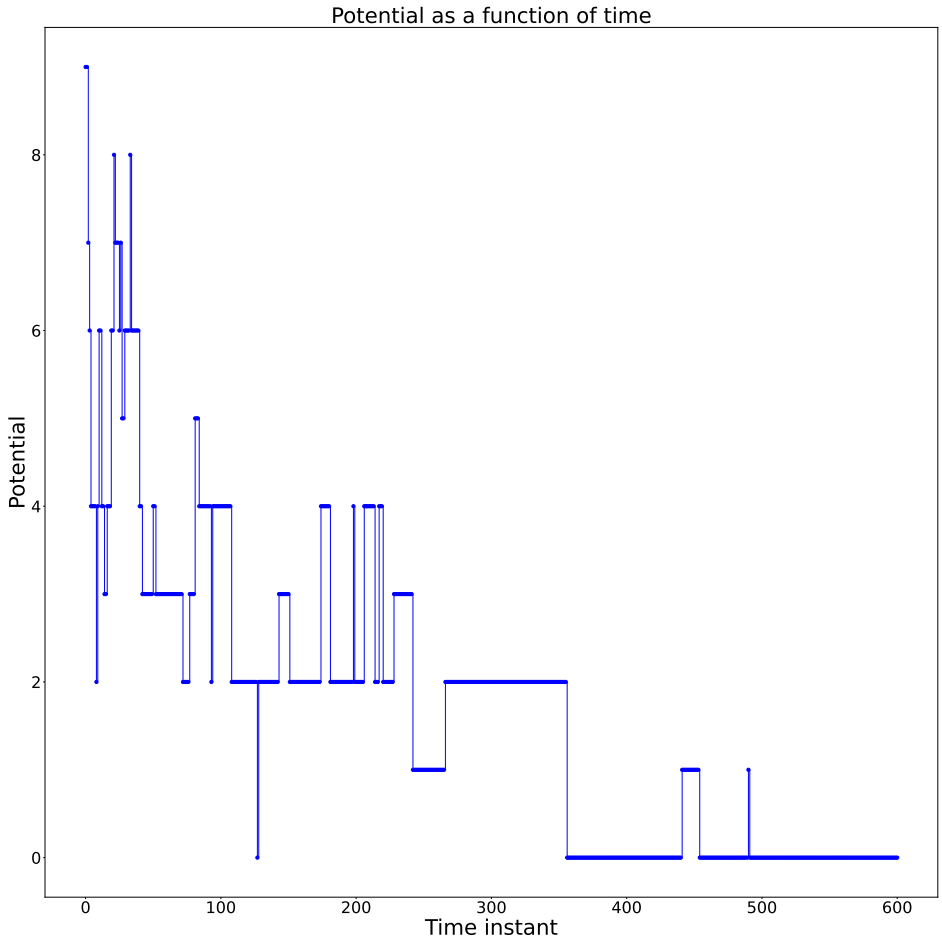

Figure 17: potential as a function of time instant

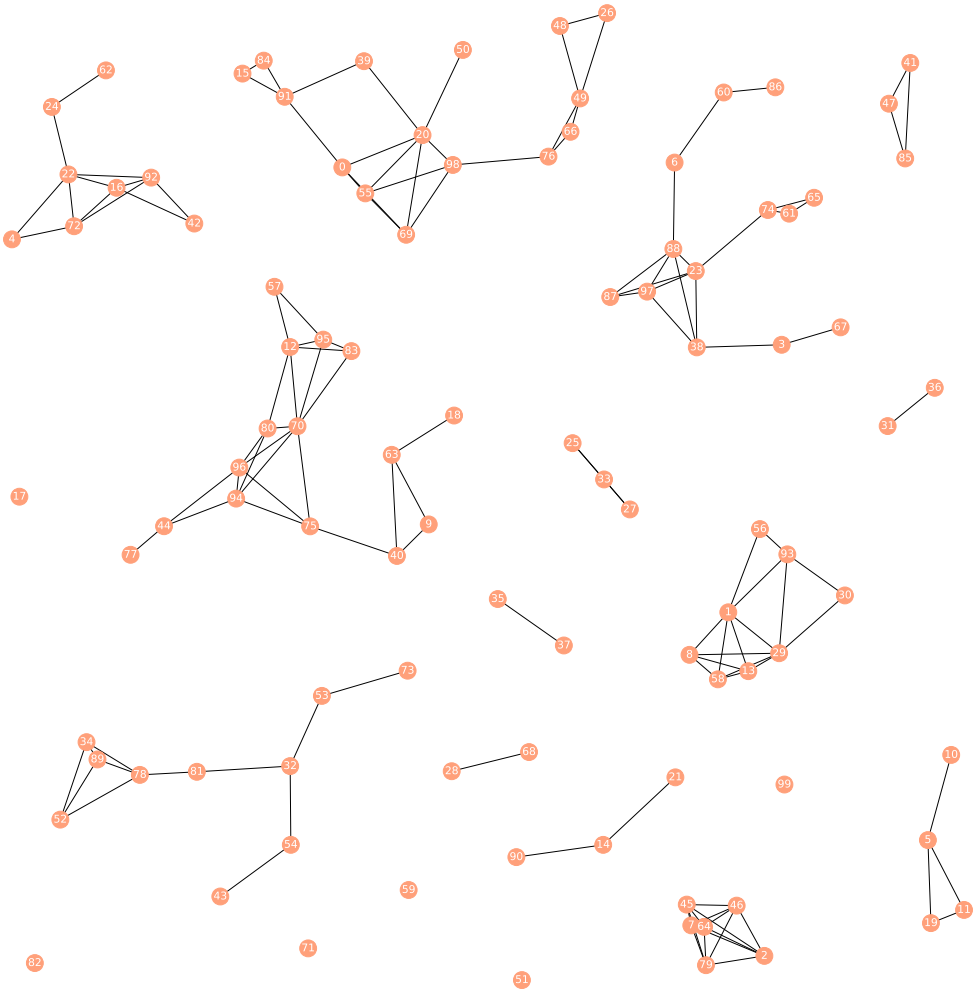

Figure 18: network of 100 routers

$$c(s, X_j(t)) = \begin{cases} 2, & \text{if } X_j(t) = s \\ 1, & \text{if } |X_j(t) - s| = 1 \\ 0, & \text{otherwise} \end{cases}$$

which symbolizes that routers that are close by should not use channels with the same frequency band or a frequency band right next to each other.

The graph $\mathcal{G}$ is shown in Fig 18.

The learning dynamics has been simulated until the termination condition is satisfied. The latter is based on the potential of a configuration and it has been implemented in 2 different ways:

- the simulation ends when potential reaches *value* value
- the simulation ends when potential does not change more than *value* for *iterations* consecutive iterations

With respects to the first condition, different simulations with different *value* $\in \{6, 5, 4, 3\}$ have been run. The lower-potential configuration found has a potential equals to 4. In the attached code

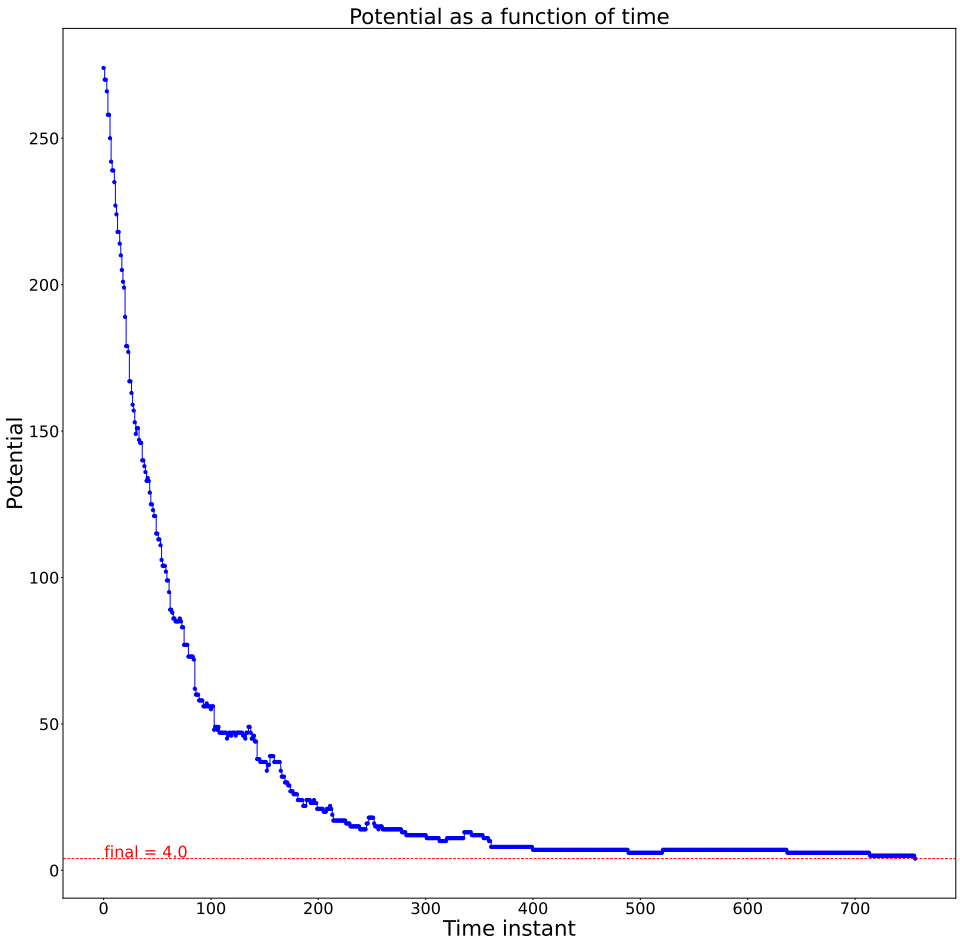

Figure 19: potential as a function of time instant

plots of potential as a function of time for $value \in \{6, 5\}$ with the corresponding final configuration are reported, here it is reported the plot of potential as a function of time (Fig 19) for $value = 4$ with its corresponding final configuration (Fig 20). As can be seen from the plot, as the time increases, the inverse noise increases (i.e., the noise decreases) and with a higher probability the following configuration is a lower-potential configuration. In the limit of $t \to +\infty$, the noise tends to 0: for high values of t, when the dynamics reach the 4-potential configuration, it very unlikely to leave it.

With respects to the second condition, different simulations with different $value \in \{0, 1, 2\}$ and $iterations \in \{50, 100\}$ have been run. The lower-potential configuration found has a potential equals to 4. In the attached code plots of potential as a function of time for $value \in \{0, 1, 2\}$ and $iterations \in \{50, 100\}$ are reported, here it is reported the plot of potential as a function of time (Fig 21) for $value = 0$ and $iterations = 100$ with its corresponding final configuration (Fig 22)

2.2.1 Test of different initial configurations

To evaluate how the initial configuration influences the dynamics, 10 different simulations, each with a different initial configuration randomly generated, have been run. Each simulation ends at the

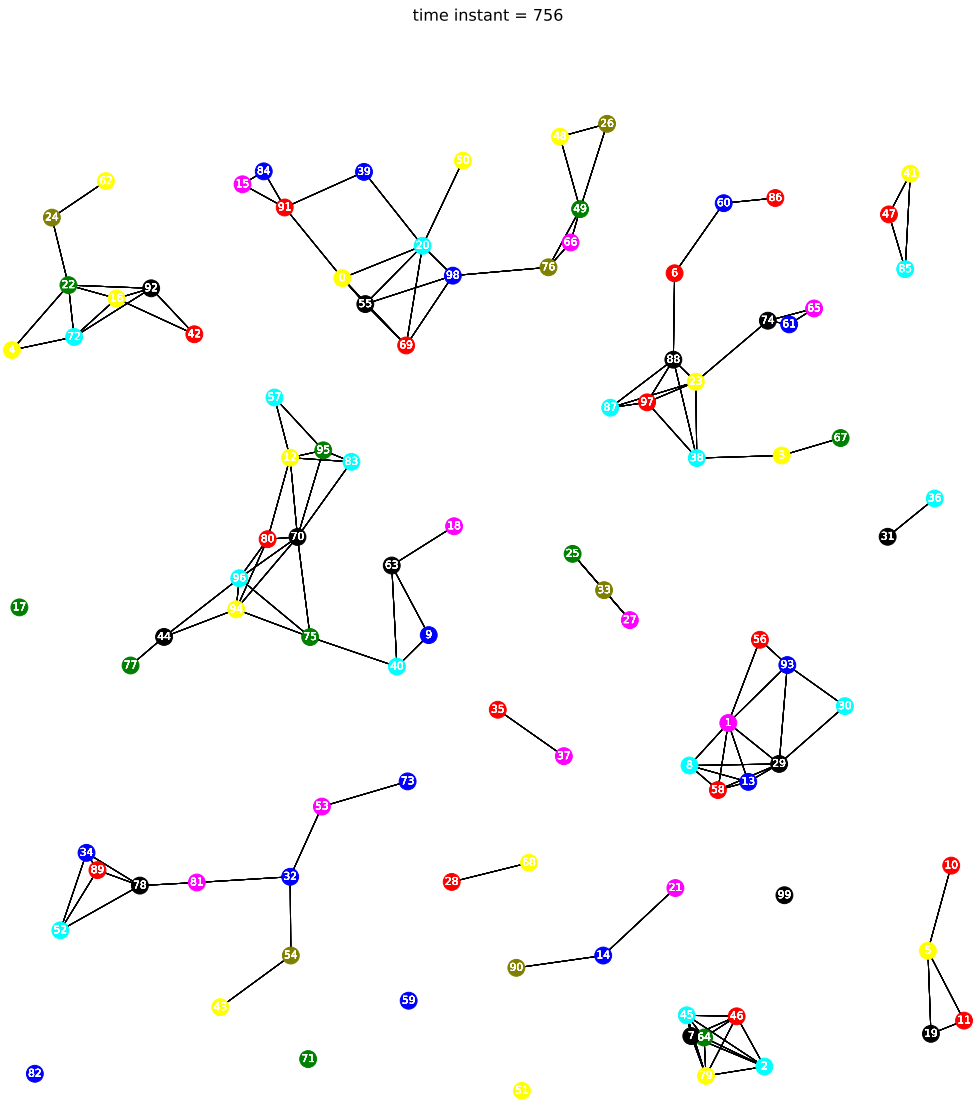

Figure 20: final configuration of the network

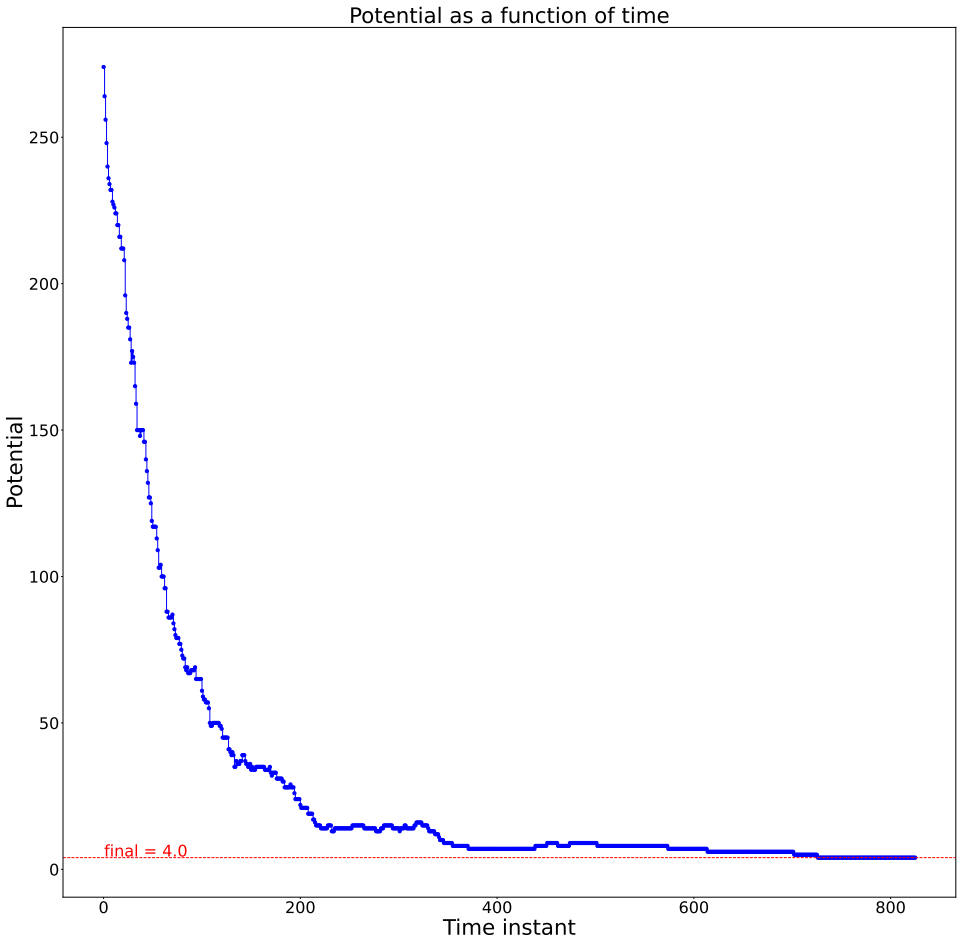

Figure 21: potential as a function of time instant

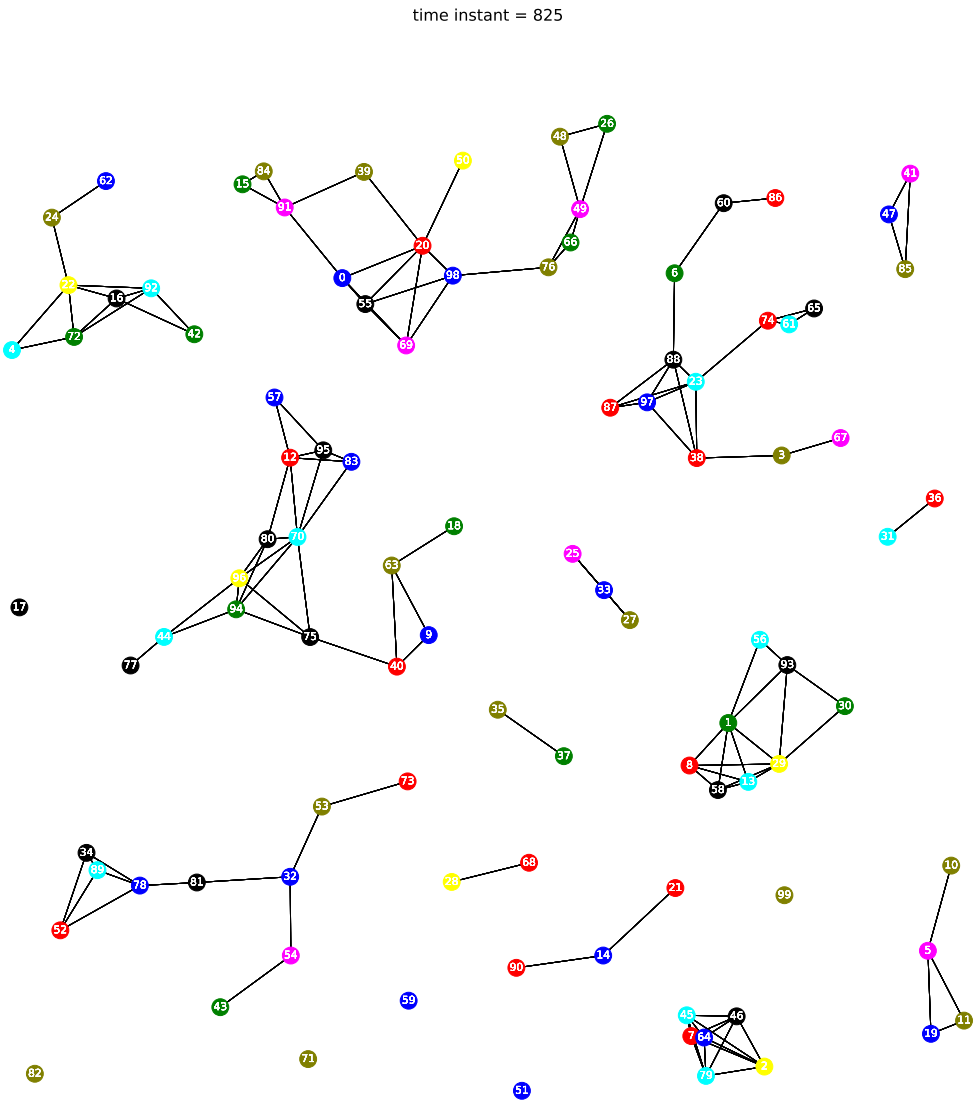

Figure 22: final configuration of the network

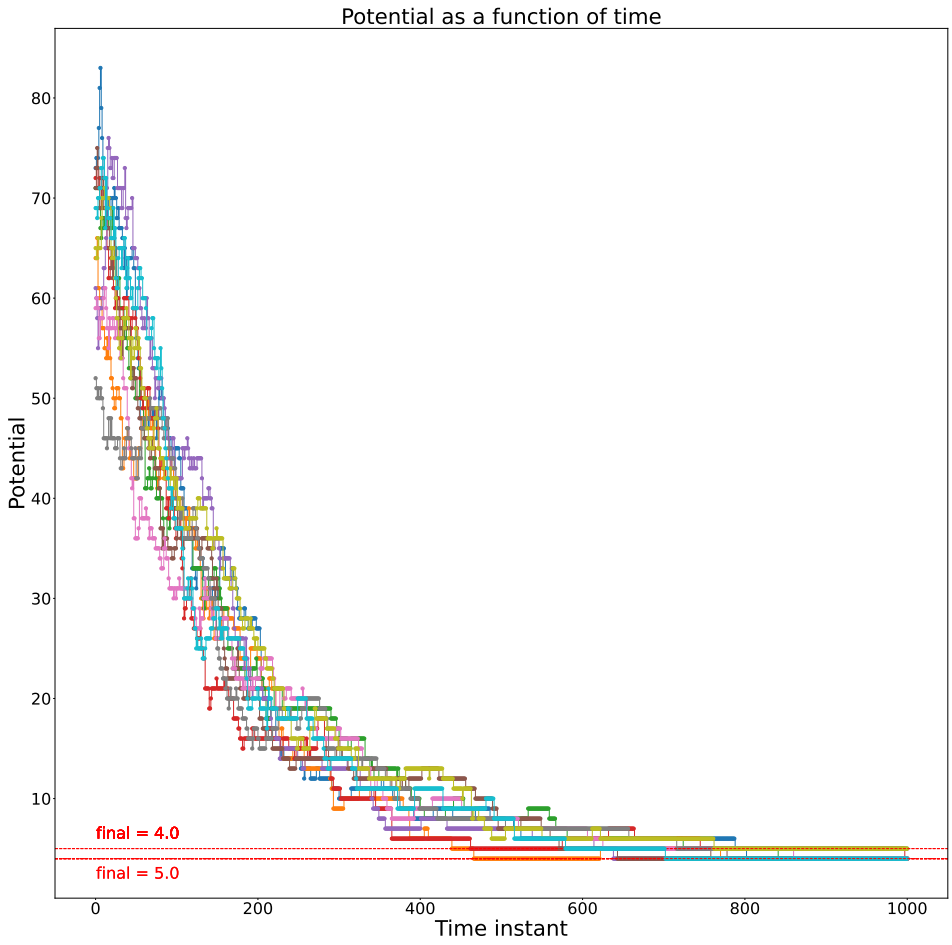

Figure 23: potential as a function of time instant for different initial configurations, each labeled with a different color

1000^{th} time unit. Fig 23 shows the plot of potential as a function of time for each different initial configurations (the final configurations are reported in the attached code). As can be seen from the plot, the choice of the initial configuration does not influence significantly the dynamics, since the potentials of the different simulations show a similar behaviour and, in the final configuration, the 90% of the simulations reaches a 4-potential configuration (the only only simulation which does not reach it reaches a 5-potential configuration).

2.2.2 Study of inverse noise parameter

To evaluate how the inverse noise parameter influences the dynamics, different inverse noise have been evaluated:

- constant inverse noise
  - small values
    - $\eta = 0.1$
    - $\eta = 0.01$

        * $\eta = 0.001$
        * $\eta = 0.0001$
        * $\eta \to 0^+$
    - $\eta = 1$
    - $\eta = 2$
    - $\eta = 5$
    - large values
        * $\eta = 10$
        * $\eta = 100$
        * $\eta \to +\infty$

- inverse noise as an (increasing) function of time
    - $\eta(t) = t/100$
    - $\eta(t) = t^2/5000$
    - $\eta(t) = \sqrt{t}/100$

For each choice of η, a simulation ending at the 1000^{th} time unit has been run. Each simulation starts from the same initial configuration, randomly chosen.

Fig 24, 25 and 26 show the plot of potential as a function of time for each different inverse noise parameter.

As can be seen from the plot, the choice of the inverse noise parameter influence significantly the dynamics.

With a constant high value for the inverse noise (i.e., small noise) the dynamics converges really fast to the lowest potential configuration. In addition, with a high probability, for each time instant, the following configuration is a lower-potential configuration (or equal-potential configuration).

The speed of convergence to the lowest potential configuration and the probability, for each time instant, that the following configuration is a lower-potential configuration (or equal-potential configuration) generally decrease as the inverse noise decreases.

With a constant small value for the inverse noise (i.e., high noise) the dynamics does not converge to the lowest potential configuration. In addition, with a high probability, for each time instant, the following configuration is *not* a lower-potential configuration (or equal-potential configuration).

With an increasing function of time for the inverse noise (i.e., decreasing noise), the dynamics generally converges to the lowest potential configuration. The speed of convergence to such configuration is generally related to the way the function increases. The simulation performed with a quadratic inverse noise function is the fastest to converge while the square root inverse noise function is the lowest (after 1000 time units, it does not reach such configuration). Since at the beginning of the simulation the inverse noise is not so high, for the first time instants the following configuration is a lower potential configuration with a lower probability with respect to the constant high inverse noise case. This can be clearly seen by evaluating the average potential and it is much more evident if the square root inverse noise function is considered.

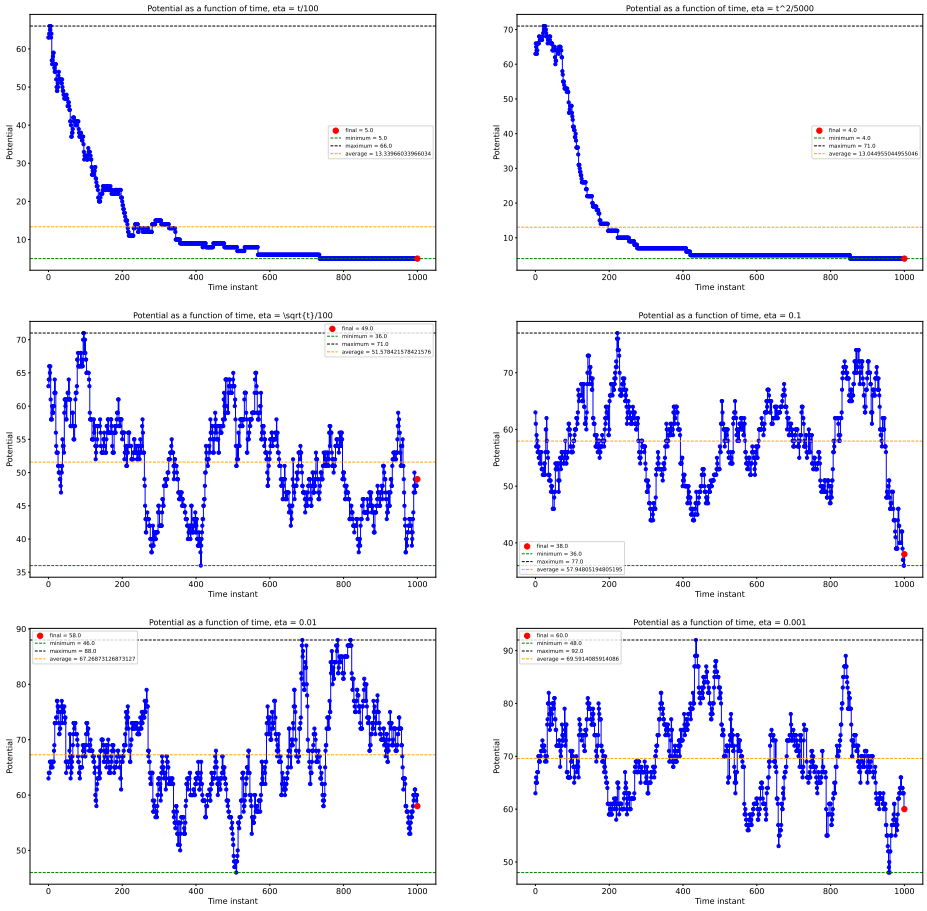

Figure 24: potential as a function of time instant for each different inverse noise parameter

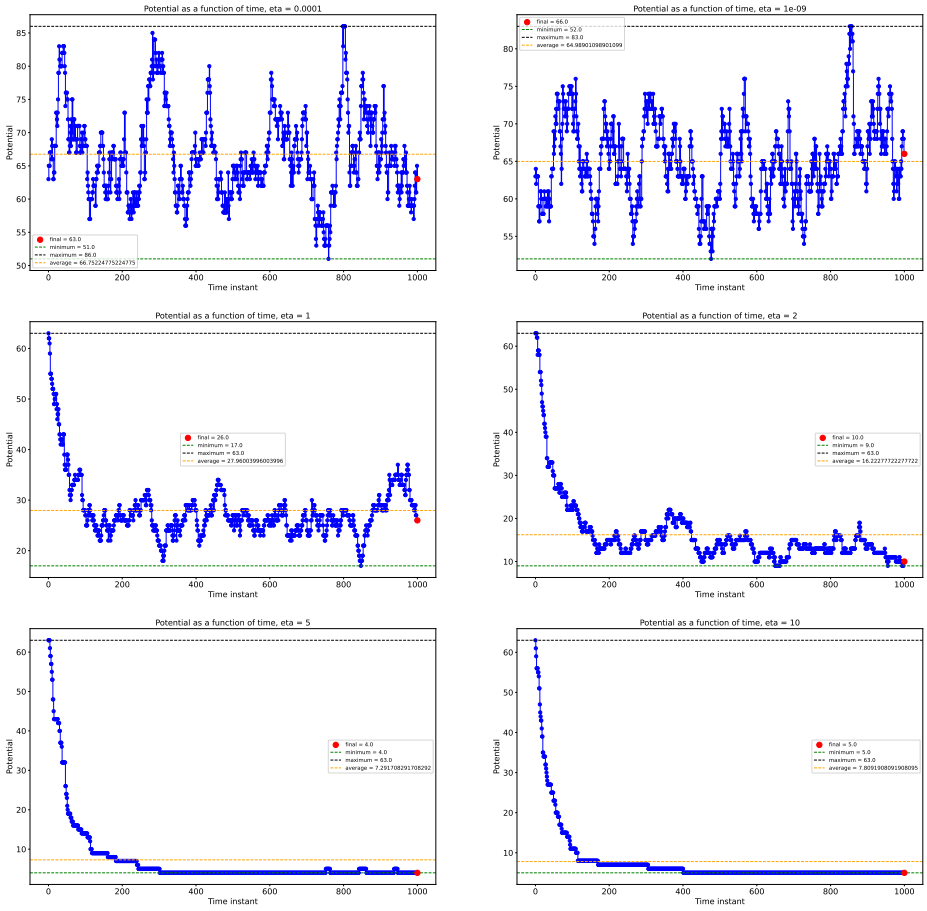

Figure 25: potential as a function of time instant for each different inverse noise parameter

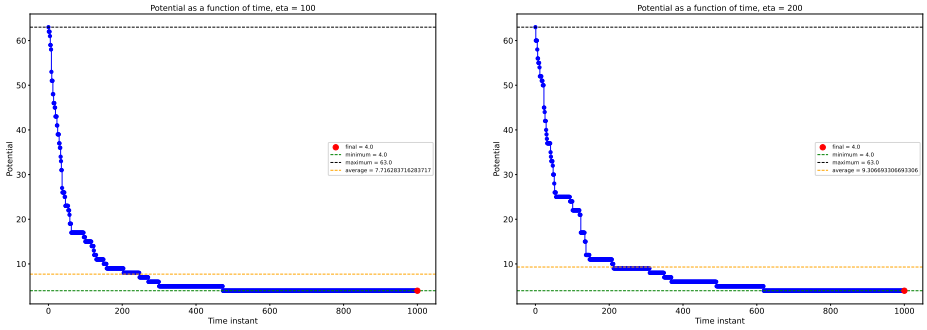

Figure 26: potential as a function of time instant for each different inverse noise parameter