

Network Dynamics and Learning

Homework III

Davide Gariglio
s292964@studenti.polito.it

1 Introduction

This is the report of the last homework of the *Network Dynamics and Learning* course; it will cover *Epidemics*, *Random graphs* and *Graph coloring* topics. All the exercise have been solved in *python* language using the *NetworkX library*; the solutions of the exercises are provided with commented python notebook and explained in the following report.

2 H1N1 2009 Pandemic in Sweden

The goal of this exercise is, by means of intermediate steps, to be able to build a model able to describe the H1N1 Pandemic that occurred in Sweden in 2009. First, we are required to start with some preliminary steps.

2.1 Preliminary steps

The first preliminary step consists in simulating an epidemic on a *symmetric k -regular graph* and then construct a random graph according to *preferential attachment model*.

A *symmetric k -regular graph* is a network with a node set $\mathcal{V} = \{1, \dots, n\}$ in which every node is directly connected to the k nodes whose index is closest to their own modulo n . In Fig. 1 it is visible an example of this graph. in Section 2.1.1 we performed the simulation asked in the homework assignment and briefly described the epidemic model that is simulated.

A random graph constructed with the *preferential attachment model* instead, is a randomly generated network with average degree close to k . We start from $t = 1$ and an initial graph \mathcal{G}_t is initialized (we call it \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 connect it to some

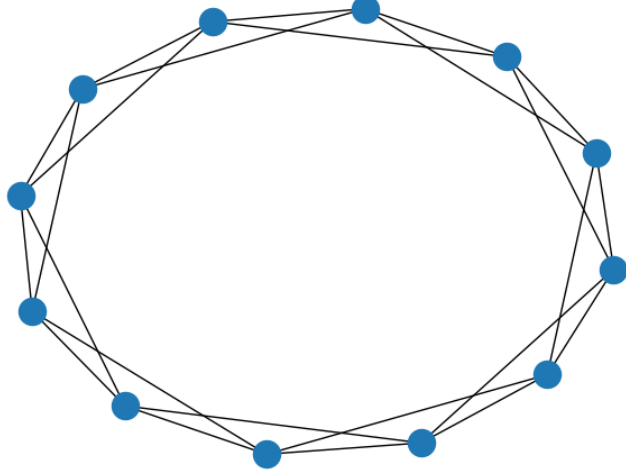


Figure 1: Example of a symmetric k -regular graph with 12 nodes and $k = 4$.

of the existing nodes \mathcal{V}_{t-1} of \mathcal{G}_{t-1} chosen according to some stochastic rule, in this case *preferential attachment*. So, at every time-step $t \geq 2$, every new node added at time t will have a degree $w_t(t) = c = k/2$. Hence, it should add c undirected links to the existing graph \mathcal{G}_{t-1} .

The new connection of the nodes in \mathcal{V}_{t-1} is chosen accordingly to some probability; the one used in this assignment is described in Eq. 1, where $W(t)$ is the adjacency matrix for the next step t and $\omega_i(t-1)$ is the degree of node i before adding the new node. In Section 2.1.2 it is provided an example of a random graph generated according to the *preferential attachment model*.

2.1.1 Epidemic on a known graph

As a first step, it is required to simulate an epidemic on a *symmetric k -regular undirected unweighted* graph \mathcal{G} with $|V| = 500$ nodes and $k = 4$ for 15 weeks (or time units). The simulation we perform is a discrete-time simplified version of the SIR model in which there are pairwise interactions.

First of all we provide a brief overview of the problem and its main parameter.

Each node is called *agent* and can assume three states described by the set $\mathcal{A} = \{0, 1, 2\}$ which corresponds respectively to:

- **S** (susceptible)

- **I** (infected)
- **R** (recovered)

It is possible to interpret the problem as a discrete-time Markov chain $X(t)$ on the space $\chi = \mathcal{A}^V$ defined by a transition probability matrix P . The main probabilities involved are:

- The probability $\beta \in [0, 1]$ that the infection is spread from an infected individual **I** to a susceptible one **S**, given that they are connected by a link, during one time step.
- The probability $\rho \in [0, 1]$ that an infected individual **I** will recover, so its state changes to **R**, during one time step.

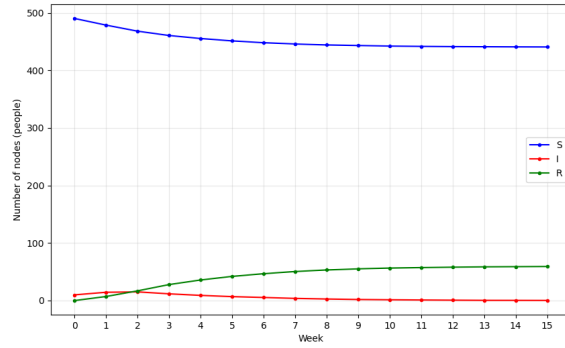


Figure 2: Graph reporting the average total number of susceptible (S), infected (I) and recovered (R) individuals during the weeks over a k -regular graph with 500 nodes, $k = 4$ and 10 infected agents randomly initialized.

From these values it is possible to define the transition probabilities that are reported in the homework assignment.

We are required to run a SIR model 100 times for 15 weeks, with parameters $\beta = 0.3$ and $\rho = 0.7$. In every simulation, 10 different nodes are randomly initialized (according to a uniform distribution) as infected (I), while all the other nodes are labeled as susceptible (S).

In Fig. 3 is reported the average number of newly infected individuals each week resulting from the experiment conducted, while in Fig. 2 it is shown the average total number of susceptible (S), infected (I) and recovered (R) in each week.

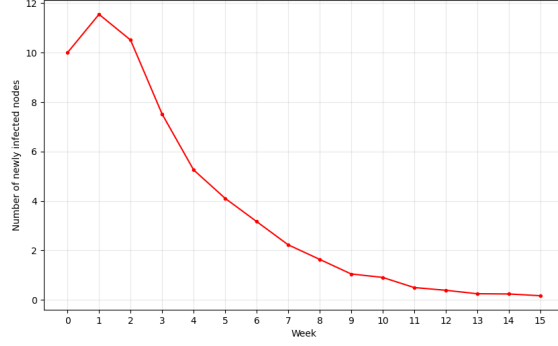


Figure 3: Graph reporting the average number of newly infected individuals during the weeks over a k -regular graph with 500 nodes, $k = 4$ and 10 infected agents randomly initialized.

We can see that in the first weeks the 10 infected agents propagated the infection, resulting in a high number of new infected people in the first 2 time periods (Fig. 3). After that period of time, the epidemic slows down quickly and the majority of people started their recovering process, visible in Fig 2 in which S, I and R curves stabilize during the time; in particular, after the fifth week the curves reach a plateau due to the absorbing configuration achieved. This is the main reason why the epidemic did not propagate so much, but the low value of β and the high value of ρ influenced the results too. Having $\beta > 0.3$ and $\rho < 0.7$ would have resulted in much more infected agents, however the trend of the curves would have been the same due to the graph structure and the absorbing configuration discussed before.

2.1.2 Generate a random graph

As we said in Section 2.1, we are required to construct a random graph with *preferential attachment method*. The stochastic rule used is reported in Eq. 1.

$$\mathbb{P}(W_{n_t,i}(t) = W_{i,n_t}(t) = 1 \mid \mathcal{G}_{t-1} = (\mathcal{V}_{t-1}, \mathcal{E}_{t-1})) = \frac{w_i(t-1)}{\sum_{j \in \mathcal{V}_{t-1}} w_j(t-1)}, \quad i \in \mathcal{V}_{t-1} \quad (1)$$

In Fig. 4 we provide a small example of a random graph generated with this technique. However, in this preliminary step we are required to create large size graph (at least 900 nodes) of this kind starting from a complete

graph with $|\mathcal{V}_1| = k_0 = k + 1$. In this report and in the attached notebook the visualization of the graph is not provided because the number of nodes and links do not allow a proper view.

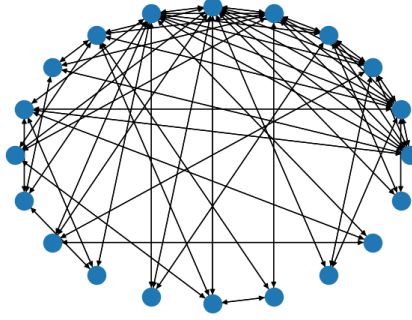


Figure 4: Example of a random graph built with *preferential attachment model* characterized by 20 nodes and average degree $k = 6$.

2.2 Simulate a pandemic without vaccination

After the previous preliminary step, the first task consists in performing a discrete-time SIR model simulation using a random graph based on *preferential attachment*. In particular we performed 100 simulations with a graph generated as explained in Section 2.1.2 with the following parameters:

- average degree $k = 6$
- number of nodes $|\mathcal{V}| = 500$
- $\beta = 0.3$
- $\rho = 0.7$

Every simulation has been carried out for 15 weeks and with 10 initial nodes infected randomly chosen, like in Section 2.1.1.

The results of the simulations are shown in Fig. 6, 5. We can see that the infection rate grows much faster than the epidemic studied in Section 2.1.1 on a k -regular graph and both the infection cases and the newly infections present a more defined bell shape with a peak in the fifth week. Susceptible agents decrease really fast too, with only 30% remaining in the sixth week. The recover process is not so fast and a lot of the agents have been infected; the epidemics has propagated a lot due to the graph configuration, especially

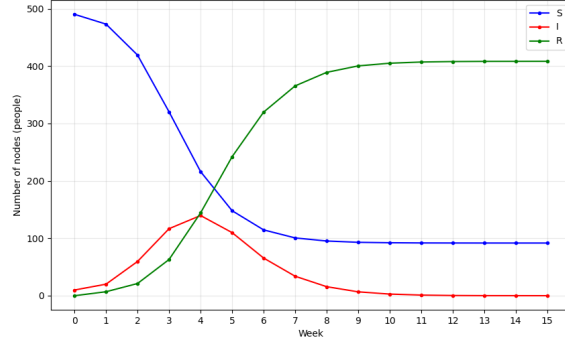


Figure 5: Graph reporting the average total number of susceptible (S), infected (I) and recovered (R) individuals during the weeks over a *random graph* generated with *preferential attachment* criteria with 500 nodes, average degree $k = 6$ and 10 infected agents randomly initialized.

the fact that the average degree k was 6 (which means that there are also infected agents with potentially more than 6 neighbors). Towards the end of the simulation the infection rate is equal to 0, meaning that a stable configuration has been found.

The epidemic simulated is very similar to a real case scenario, however in the real life the infections are countered by vaccines which helps to reduce the number of infected over the weeks, leading to a stable configuration much earlier in the time.

2.3 Simulate a pandemic with vaccination

Similarly to what we did in Section 2.2 we simulate an epidemic over the same type of graph, but introducing vaccines. Therefore, now the states of the agents are four, in particular $\mathcal{A} = \{1, 2, 3, 4\}$; these correspond respectively to:

- **S** (susceptible)
- **I** (infected)
- **R** (recovered)
- **V** (vaccinated)

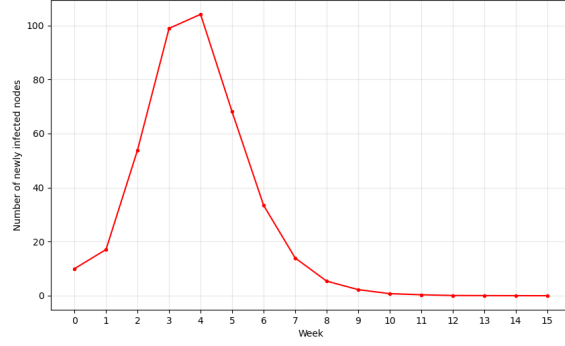


Figure 6: Graph reporting the average number of newly infected individuals during the weeks over a *random graph* generated with *preferential attachment* criteria with 500 nodes, average degree $k = 6$ and 10 infected agents randomly initialized.

We are given the following vector representing the cumulative percentage of the agents that have been vaccinated over the week t (15 in total):

$$\text{Vacc}(t) = [5, 15, 25, 35, 45, 55, 60, 60, 60, 60, 60, 60, 60, 60, 60]$$

In performing this simulation we assume that the vaccination takes effect immediately once given to an agent. The number of individuals to vaccinate per week is retrieved from the $\text{Vacc}(t)$ and those are randomly chosen according to a uniform distribution from the population that has not yet received vaccination. This means that also an infected person can receive the vaccine as well; this reproduces a real case scenario in which an agent can not tell whether she/he has the infection or just the common cold. Moreover we assume that if an infected person get vaccinated, she/he can not infect other individuals.

The simulations are performed on a graph created with the same parameter as the one used in Section 2.2 and the average results are show in Fig. 7, 8.

In Fig. 7 is visible that the number of recovered people R is much lower than the simulation performed without vaccines, meaning that the vaccination campaign had a huge impact over the health of the agents. Another consequence is the number of infected people which is lower. The curve of susceptible people has the same trend as in the previous case (Fig. 5),

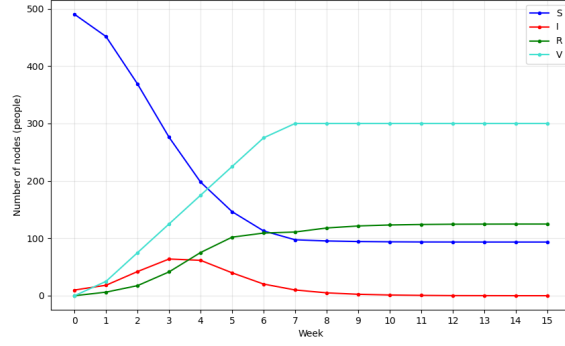


Figure 7: Graph reporting the average total number of susceptible (S), infected (I), recovered (R) and vaccinated (V) individuals during the weeks over a *random graph* generated with *preferential attachment* criteria with 500 nodes, average degree $k = 6$ and 10 infected agents randomly initialized.

however there is a huge advantage; with the vaccines, susceptible people assumed the state V, while in the previous simulation they became infected.

The last consideration is related to Fig. 8, in which we can see the effect of vaccines. In particular there is a peak in the fourth week as before, but the average number of newly infected people is much lower than before.

As a conclusion, thanks to the vaccination campaign, it is impossible to fight the infection in the early stage of the epidemic but the real advantages arrive with time; less people are infected and more people become immune to the virus.

2.4 The H1N1 pandemic in Sweden 2009

In this final part related to the epidemics we are required to estimate the parameters k , β and ρ that fit best the data from **H1N1 pandemic** in Sweden in 2009. The simulation lasts 15 weeks and it is given the vectors of cumulative percentage of vaccinations and newly infected people over the weeks, respectively:

$$\text{Vacc}(t) = [5, 9, 16, 24, 32, 40, 47, 54, 59, 60, 60, 60, 60, 60, 60]$$

$$I_0(t) = [1, 1, 3, 5, 9, 17, 32, 32, 17, 5, 2, 1, 0, 0, 0]$$

The population of Sweden is scaled down to 934 node (for computational purposes). In order to estimate the values k , β and ρ , we run the algorithm

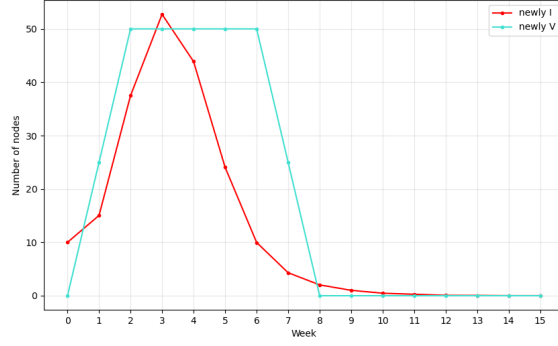


Figure 8: Graph reporting the average number of newly infected individuals and the number of newly vaccinated during the weeks over a *random graph* generated with *preferential attachment* criteria with 500 nodes, average degree $k = 6$ and 10 infected agents randomly initialized.

described in the homework assignment, which performs a grid search of the parameters. For every combination, we measure the distance of our simulation with the real case data with the RMSE (Eq. 2) and take the set of values characterized by the lower error.

$$\text{RMSE} = \sqrt{\frac{1}{n_{\text{weeks}}} \sum_{t=1}^{n_{\text{weeks}}} (I(t) - I_0(t))^2} \quad \text{with } n_{\text{weeks}} = 15 \text{ steps of the simulation} \quad (2)$$

During the search we added an integer value α that multiplies Δ_k , Δ_{β_0} , Δ_{ρ_0} in order to modify the amplitude of the interval and try to avoid a possible *local minima*. The initialization of the parameters is the following:

- $k_0 = 8$, $\Delta_k = 1$
- $\beta_0 = 0.3$, $\Delta_\beta = 0.025$
- $\rho_0 = 0.6$, $\Delta_\rho = 0.025$

At the end of the search we found the optimal parameter set $\{k = 10, \beta = 0.17, \rho = 0.62\}$ with $RMSE = 2.46$. The error is still high, meaning that the parameter found do not suit optimally the real data.

In Fig. 9 we report the final result of the epidemic simulations executed with the best parameters found by the algorithm. We can see that the trends of the curves are similar to what we have obtained in Section 2.3.

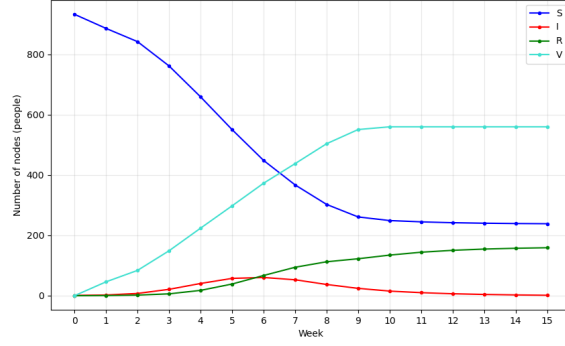


Figure 9: Graph reporting the average total number of susceptible (S), infected (I), recovered (R) and vaccinated (V) individuals during the weeks over a *random graph* generated with *preferential attachment* criteria with 943 nodes and the parameters found by the algorithm ($\{k = 10, \beta = 0.17, \rho = 0.62\}$).

The most interesting result is shown in Fig. 10, in which we can clearly see that the parameters obtained by our algorithm do not suit well the real data. The prediction newly infected agents is "optimistic", with a slow growth, lower peak with respect to the actual one and finally a lower number of newly infected in the last weeks while the actual epidemic was just fading away

These errors could exist due to the variability of the different simulations or maybe because the algorithm stopped in a local minima, even if the number of parameter tested was large.

3 Coloring

The last part of this third assignment is related to *Graph coloring problem*. The aim 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.

3.1 Exercise a

In this task we work on the simple line graph with 10 nodes shown in Fig. 11 and the set of possible states $\mathcal{C} = \{red, green\}$. All the nodes are initialized

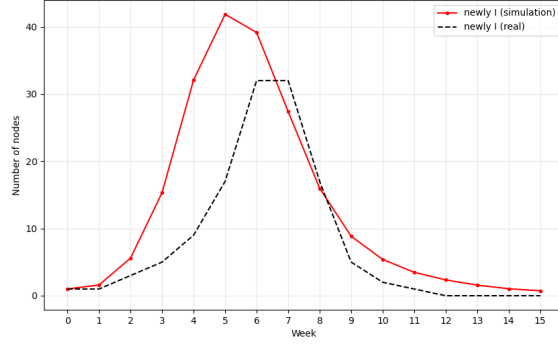


Figure 10: Comparison between the average number of newly infected predicted by the epidemic simulations executed with the best parameters found by the algorithm and the actual infected people from the H1N1 real case in Sweden.

as red, so $X_i(t) = red$ for all $i = 1, \dots, 10$. Every time instant t we randomly choose a node i according to a uniform distribution and update its color.

The new color is chosen according to the following probability distribution

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

and the cost function below

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

where $\eta(t)$ is the inverse of the noise and is chosen according to the following criteria, as a function of the time instant:

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

Finally, in order to study how close the algorithm is to the solution, we use the cost function (the goal is to obtain zero as value, meaning that there are no conflicting nodes):

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

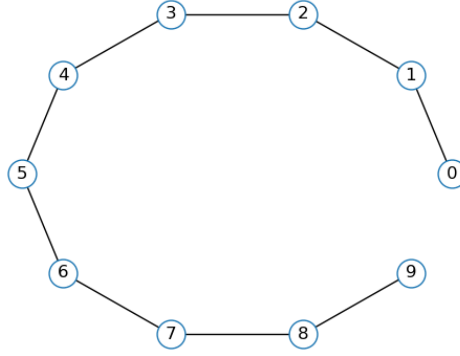


Figure 11: Line graph composed by 10 nodes on which we apply the graph coloring exercise a.

We set a maximum number of iteration equal to 500 (which is quite big for this problem, but the goal is to reach the optimal solution). In Fig. 12 we report the intermediate states of the graph, reaching the solution of the *graph coloring* problem after 140 iterations.

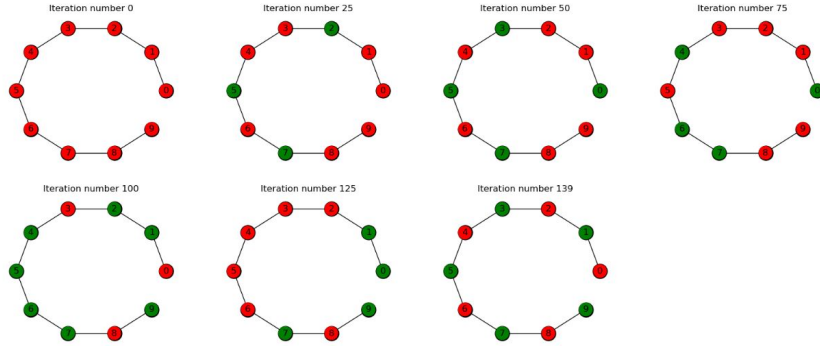


Figure 12: Visualization of the intermediate states of the graph coloring problem over a line graph with two states.

Finally, in Fig. 13 it is visible the potential of the learning process, which stops at the 140th iteration when the final solution is found. In the visualization we can see big jumps of the function; this is because the graph

is composed by only 10 nodes and changing the state of the wrong one will have a significant impact on the potential value.

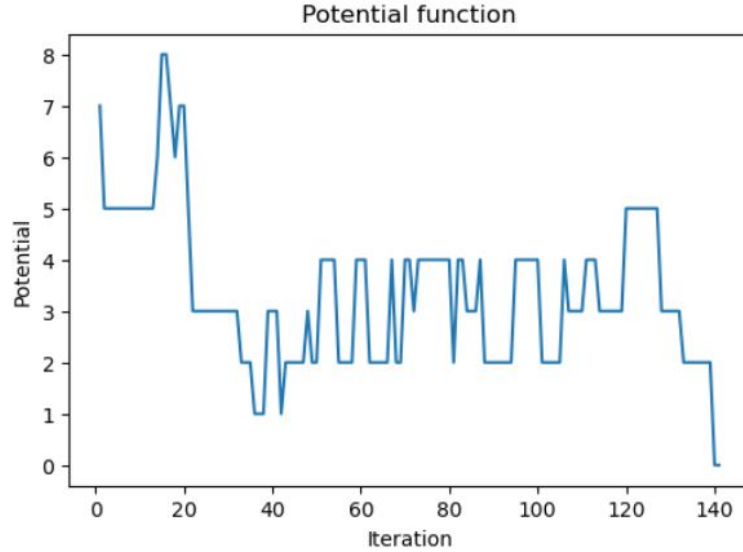


Figure 13: Potential function over the different iterations for the graph coloring problem over a line graph with two states.

3.2 Exercise b

In this last exercise it is required to solve the *graph coloring* problem over a much more complex network shown in Fig. 14 (built with the given matrices `coords.mat` and `wifi.mat`).

The goal is to assign a wifi-channel (color) to the given routers (nodes) because two adjacent nodes could interfere with each other, so every router must be tuned to a different frequency with respect to its neighbors. For this problem we are given 8 different states, represented by $\mathcal{C} = \{f1 : red; 2 : green; 3 : blue; 4 : yellow; 5 : magenta; 6 : cyan; 7 : white; 8 : black\}$.

Moreover it is given the associated cost function

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

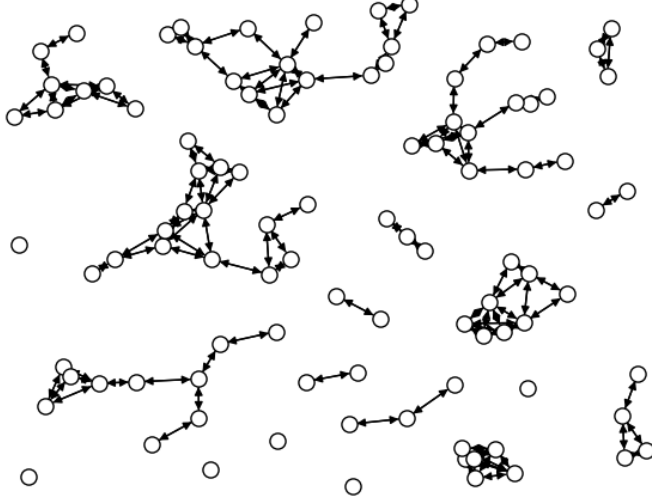


Figure 14: Graph representing the routers network over which the wifi-channels are assigned with the solution of the *coloring graph* problem.

We use the same $\eta(t)$ and the same potential function $U(t)$ as in Section 3.1. We have executed the learning algorithm initializing all the routers states to *red* and studied the behaviour of the algorithm for 3 different number of iterations, in particular we tested with: $t = \{1000, 2500, 5000\}$.

With $t = 1000$ we can observe from Fig. 15 there are many pair of connected nodes with the same color, therefore this value of t is too small. This is confirmed also by Fig. 16 in which is clear that the learning process is not able to find the solution; in fact the final value of the potential is approximately 6.

With $t = 2500$ we can observe from Fig. 17 there are still some connected nodes with the same color, therefore the value of t is reasonable. This is visible also in Fig. 18 in which we can see that the final potential function is comparable with the previous one, however during the learning process it was very low for some iterations. This means that the algorithm was reaching the final solution and then some wrong change of states occurred. This is probably due to a local minima.

With $t = 5000$ we obtained the best possible result; in fact we found the solution of the *graph coloring* problem approximately after 4100 iterations. In fact the potential shown in Fig. 19 reaches the zero value.

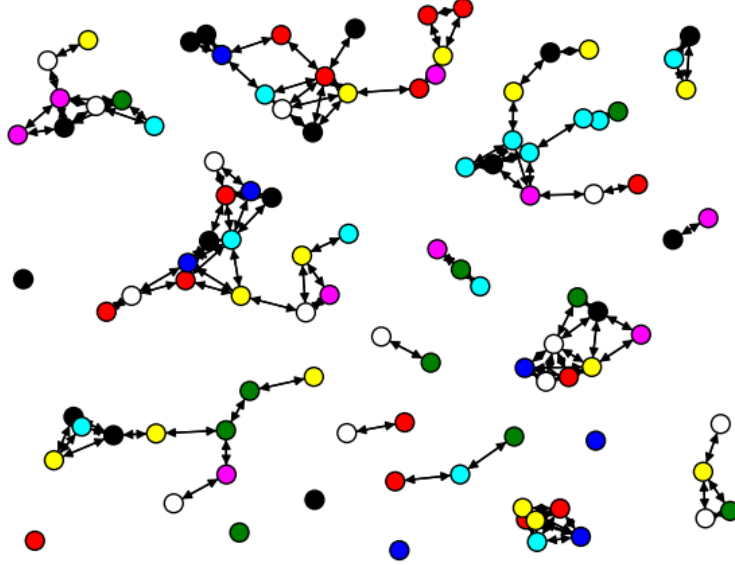


Figure 15: Final node states after 1000 iterations.

As a final consideration it is important to recall that the operations performed are stochastic, therefore the number of iteration in which the algorithm finds the solution may vary, but the trend is the same.

The algorithm is able to reach a possible (local) minima after 200 or 300 iterations; this implies that most of the node states satisfy the constraint of the problem. in a real case scenario with even bigger networks, taking a sub-optimal solution associated with a local minima could be a feasible solution too; it depends on the value of the potential and in particular on the number of routers conflicts.

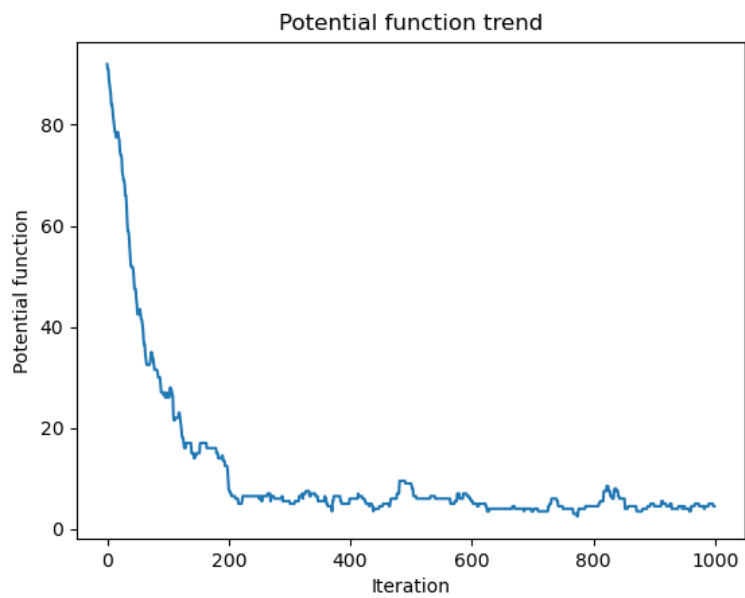


Figure 16: Value of potential over the 1000 iterations

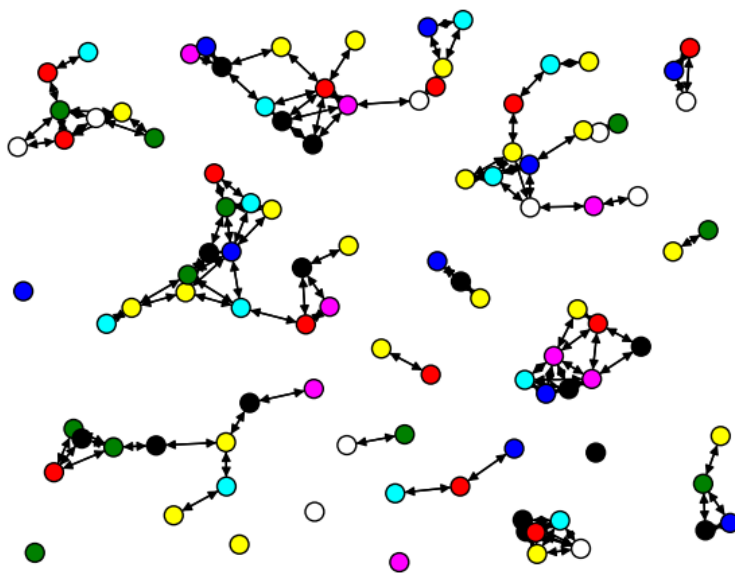


Figure 17: Final node states after 2500 iterations.

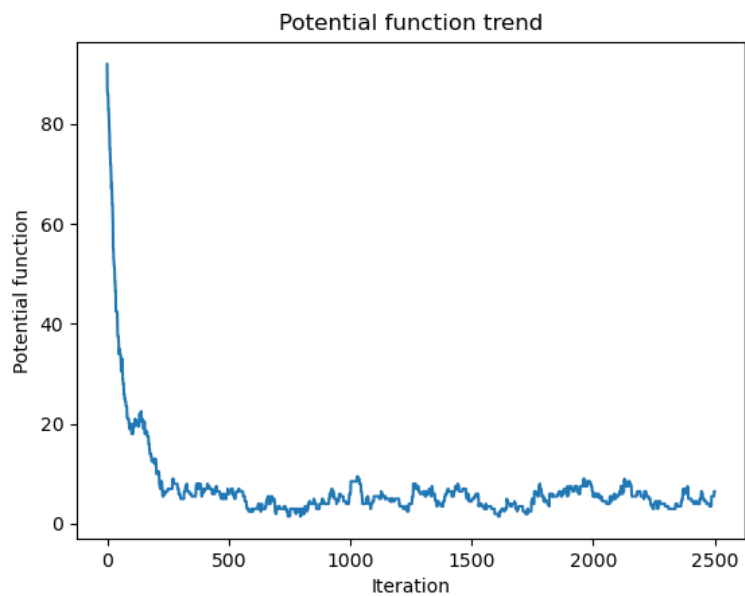


Figure 18: Value of potential over the 2500 iterations.

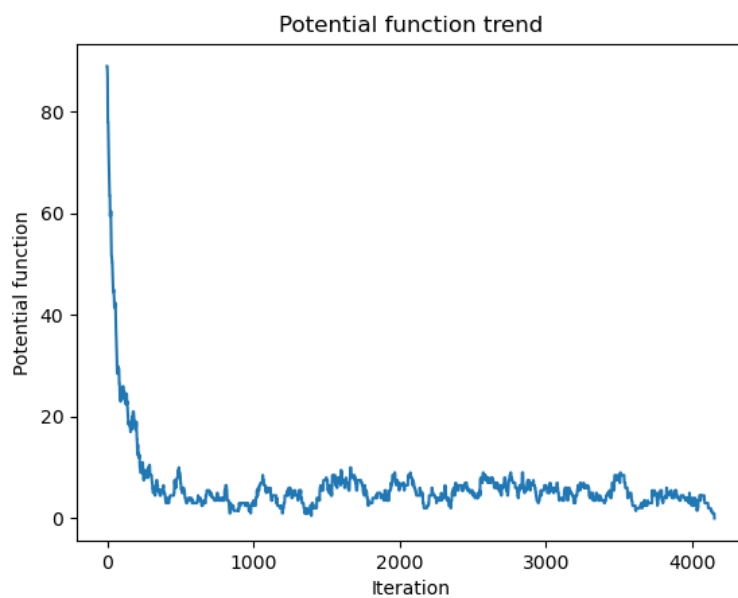


Figure 19: Value of potential over the 5000 iterations.