Networks Dynamics Homework III

Alessia Leclercq
Politecnico di Torino
s291871@studenti.polito.it

I. Exercise

A. Preliminary Parts: Epidemics on a known graph

In this section, a SIR epidemic has been simulated 100 times using a discrete-time Markov Chain. Each simulation consists of 15-time units (weeks in our case) in which each node can change state according to the following:

1. Given a node in state S with m neighbors in state I, call $\beta \in [0,1]$ the probability that the infection is spread from an infected individual to a susceptible one. Then, such an individual gets infected with probability:

$$\mathbb{P}(X_i(t+1) = I \mid X_i(t) = S, \sum_{j \in \mathcal{V}} W_{ij}\delta^I_{X_j(t)} = m) = 1-(1-\beta)^m \tag{1}$$

2. Given an individual in state I, call $\rho \in [0,1]$ the probability that the individual recovers (hence it changes its state to R). Then:

$$\mathbb{P}(X_i(t+1) = R \mid X_i(t) = I) = \rho \tag{2}$$

3. Given an individual in state R, it can not change state anymore

In the simulation, β = 0.3 and ρ = 0.7, and the initial infected group is composed of 10 randomly chosen individuals without replacement.
The graph is an undirected k-regular graph with 500 nodes and k=4.

In Fig.1 one can find the average total number of susceptible, infected, and recovered individuals each week after 100 simulations. Whereas, in Fig.2 one can find the average number of newly infected individuals each week.

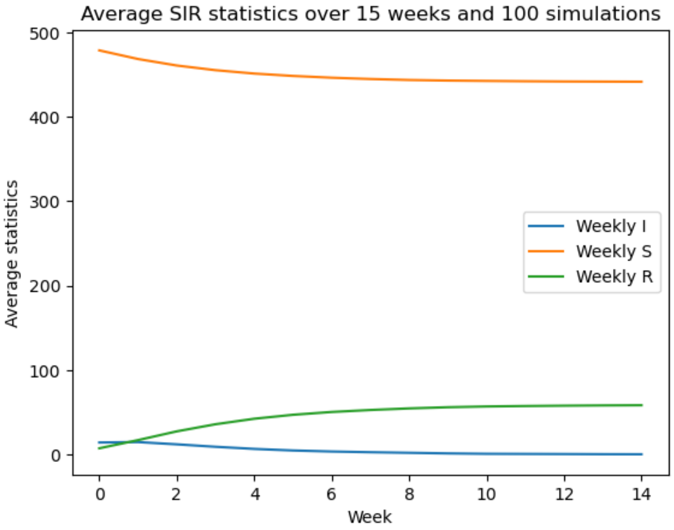

Fig. 1. Average total number of individuals in the SIR compartments

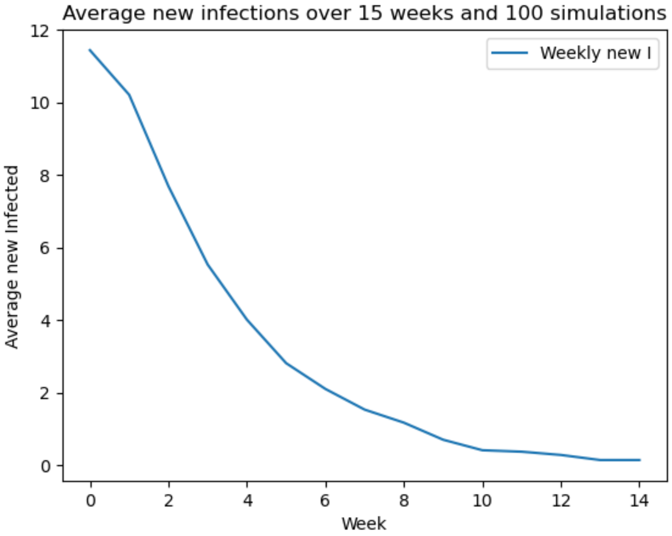

Fig. 2. Average number of newly infected individuals each week

B. Preliminary Parts: Generate a random graph

An algorithm to generate a random graph according to the *preferential attachment model* has been developed. The final goal is to create a random graph in which the average degree of each node is k.
Starting with a complete graph with $k_0 = k+1 = 5$ nodes, at each iteration, a new node is added to the graph and connected using $c = k/2$ undirected edges. Given node $i \in \mathcal{V}_{t-1}$, at time t there will be a link connecting i with the incoming node J with probability:

$$\mathbb{P}(W_{iJ} = W_{Ji} = 1 \mid G_{t-1}) = \frac{w_i(t-1)}{\sum_{j \in \mathcal{V}_{t-1}} w_j(t-1)} \tag{3}$$

where $w_i(t-1)$ is the degree of node i before adding J to the graph. In this preliminary exercise, a total of 900 nodes

TABLE I
EPIDEMICS SIMULATION PARAMETERS

k	$\lvert\mathcal{V}\rvert$	β	ρ	time units
6	500	0.3	0.7	15

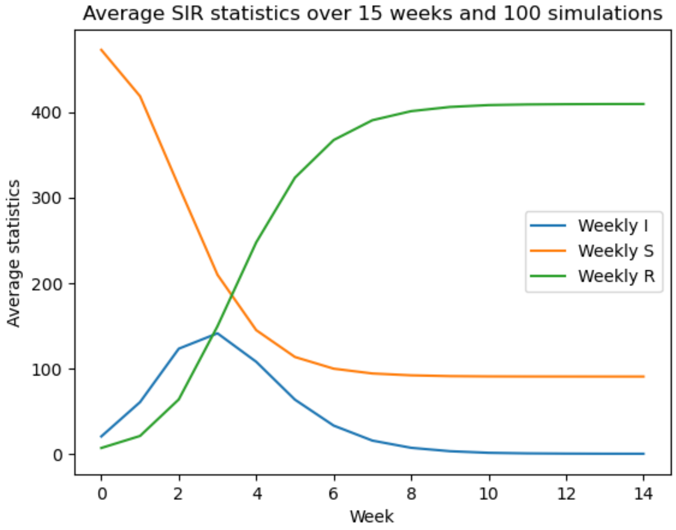

Fig. 3. Average total number of individuals in the SIR compartments

has been reached, and verified that the average degree of each node is k=4.

C. Simulate a pandemic without vaccination

In this exercise, the graph has been generated using the preferential attachment algorithm, and the SIR epidemic has been simulated on such a model for 100 times. In Table I the parameters for the simulation can be found.

The average number of newly infected individuals each week is shown in Fig. 4, whereas the total average number of individuals in the SIR compartments is plotted in Fig. 3.

The following assumptions hold: the population is isolated, so its size is constant in time, there is no incubation time for the disease, and the infection rate is not time-decaying.
The ratio between the recovery probability and the probability of infection is $\frac{\rho}{\beta} = 1.43 > 1$, therefore, we expect the pandemic to disappear. In fact, the number of infected individuals linearly increases in the first weeks, reaching its peak in week 3. Both the infected curve and the curve describing the number of weekly new infected reach 0 around week 10, and the system enters an absorbing configuration $\{S, R\}^{\mathcal{V}}$.

D. Simulate a pandemic with vaccination

The same setting as in the previous exercise is used (see Table I), however, vaccination is introduced. The vaccines can be given to any individual independently from its state, and vaccines are assumed to be able to immunize the individual as soon as the dose is given. For each week the fraction of the population that receives the vaccination is: $Vacc(t)$ = [0, 5, 15, 25, 35, 45, 55, 60, 60, 60, 60, 60, 60, 60]. Also, before the simulation starts any individual in the population is vaccinated and the initial number of infected individuals is 10.
The algorithm at each week first finds the number of people to vaccinate among the whole population according to:

$$|vaccines| = \frac{(Vacc(t) - Vacc(t-1)) * |\mathcal{V}|}{100} \tag{4}$$

And, then, it randomly selects from the non-already vaccinated individuals in the population $|vaccines|$ individuals. On the remaining individuals, the SIR model is simulated as in point A.

In Fig. 6 the average number of newly infected and newly vaccinated individuals each week are plotted, whereas in Fig. 5 the average total numbers of individuals in the SIR and V compartments are shown.
Since we can vaccinate any individual (except those already in V), the vaccination results in the following effects:

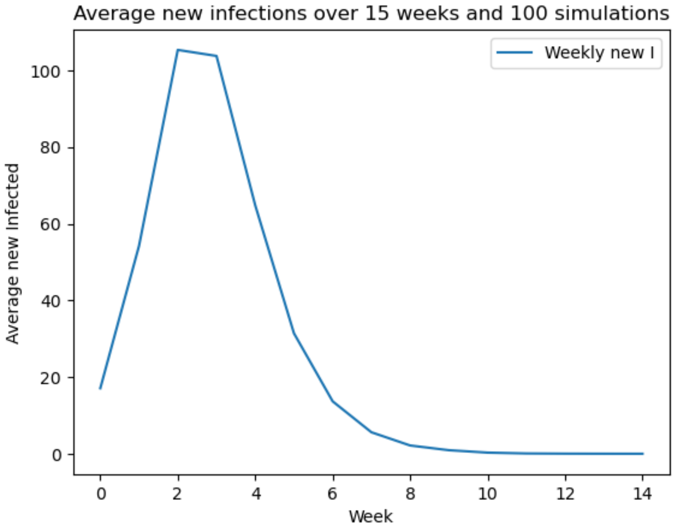

Fig. 4. Average number of newly infected individuals each week

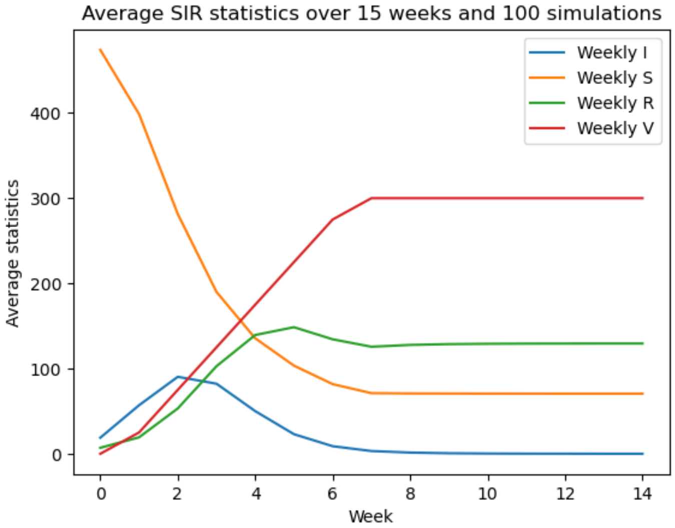

Fig. 5. Average total number of individuals in the SIR compartments and V

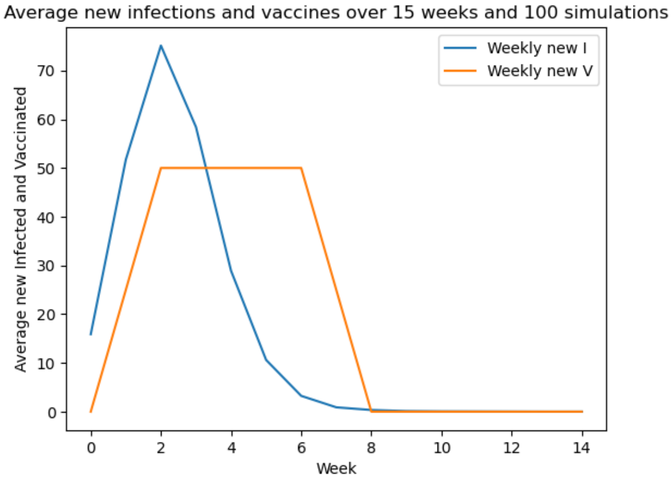

Fig. 6. Average number of newly infected and newly vaccinated individuals each week

- It reduces the number of individuals in I and S, also reducing the risk of transmission of the infection: the peak of I is smaller in size and reached one week earlier than in the same model without vaccination.
- It reduces the number of individuals in R: the size of R is widely reduced with respect to the same model without vaccination, especially after the infection peak.
- It helps in reaching the absorbing configuration faster: with respect to the same model without vaccination $\{S, R\}^{\mathcal{V}}$ is reached two weeks earlier.

E. The H1N1 pandemic in Sweden 2009

In this section, a search algorithm has been developed to approximate the H1N1 Pandemic in Sweden. The algorithm estimates k, β, and ρ using a SIR model with vaccination on a graph using preferential attachment and $|\mathcal{V}| = 934$.
The used search space is $k_{curr} = \{k, k{+}\Delta_k, k{-}\Delta_k\}$, $\beta_{curr} = \{\beta, \beta + \Delta_\beta, \beta - \Delta_\beta\}$, $\rho_{curr} = \{\rho, \rho + \Delta_\rho, \rho - \Delta_\rho\}$.
The root-mean-squared-error is used to compute the difference between the estimated number of newly infected each week $I(t)$ and the ground truth $I_0(t)$:

$$RMSE = \sqrt{\frac{1}{15}\sum_{t=1}^{15}(I(t) - (I_0(t))^2} \tag{5}$$

Where $I_0(t)$ = [1, 1, 3, 5, 9, 17, 32, 32, 17, 5, 2, 1, 0, 0, 0], while $Vacc(t)$ = [5, 9, 16, 24, 32, 40, 47, 54, 59, 60, 60, 60, 60, 60, 60, 60].

The algorithm stops whenever the same set of parameters is found to be the best-performing in two consequent iterations. The algorithm assumes that 5% of the entire population is already vaccinated when the simulation starts and that there is a single infected individual in the population. In Table II the initial and the best-performing parameters are reported, and in Fig. 7 both I_0 and I_{best} are plotted.

TABLE II
H1N1 PARAMETERS

	k	β	ρ	Δ_K	Δ_β	Δ_ρ
initial	10	.3	.6	1	.1	.1
best	9	.2	.6			

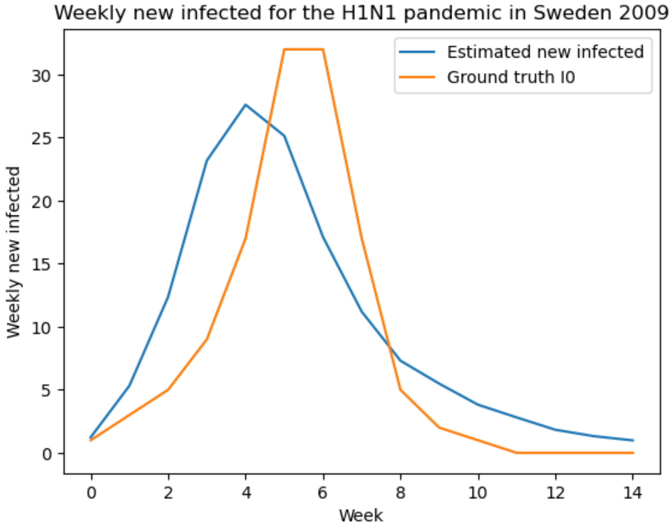

Fig. 7. H1N1 simulation

II. Exercise

The goal of the exercise is to give each node $i \in \mathcal{V}$ a color (i.e., a state) in such a way that any neighbor $j \in N_i(t)$ has the same one. Starting with all nodes having the same state $X_i(t = 0) = red$, at each iteration, a node is randomly chosen with uniform probability and it updates its state with probability:

$$P(X_i(t{+}1) = a|X(t), I(t) = i) = \frac{e^{-\eta(t)\sum_j W_{ij}c(a,X_j(t))}}{\sum_{s\in C} e^{-\eta(t)\sum_j W_{ij}c(s,X_j(t))}} \tag{6}$$

Where $c(s, X_j(t))$ is the cost function.
The algorithm is run until the potential function reaches 0, hence the solution with no conflicting nodes. The potential function is computed according to:

$$U(t) = \frac{1}{2}\sum_{i,j\in\mathcal{V}} W_{ij}c(X_i(t), X_j(t)) \tag{7}$$

A. Coloring on a line graph

In this section, the coloring is simulated on a line graph with $|\mathcal{V}|$=10. There are two possible states, specifically $C = \{red, green\}$. Furthermore, $\eta(t) = t/100$, while the cost function is:

$$c(s, X_j(t)) = \begin{cases} 1 & if\ X_j(t) = s, \\ 0 & otherwise. \end{cases} \tag{8}$$

In Fig. 8 the potential variation over time is plotted. At iteration 196 the potential (Eq. 7) reaches 0 and in Fig. 9 the

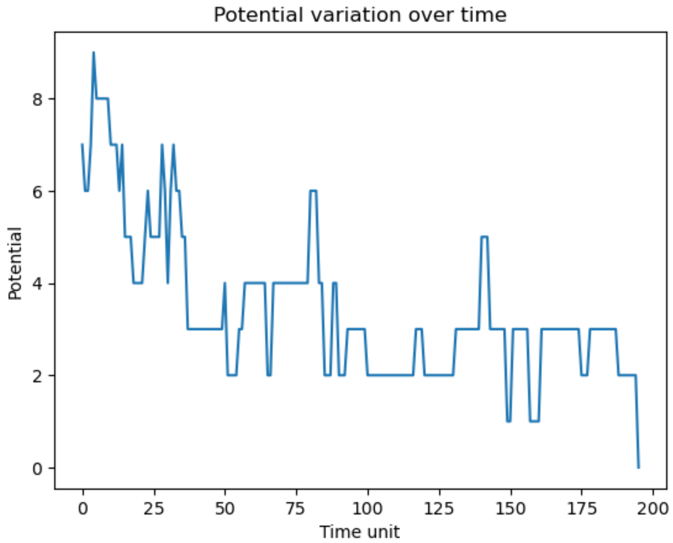

Fig. 8. Potential of the coloring algorithm on the line graph

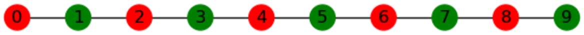

Fig. 9. Final result of the coloring algorithm on the line graph

equilibrium configuration state is shown. The potential reaches zero because the line graph is 2-colorable and also the cost function only penalizes nodes having the same color.

B. Assign WiFi to routers

The same coloring algorithm has been applied in the case of nodes representing routers and edges modeling the possibility of two routers interfering. Here, colors represent bandwidths. Again, $\eta(t) = time/100$. The cost function is the following:

$$c(s, X_j(t)) = \begin{cases} 2 & if\ X_j(t) = s, \\ 1 & if\ |X_j(t) - s| = 1, \\ 0 & otherwise. \end{cases} \tag{9}$$

And it penalizes neighbor routers using channels with the same frequency band or frequency band right next to each other.

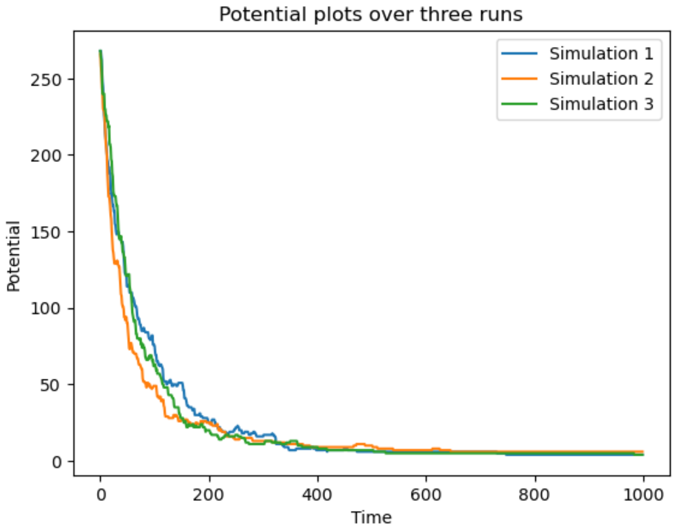

Fig. 10. Potential of the coloring algorithm over three simulations

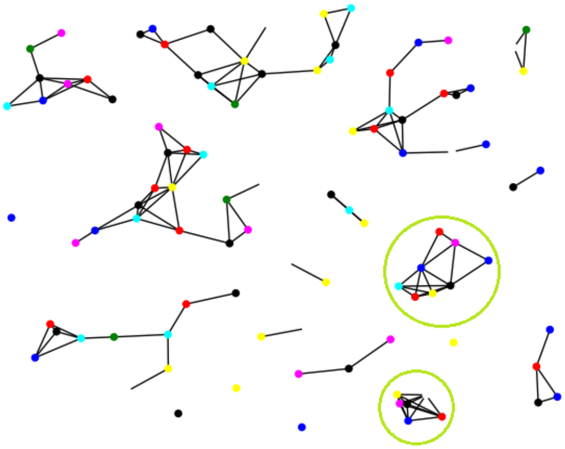

Fig. 11. Minimum potential configuration in the coloring

Therefore, two routers do not interfere with each other only if they assume two colors having distance greater than 1 in the color vector C = {1 = "red", 2 = "green", 3 = "blue", 4 = "yellow", 5 = "magenta", 6 = "cyan", 7 = "white", 8 = "black"}.

Three different simulations have been performed and their potentials are plotted in Fig. 10, while the best color configuration is reported in Fig. 11.

With respect to the previous exercise, the potential function (Eq. 7) reaches the minimum value of 4 and never 0. This behavior is due to the presence of some complete components of five and six nodes (see the highlights in Fig. 11). Because of Eq. 9, to get zero potential we require at least six colors $X_i(t), X_j(t) : |X_i(t) - X_j(t)| \geq 2$ in C, while we have only four of them. Therefore, the graph does not admit a coloring with only eight colors.

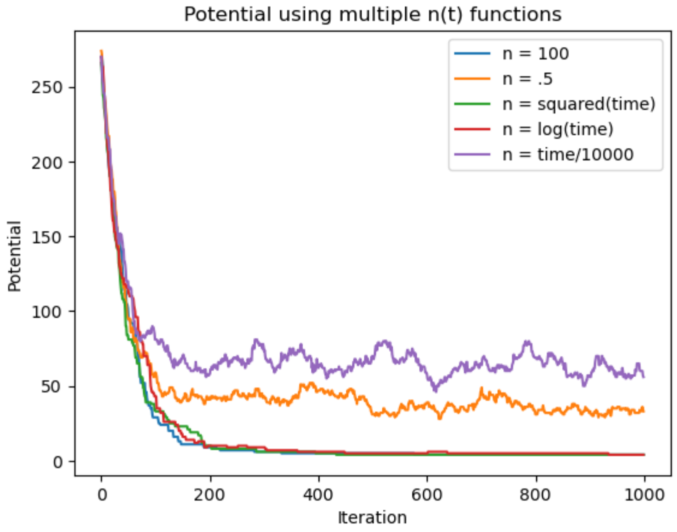

Fig. 12. Potential using different $\eta(t)$

C. Optional part

In Fig. 12 the potential computed using different $\eta(t)$ functions is plotted. In Eq. 6, $\eta(t)$ is the inverse of the noise, hence higher values of $\eta(t)$ reduce the noise. Furthermore, the bigger $\eta(t)$, the higher the probability that a node changes its state into a color that improves the potential at time t. As expected (see Fig. 12), the potential related to increasing or constant (with big values) $\eta(t)$ reaches lower values and it decays smoother than the potential of slower, or constant $\eta(t)$ with small value.