Homework 3 - Network Dynamics and Learning

Tommaso Mazzarini s334004
Network Dynamics and Learning
January 2025

Problem 1: Influenza H1N1 2009 Pandemic in Sweden

In the fall of 2009, Sweden faced a significant pandemic of the H1N1-virus, commonly known as *swine flu*. This pandemic affected approximately 1.5 million people across the country. The Swedish government responded by implementing a vaccination program that began in week 40 of 2009, ultimately reaching over 60% of the population in the following weeks.

Our analysis aims to understand this pandemic through mathematical modeling and computer simulation. We will focus on understanding how the disease spread through the population, analyzing the effectiveness of the vaccination program, and estimating key parameters that characterized the pandemic.

This study will progress through four stages:

- Learning the basics of simulation on a known graph and generating random graphs.
- Simulating disease propagation on a random graph without vaccination.
- Simulating disease propagation on a random graph with vaccination.
- Estimating network-structure characteristics and disease-dynamics parameters to fit real-world data about Swedish pandemic.

a) Preliminary Parts

In this section, we will address two main tasks. The first involves simulating an epidemic on a predefined graph, while the second focuses on generating a random graph using the preferential attachment model.

a.1) Epidemic on a Known Graph

The objective of this subsection is to simulate the SIR epidemic model on a symmetric k-regular undirected graph with a node set $V = \{1, \dots, n\}$, where each node is directly connected to the k nodes whose indices are closest to its own modulo n.

An example of this type of graph is illustrated in Figure1, where the number of nodes is 8 and the degree $k = 6$.

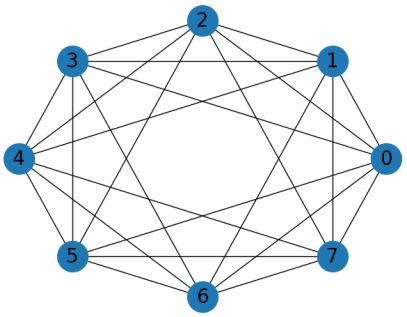

Figure 1: Symmetric k-regular graph with $k = 6$ and $n = 8$

To model the pandemic, we use the SIR (Susceptible-Infected-Recovered) epidemic model, which divides the population into three groups:

- **Susceptible (S)**: People who can catch the disease.
- **Infected (I)**: People who have the disease and can spread it.
- **Recovered (R)**: People who have recovered and are now immune.

The spread of the disease follows two main mathematical equations:

1. **Infection Equation**: The probability of a susceptible individual becoming infected depends on the number of their infected neighbors. This probability is given by:

$$P(X_i(t+1) = I \mid X_i(t) = S, m \text{ infected neighbors}) = 1 - (1-\beta)^m \quad (1)$$

2. **Recovery Equation**: The probability of an infected individual recovering is constant at each time step:

$$P(X_i(t+1) = R \mid X_i(t) = I) = \rho \quad (2)$$

Where:

- β is the probability of infection spreading between two people (between 0 and 1)
- ρ is the probability of recovery in each time step (between 0 and 1)
- m is how many infected neighbors a person has
- $X_i(t)$ represents the state (S,I,R) of individual i at time t

For our simulation, we considered a symmetric k-regular undirected graph with 500 nodes and $k = 6$. The disease transmission probability was set to $\beta = 0.25$, and the recovery rate was $\rho = 0.6$. The simulation covered a 15-week period, with each week representing one unit of time. The initial number of infected individuals was set to 10.

After running $N = 100$ simulations, we obtained the following results: the average number of new infections per week and the average total number of

susceptible, infected, and recovered individuals for each week. The results of our analysis are shown in Figures2.

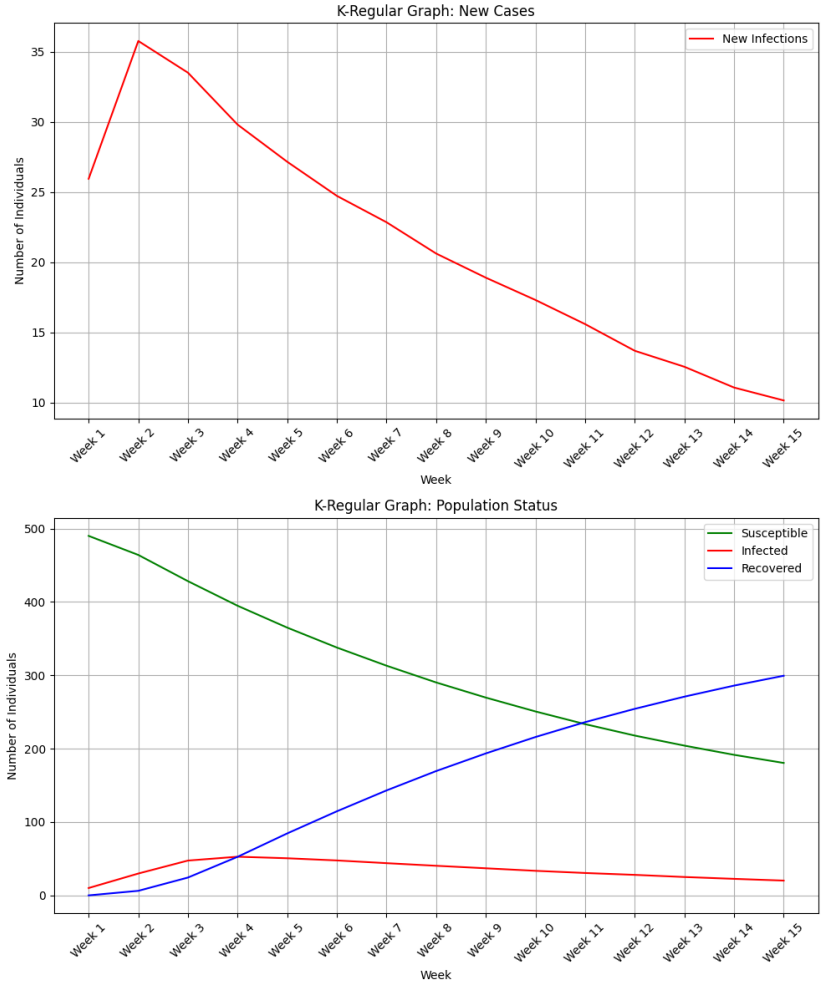

Figure 2: The average number of new infections per week (above) and the average total number of susceptible, infected, and recovered individuals for each week (below), simulated on a symmetric k-regular graph.

Looking at the graphs obtained, some relevant considerations regarding the dynamics of the epidemic emerge.

In the upper graph of Figure 2, which shows the number of newly infected cases per week, we observe a typical epidemic pattern according to the SIR model. Specifically, the number of new infections peaks around the second week, with approximately 35 new cases recorded. This spike indicates a rapid spread of the disease during the early stages. Afterward, the number of new infections gradually decreases, reflecting the slowing down of the epidemic as the susceptible population diminishes. By the final weeks of the simulation, the number of new cases becomes very low, signaling that the epidemic is subsiding.

The lower graph of Figure 2, instead, illustrates the overall status of the population, divided into susceptible, infected, and recovered individuals for each week. The susceptible population (green line) declines rapidly in the first few weeks, indicating widespread exposure to the disease. The infected population (red line) initially increases, reaching a peak of around 50 in week 3, before gradually declining as the number of new infections drops. Lastly, the recovered population (blue line) steadily rises throughout the simulation, eventually becoming the dominant group by the end of the period.

a.2) Random Graph Generation

After simulating the epidemic spread on a symmetric k-regular undirected graph, the next step is to repeat the simulation on a more complex graph, specifically a random graph generated according to the preferential attachment model.

In this section, we focus only on the process of generating the random graph, which will serve as the basis for the simulations discussed in the following sections.

The random graph is generated using the following approach: At time $t = 1$, we start with an initial graph G_1, which is a complete graph containing $k + 1$ nodes. Then, at each time step $t \geq 2$, a new graph $G_t = (V_t, E_t)$ is created by adding a new node to the graph G_{t-1}. This new node is connected to some of the existing nodes in V_{t-1} of G_{t-1} according to a stochastic rule based on the concept of preferential attachment.

The preferential attachment rule dictates that at each time step $t \geq 2$, the new node added at time t will have a degree $w_t(t) = c = k/2$, meaning it should establish c undirected links to the existing graph G_{t-1}. The choice of which nodes in V_{t-1} to connect to is made probabilistically, based on their current degrees.

Specifically, the probability that the new node n_t will form a link with an existing node $i \in V_{t-1}$ is given by:

$$P(\text{connection to node } i) = \frac{w_i(t-1)}{\sum_j w_j(t-1)},$$

where $w_i(t-1)$ is the degree of node i before the addition of the new node, and $\sum_j w_j(t-1)$ represents the total degree of all nodes in G_{t-1}.

In addition, if k is an odd number, then $c = k/2$ will not be an integer. However, it is still possible to achieve an average degree of k by alternating between adding $\lfloor k/2 \rfloor$ and $\lceil k/2 \rceil$ links when introducing a new node. As the graph grows larger, this approach converges to an average degree of k.

For this experiment, we generated a random graph with a large size of at least 900 nodes to ensure the desired properties. We observed that the average degree k of the graph, whether k is even or odd, consistently matches the input value.

b) Pandemic Simulation without Vaccination

In this section, using the SIR epidemic model implemented in Section a.1 and the random graph with preferential attachment from Section a.2, we simulate the spread of the disease in the absence of vaccination.

To perform this simulation, we generated a random graph as described in Section a.2, with 500 nodes and $k = 6$. The disease dynamics parameters were set to $\beta = 0.25$ (transmission probability) and $\rho = 0.6$ (recovery rate). As in Section a.1, the simulation covered a period of 15 weeks, with each week representing one unit of time and the initial number of infected individuals was set to 10.

After running the simulation $N = 100$ times, we obtained the following results: the average number of new infections per week and the average total number of susceptible, infected, and recovered individuals for each week. The results of our analysis are shown in Figure 3, respectively.

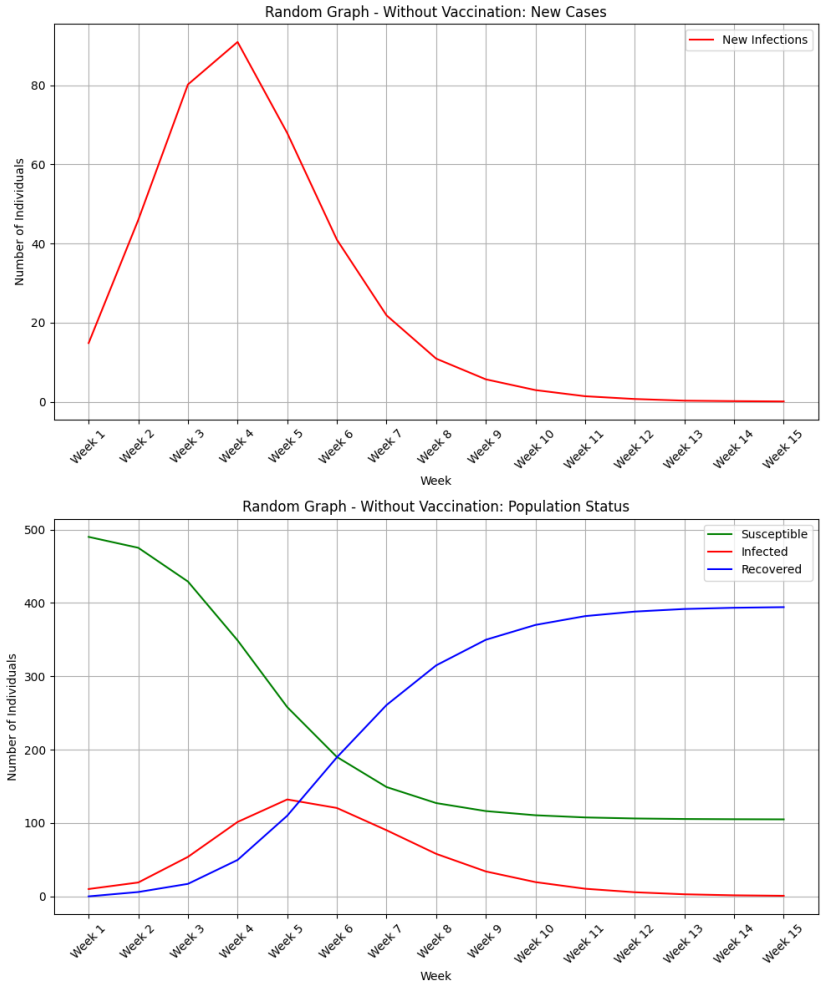

Figure 3: The average number of new infections per week (above) and the average total number of susceptible, infected, and recovered individuals for each week (below), simulated on a random graph with preferential attachment.

From the graphs, we observe that the epidemic spreads rapidly during the first few weeks, peaking around week 4 with over 80 new infections. This initial

surge reflects the structure of the random graph. After the peak, the number of new infections declines quickly as the susceptible population (green line) decreases and the recovered population (blue line) increases. The infected population (red line) peaks around week 5 and then steadily declines as individuals recover. Meanwhile, the recovered population grows consistently, eventually becoming the dominant group by the end of the simulation, when most individuals have recovered from the infection.

In comparison to the results in Section a.1, where a symmetric k-regular undirected graph was used, the random graph generated with the preferential attachment model shows a faster and more intense spread of the epidemic. The random graph, characterized by the presence of densely connected nodes, leads to a higher peak of infections (over 80 cases compared to 35 in the k-regular graph) and a faster decline in the susceptible population. On the other hand, the k-regular graph produces a more uniform and gradual spread due to its balanced structure. These differences emphasize the crucial role of network topology in shaping epidemic dynamics. Specifically, random graphs with preferential attachment amplify both the speed and intensity of contagion, in contrast to symmetric k-regular graphs, which result in a slower and more controlled spread.

c) Pandemic Simulation with Vaccination

To study the effect of vaccination on epidemic dynamics, we simulate the spread of the disease using the SIR model on a random graph generated with the preferential attachment model. The graph contains n nodes, with an average degree of k. The disease parameters remain the same as in the previous simulations: $\beta = 0.25$ (transmission probability) and $\rho = 0.6$ (recovery probability). The simulation spans 15 weeks, with an initial configuration of 10 randomly selected infected individuals.

Vaccination is introduced according to a predefined schedule, where the cumulative fraction of vaccinated individuals $\text{Vacc}(t)$ is represented by the list:

$$\text{Vacc}(t) = [0, 5, 15, 25, 35, 45, 55, 60, 60, 60, 60, 60, 60, 60, 60].$$

This means that by week 7, for example, 55% of the population is vaccinated, and 5% of the population is newly vaccinated in week 7. Vaccinated individuals are selected uniformly at random from the susceptible and infected populations. Once vaccinated, individuals immediately transition to the vaccinated state and can no longer spread or contract the disease.

We conducted the simulation $N = 100$ times, and the results are presented in Figure 4. The first plot displays the average number of new infections and new vaccinations per week, while the second plot illustrates the average total number of susceptible, infected, recovered and vaccinated individuals for each week.

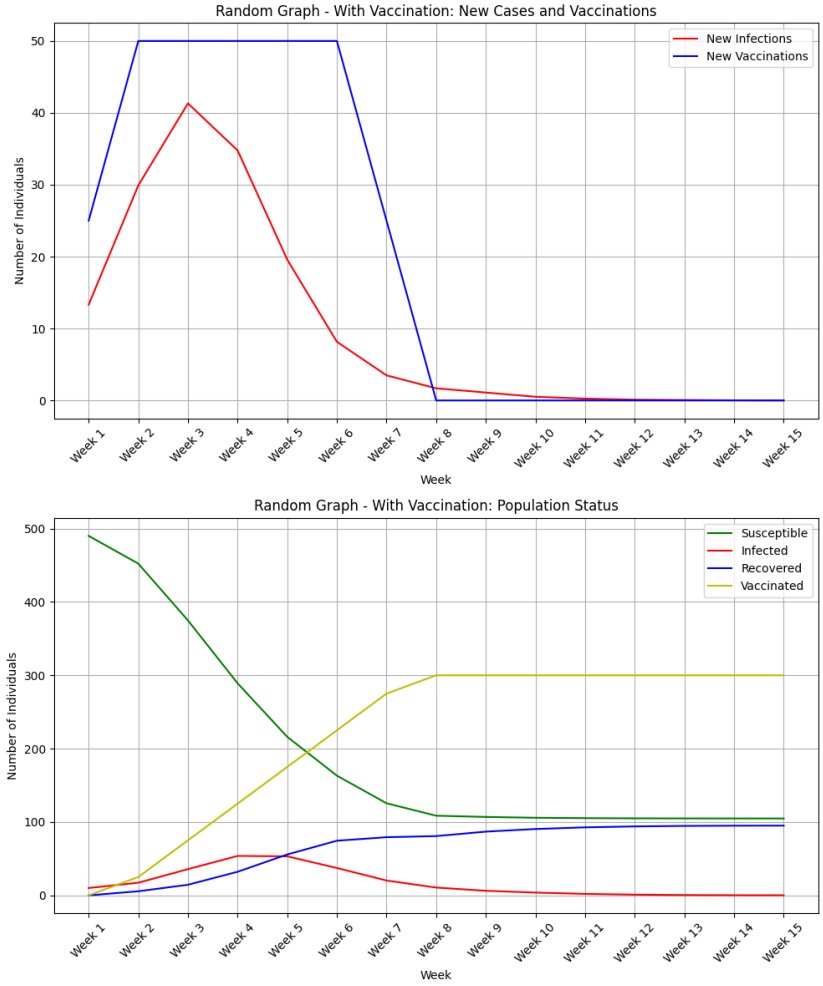

Figure 4: The average number of new infections and new vaccinations per week (above) and the average total number of susceptible, infected, recovered and vaccinated individuals for each week (below), simulated on a random graph with preferential attachment.

From the graphs, we can clearly observe the significant impact of vaccination on the epidemic's progression. The number of new infections peaks in week 3 with approximately 40 cases and then declines sharply, dropping to negligible levels by week 7. This rapid decline reflects the protective effect of vaccination, which curtails the spread of the disease. Simultaneously, the number of new vaccinations increases steadily until week 6, when the vaccination program reaches its target coverage, after which no new vaccinations are administered.

The population of susceptible individuals decreases rapidly in the early weeks due to both infections and vaccinations, stabilizing at a low level by week 7. The infected population peaks in week 4, similar to scenarios without vaccination but with a significantly smaller magnitude. The recovered population grows steadily over time but stabilizes at a lower value than in the absence of vaccination. This

outcome is due to vaccination preventing many individuals from contracting the disease, reducing the number of recoveries required. Meanwhile, the vaccinated population increases quickly, reaching 60% of the total population by week 6 and remaining constant thereafter.

In comparison to the scenario without vaccination (Section b), the introduction of vaccination dramatically reduces the peak number of infections (from over 80 to approximately 40) and causes the epidemic to subside earlier. The epidemic ends around week 7, whereas it persists much longer without vaccination. Additionally, the number of individuals remaining susceptible is significantly lower, as vaccination protects a large portion of the population before they can be exposed. Consequently, the overall number of infections and recoveries is substantially reduced in the vaccinated scenario, underscoring the effectiveness of the vaccination program in limiting the epidemic's spread and severity. These results demonstrate how vaccination not only reduces the epidemic's diffusion but also accelerates its resolution.

d) Swedish H1N1 Pandemic Data

In this section, we apply the methodologies and tools developed earlier to simulate the 2009 Swedish H1N1 pandemic.

In our simulation of the 2009 Swedish H1N1 pandemic, we focus on the period from week 42 of 2009 to week 5 of 2010. The weekly vaccination rates as a percentage of the population are given by:

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

For simplicity, Sweden's population is scaled down by a factor of 10^4, resulting in a simulated population of $n = |V| = 934$.The weekly number of newly infected individuals, based on real data, is:

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

where the first entry represents the initial number of infected individuals at the start of the simulation (week 42, 2009).

Before running the simulation, we must estimate the parameters k (average degree of the network), β (transmission probability), and ρ (recovery probability) that best approximate the real social structure and epidemic. To achieve this, we developed a gradient-based algorithm to explore the parameter space and find the values that minimize the difference between the simulated and real data.

The algorithm starts with an initial guess for the parameters, $k_0 = 10$, $\beta_0 = 0.3$, $\rho_0 = 0.6$, and initial steps $\Delta k = 1$, $\Delta\beta = 0.1$, $\Delta\rho = 0.1$. For each set of parameters within the ranges $k \in \{k_0 - \Delta k, k_0, k_0 + \Delta k\}$, $\beta \in \{\beta_0 - \Delta\beta, \beta_0, \beta_0 + \Delta\beta\}$, and $\rho \in \{\rho_0 - \Delta\rho, \rho_0, \rho_0 + \Delta\rho\}$, the algorithm performs the following steps:

- **Generate a Random Graph**: Using the preferential attachment model, a random graph $G = (V, E)$ with $|V| = 934$ nodes is generated. The graph's average degree is set to k.
- **Simulate the Pandemic**: Starting from week 42, the pandemic is simulated for 15 weeks using the vaccination scheme described above. This

simulation is repeated $N = 10$ times to compute the average number of newly infected individuals each week, $I(t)$.

- **Compute RMSE**: The root-mean-square error (RMSE) between the simulated data $I(t)$ and the real data $I_0(t)$ is calculated as:

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

- **Update Parameters**: The parameters k_0, β_0, ρ_0 are updated to the set of values that yield the lowest RMSE. If no improvement is found, the algorithm reduces the step sizes $\Delta k, \Delta\beta, \Delta\rho$ (e.g., halving them) and continues the search until convergence.

After applying this algorithm, the parameters that best approximate the real epidemic are found to be $k = 10$, $\beta = 0.2$, and $\rho = 0.7$, with a root-mean-square error of approximately 3.588. These parameters enable us to simulate the pandemic with a high degree of accuracy.

The following figure (Figure 5) compares the number of newly infected individuals each week, both real and simulated, during the period under study.

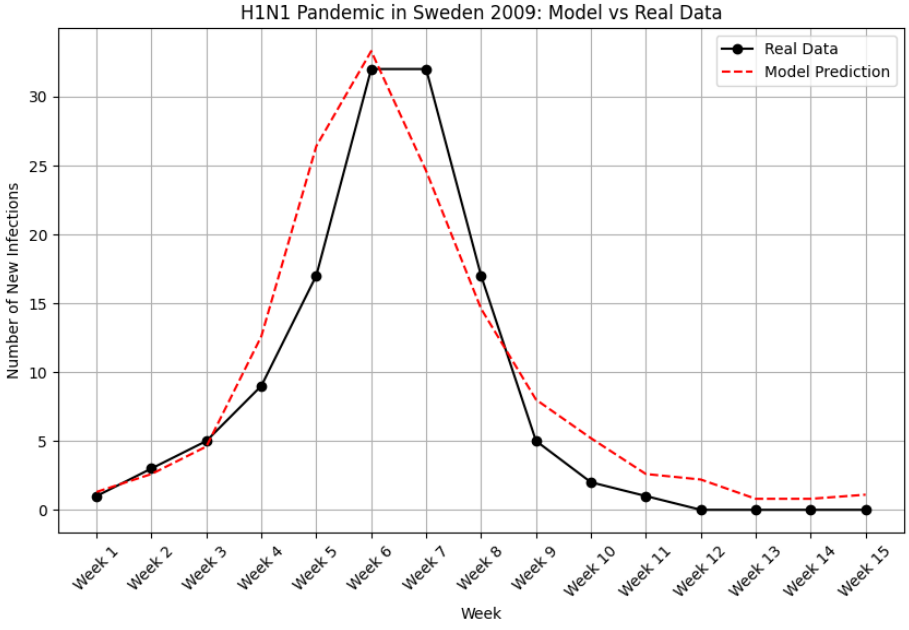

Figure 5: Weekly number of newly infected individuals during the H1N1 pandemic: real data (black line) and simulated data (red line).

The graph in Figure 5 demonstrates a strong agreement between the simulated model and the real data in capturing the trend and dynamics of new infections over the 15-week period. The simulation accurately reproduces the initial growth phase, where the number of new infections rises sharply, peaking

in week 6. Both the simulated curve (red dashed line) and the real data align closely, with a slight overestimation at the peak (approximately 33 infections in the simulation versus 32 in the real data).

The root-mean-square error (RMSE) of 3.588 highlights the model's high accuracy in capturing the progression of the pandemic.

With the calibrated parameters ($k = 10$, $\beta = 0.2$, $\rho = 0.7$), we extended the simulation to analyze the total number of susceptible, infected, recovered, and vaccinated individuals for each week. The results are presented in Figure 6.

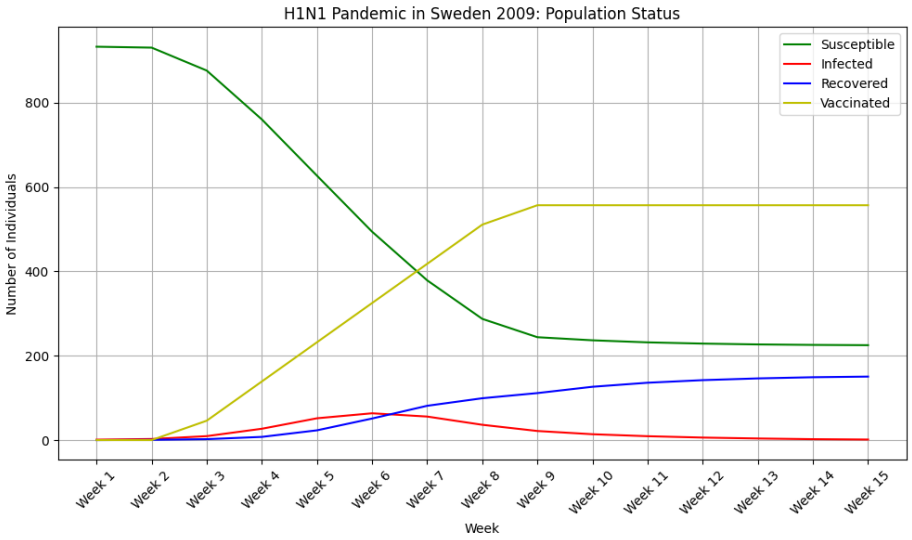

Figure 6: Average total number of susceptible, infected, recovered and vaccinated individuals for each week (below), simulated on a random graph with preferential attachment, with the best network-structure adn disease-dynamics parameters

The graph in Figure 6 clearly illustrates the effectiveness of the vaccination campaign in safeguarding a significant portion of the population from infection. Around week 6, the point where the susceptible (green) and vaccinated (yellow) lines intersect highlights a key moment when roughly half of the population had received the vaccine. his event played a significant role in reducing the epidemic's transmission. Throughout the simulation, active infections remained relatively low, likely due to the combined impact of vaccination and natural immunity acquired through recoveries. By the end of the simulation, the population was predominantly composed of vaccinated individuals (around 60%), with approximately 27% still susceptible and about 13% having recovered from the infection.

Problem 2: Graph Coloring

In this section, we examine graph coloring as a practical application of distributed learning within potential games. The goal of graph coloring is to assign a unique "color" to each node in an undirected graph, ensuring that no two

adjacent nodes share the same color.

This problem, in particular, consists of two distinct tasks:

1. **Line Graph Coloring:** In this task, we consider a simple scenario involving a line graph with 10 nodes. Each node can choose one of two colors (red or green). We simulate a distributed learning algorithm where, at each time step, a randomly selected node "wakes up" and updates its color based on a probability distribution. The aim is to track the evolution of this process and observe the behavior of the potential function over time.

2. **Wi-Fi Channel Assignment:** This more complex problem involves assigning frequency bands (represented by colors) to 100 routers in a network. The routers are modeled as nodes in a graph, with edges indicating interference between neighboring routers. The objective is to ensure that no two adjacent routers use the same or adjacent frequency bands, minimizing interference.

a) Line Graph Coloring

In this section, we simulate a distributed learning algorithm on a line graph consisting of 10 nodes arranged sequentially, where each node is connected to its immediate neighbors. The state of node i, or the action taken by node i in the context of game theory, is denoted by X_i, while the set of possible states or actions is $C = \{\text{red}, \text{green}\}$. Initially, all nodes are assigned the color red, as shown in Figure 7.

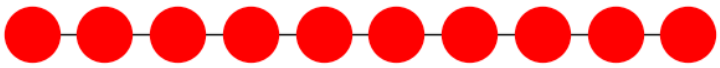

Figure 7: Line graph with all nodes initialized to the color red.

At each discrete time step, a randomly selected node "wakes up" and updates its color according to the following probability distribution:

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

where:

- $c(s, X_j(t))$ is the cost function, defined as:

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

- $\eta(t)$ is the inverse noise parameter, given by $\eta(t) = \frac{t}{50}$, which helps to control noise over time. The increasing nature of $\eta(t)$ ensures that the noise decreases as time progresses.

The potential function used to evaluate the system is given by:

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

where V is the set of nodes in the graph. The potential function $U(t)$ measures the overall "conflict" in the system.

If the potential function reaches zero, it indicates that there is no conflict between nodes, meaning that the system has reached a solution. This occurs when no node has an incentive to unilaterally change its color, leading to a Nash equilibrium.

After running the simulation for $N = 1000$ iterations, we observe the evolution of the potential function, shown in Figure 8.

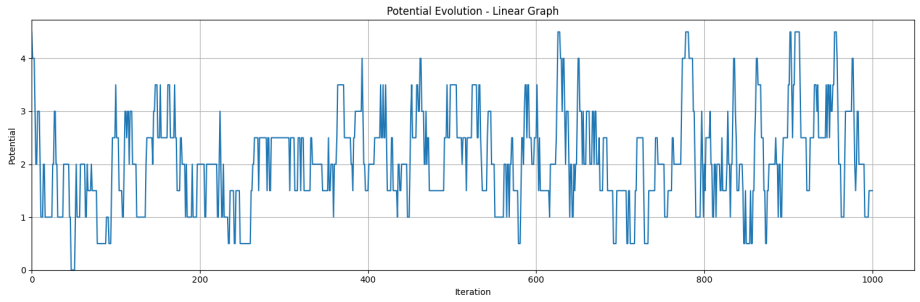

Figure 8: Evolution of the potential function on a line graph with 10 nodes.

From the graph, we observe that the potential function exhibits oscillatory behavior. This suggests that the algorithm does not converge to a solution within a finite number of steps. This can be attributed to the nature of the problem: each node must alternate between two colors, and adjacent nodes must have different colors. The game's zero-sum nature results in a cascading effect: when one node chooses a color, its neighbors are forced to choose the opposite color, leading to a cycle of color changes.

b) Wi-Fi Channel Assignment

In this section, we apply the distributed graph coloring algorithm from the previous section to a network of 100 routers. Each router is represented as a node in a graph, with edges indicating interference between neighboring routers. The goal is to assign frequency bands (represented by colors) to the routers such that no two neighboring routers share the same or adjacent frequency bands, thereby minimizing interference across the network.

This problem generalizes the traditional graph coloring problem by expanding the set of possible states (colors) to include multiple frequency bands. The cost function penalizes both direct conflicts, where two adjacent routers use the same frequency band, and near conflicts, where neighboring routers use adjacent

frequency bands. The cost function between two routers i and j is defined as:

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

where $X_j(t)$ represents the frequency band assigned to router j at time t, and s represents the frequency band chosen by router i.

The set of possible frequency bands (colors) is $C = \{1 : \text{red}, 2 : \text{green}, 3 : \text{blue}, 4 : \text{yellow}, 5 : \text{magenta}, 6 : \text{cyan}, 7 : \text{white}, 8 : \text{black}\}$, corresponding to the available frequency bands.

The potential function $U(t)$ and the inverse noise parameter $\eta(t)$ are the same as in the previous section. Initially, routers are assigned random frequency bands. At each step, a router is randomly selected to "wake up" and update its frequency band based on conflicts with its neighboring routers.

After running the simulation, we observe the evolution of the potential function, as shown in Figure 9, and the final router configuration, which illustrates their spatial arrangement and assigned frequency bands, shown in Figure 10.

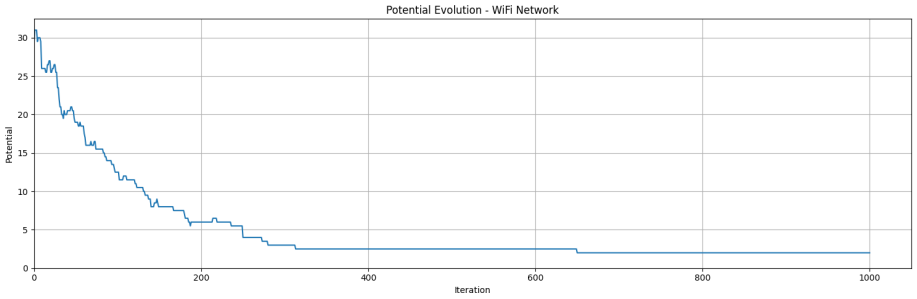

Figure 9: Evolution of the potential function in the Wi-Fi channel assignment problem.

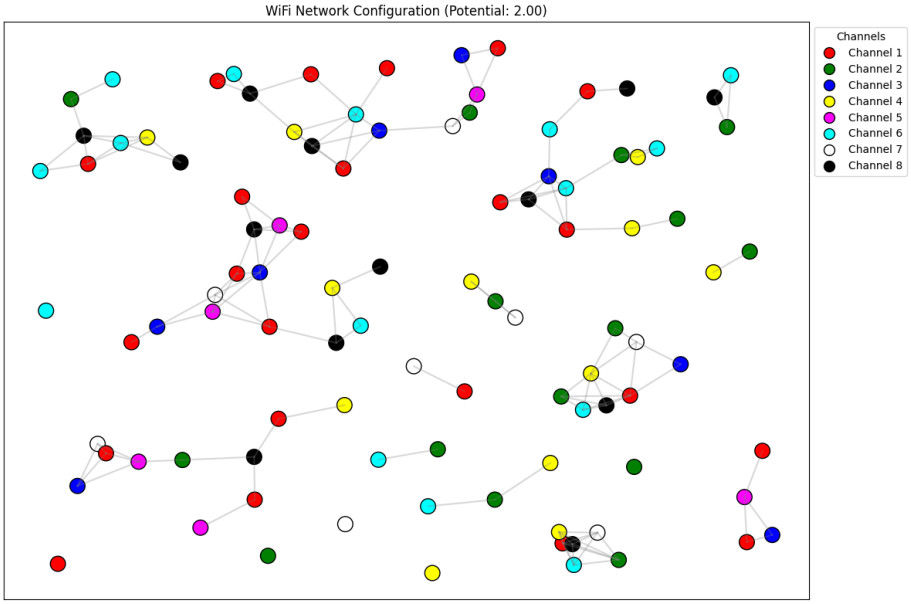

Figure 10: Optimal coloring configuration obtained for the Wi-Fi channel assignment problem.

From Figure 9, it is evident that the potential function exhibits a rapid decline during the first 100 iterations, dropping from approximately 30 to 10 as the algorithm quickly resolves major conflicts, such as overlapping or adjacent channel assignments. Between iterations 100 and 400, the decrease becomes more gradual, reflecting a fine-tuning phase aimed at resolving residual interference. After 400 iterations, the potential stabilizes at around 2.0, indicating the system has reached a robust and near-optimal configuration.

Figure 10 further supports this outcome, showing a well-distributed assignment of the eight frequency bands across the network. Neighboring routers are consistently assigned distinct or non-adjacent channels, effectively minimizing interference.

c) Effects of Inverse Noise Parameter

In this section, we examine the impact of different inverse noise parameters $\eta(t)$ on the convergence behavior of the graph coloring algorithm. Choosing an appropriate inverse noise parameter is essential for efficient convergence and obtaining high-quality solutions. The following $\eta(t)$ values were tested:

1. **Constant low value:** $\eta(t) = 0.1$
2. **Constant high value:** $\eta(t) = 1.0$
3. **Linear growth (slow):** $\eta(t) = t/500$
4. **Linear growth (moderate):** $\eta(t) = t/50$
5. **Linear growth (fast):** $\eta(t) = t/5$

6. **Logarithmic growth:** $\eta(t) = \log(1 + t)$

These parameters were chosen to explore different strategies for controlling the noise level, ranging from constant values (low and high) to various types of growth over time, including linear and logarithmic ones. The potential function $U(t)$ was tracked to monitor the algorithm's progress toward finding a solution with minimal conflicts between neighboring nodes.

Figure 11 compares the evolution of the potential function under each of these parameters.

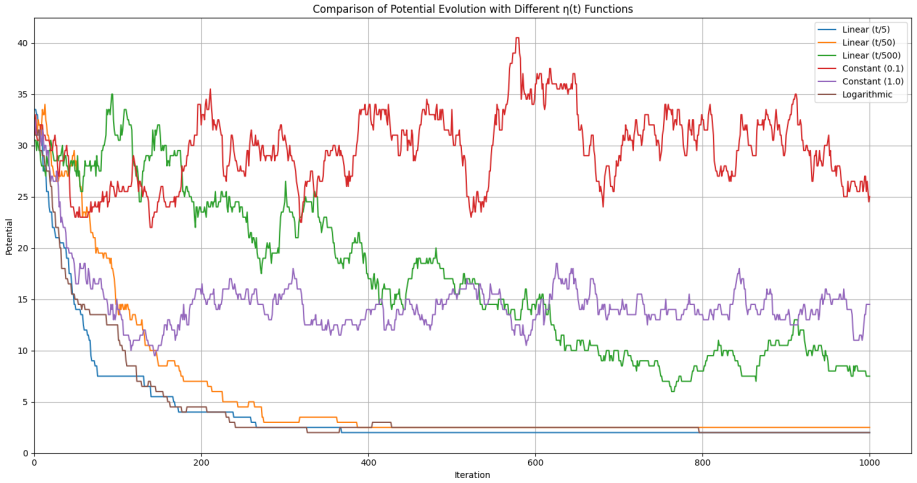

Figure 11: Comparison of different inverse noise parameters and their effect on the potential function.

From Figure 11, we observe that the choice of $\eta(t)$ significantly affects both the convergence speed and the quality of the final solution. The fast linear growth ($t/5$, blue line) exhibits the best performance, as it converges quickly and reaches the lowest potential (approximately 2) more efficiently than any other parameter. Moderate linear growth ($t/50$, orange line) follows a similar trajectory but with slightly slower convergence. On the other hand, the slow linear growth ($t/500$, green line) converges very slowly and, even after 1000 iterations, remains unstable, with potential values fluctuating around 8.

Logarithmic growth ($\log(1 + t)$, brown line) demonstrates behavior comparable to fast linear growth. It converges quickly and reaches the lowest potential alongside $t/5$, making it an effective alternative for this problem.

The constant parameter settings, $\eta(t) = 0.1$ (red line) and $\eta(t) = 1.0$ (purple line), perform poorly. The low constant value (0.1) results in high instability, with potential values oscillating widely between 25 and 35, showing no trend toward convergence. The high constant value (1.0) also exhibits instability, though to a lesser extent, with potential values fluctuating between approximately 10 and 18. In both cases, the lack of convergence suggests that constant parameters are inadequate for achieving a Nash equilibrium.