Taha Kamalisadeghian
Email: s328266@studenti.polito.it
Network Dynamics and Learning
Instructor: Prof. Fabio Fagnani

Homework - #3
ID: s328266
Term: Fall 2024 - 25
Due Date: 1^{th} December, 2024

Problem 1.1: Epidemic on a known graph
Simulate an epidemic on a symmetric k-regular graph $G = (\mathcal{V}, E)$ with $|\mathcal{V}| = 500$ nodes and $k = 6$. Let $\beta = 0.25$ and $\rho = 0.6$. With one week being one unit of time, simulate the epidemic for 15 weeks. You can choose an initial configuration with 10 infected nodes selected at random from the node set $\mathcal{V}$, or make a different choice of initial configuration (in the latter case, please briefly discuss your motivation).

Repeat the simulation $N = 100$ times and plot the following:

- The average number of newly infected individuals each week. In other words, plot how many people become infected each week (on average).
- The average total number of susceptible, infected, and recovered individuals at each week. In other words, plot how many individuals in total are susceptible/infected/recovered at each week (on average).

Solution:
Each individual is called agent and each of them is associated with a state related to the **H1N1-virus** infection. The states of the nodes $A = 0, 1, 2$ are represented as:

- S: Susceptible ($X_i(t) = 0$),
- I: Infected ($X_i(t) = 1$),
- R: Recovered ($X_i(t) = 2$).

Construct a symmetric k-regular graph G where every 500 nodes are connected to $k = 6$ neighbors. Randomly select 10 nodes as initially infected (I), while the remaining nodes are susceptible (S).

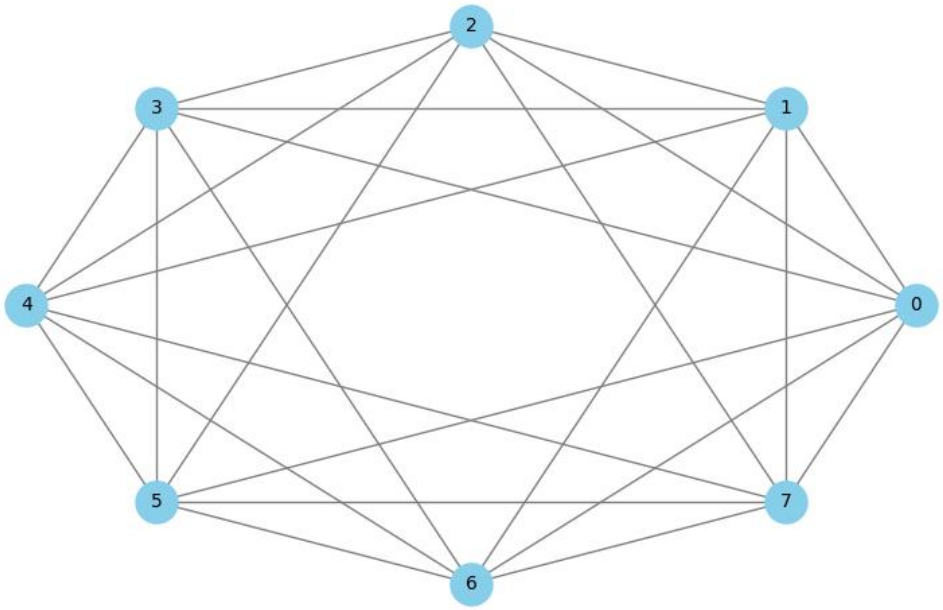


Figure 1: k-regular Graph, n=8, k=6

Disease Dynamics

For each node $i \in \mathcal{V}$ at time t:

1. **Infection:** A susceptible node i transitions to the infected state ($X_i(t+1) = 1$) with probability:

   $$P(X_i(t+1) = 1 \mid X_i(t) = 0) = 1 - (1 - \beta)^{m_i(t)},$$

   where $m_i(t)$ is the number of infected neighbors of node i at time t, and $\beta = 0.25$ is the infection probability.

2. **Recovery:** An infected node i transitions to the recovered state ($X_i(t+1) = 2$) with probability:

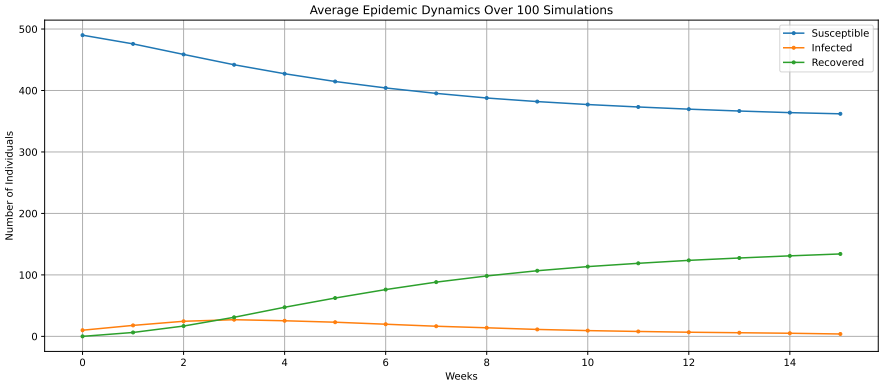
   $$P(X_i(t+1) = 2 \mid X_i(t) = 1) = \rho,$$

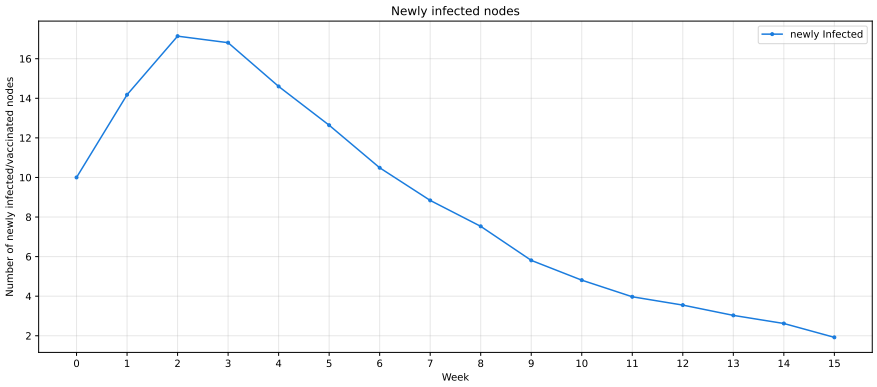
   where $\rho = 0.6$ is the recovery probability.

3. **State Update:** The state of each node is updated simultaneously at the end of each time step.

Simulation Procedure

The model will be running a discrete-time simplified version of the SIR model 100 times ($N = 100$) for 15 weeks ($T = 15$). For each of these simulations, a random different initial configuration with 10 different infected nodes is chosen on G according to a uniform distribution. The other nodes are all susceptible. In each simulation we record the $S(t)$, $I(t)$, $R(t)$, $\Delta I(t)$ at each time step.





The two plots illustrate the dynamics of an SIR epidemic model simulated over 15 weeks, averaged across 100 runs. The first plot shows the average number of susceptible, infected, and recovered individuals over time. The number of susceptible individuals decreases steadily as the infection spreads, leaving 72.9% susceptible by the end. Infected individuals initially increase, peaking around week 2–3, before declining as they recover, with only 0.7% remaining infected by week 15. The recovered population grows consistently, reaching 26.3% by the end of the simulation. The second plot highlights the number of newly infected individuals each week, which peaks early in the epidemic (around week 2–3 at 17 individuals) and then steadily declines to near zero as the susceptible population shrinks and recovery dominates. Together, these plots depict the classic epidemic progression, where infection spreads rapidly at first but slows as fewer individuals remain susceptible and more recover over time.

Problem 1.1.2: Generate a Random Graph
In this part, you will generate a random graph according to the preferential attachment model. The goal is to have a randomly generated graph with average degree close to k. The idea is the following: at time $t = 1$, we start with an initial graph G_1, that is complete with $k+1$ nodes. Then, at every time $t \geq 2$, create a new graph $G_t = (\mathcal{V}_t, \mathcal{E}_t)$ by adding a new node to G_{t-1} and connecting it to some of the existing nodes $\mathcal{V}_{t-1}$ of G_{t-1} chosen according to a stochastic rule.

Solution:
The preferential attachment algorithm generates a graph where the likelihood of a new node connecting to an existing node is proportional to the degree of the existing node. The graph generation begins with a complete graph of $k+1$ nodes, ensuring that all initial nodes are fully connected. The adjacency matrix A represents connections such that:

$$A_{ij} = \begin{cases} 1, & \text{if there is an edge between nodes } i \text{ and } j, \\ 0, & \text{otherwise.} \end{cases}$$

The degree of each node in this initial graph is k, calculated as the sum of edges for each node. For each new node added to the graph, the algorithm selects $c = \frac{k}{2}$ existing nodes as connection targets based on the preferential attachment rule. The probability of connecting to a node i is:

$$P(\text{connect to } i) = \frac{\text{degree}(i)}{\sum_j \text{degree}(j)},$$

where degree(i) is the current degree of node i, and the denominator is the sum of degrees of all existing nodes. This ensures that nodes with higher degrees are more likely to be selected. Once the targets are determined, edges are added between the new node and the selected nodes. The adjacency matrix A and the degree of each node are updated accordingly. This process repeats until the graph contains n nodes. As new nodes are added, the adjacency matrix grows, and the degree distribution evolves dynamically.

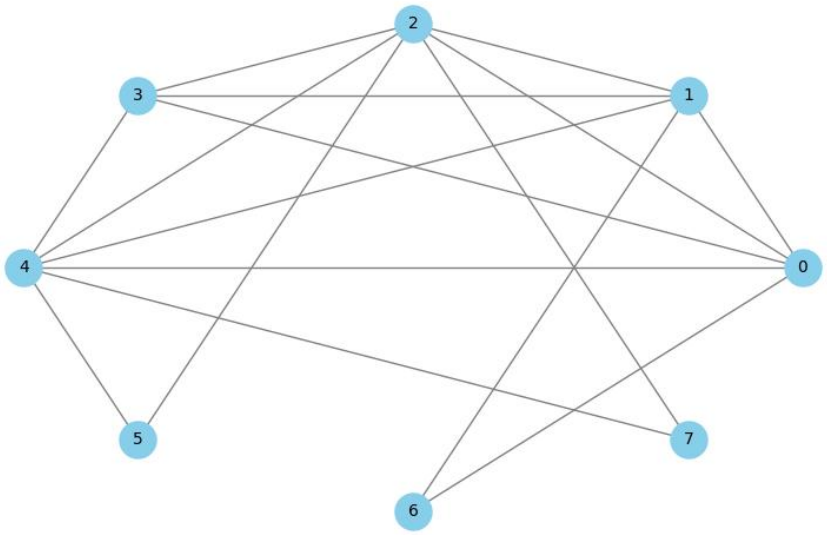


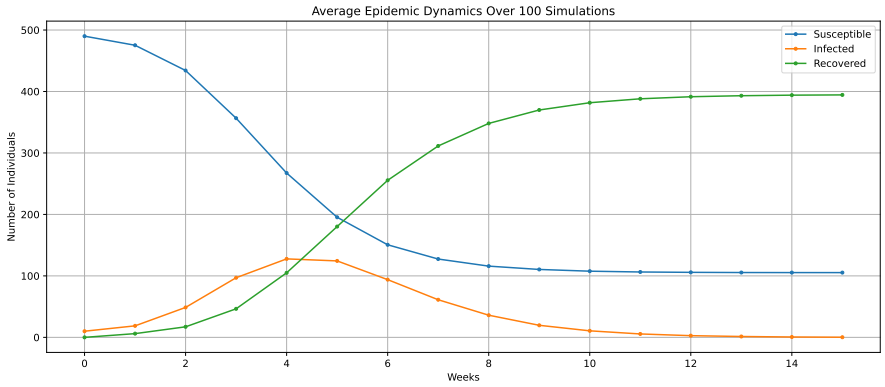
Figure 4: Preferential Attachment Graph, n=4, k=8

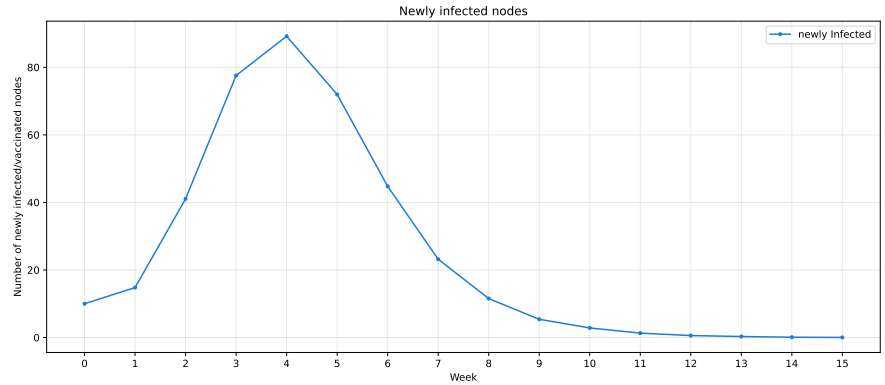
Problem 1.2: Simulate a pandemic without vaccination
Using the methods developed in Section 1.1, generate a preferential attachment random graph $G = (\mathcal{V}, \mathcal{E})$, with $|\mathcal{V}| = 500$ nodes. The average degree should be $k = 6$. Let $\beta = 0.3$ and $\rho = 0.7$. With one week being one unit of time, simulate the epidemic for 15 weeks. You can choose an initial configuration with 10 infected nodes selected at random from the node set $\mathcal{V}$, or make a different choice of initial configuration (in the latter case, please briefly discuss your motivation).

Do this $N = 100$ times and plot the following:

- The average number of newly infected individuals each week. In other words, you should plot how many individuals *become* infected each week.
- The average total number of susceptible, infected, and recovered individuals at each week. In other words, you should plot how many individuals *in total* are susceptible/infected/recovered at each week.

Solution:
Using the generated preferential attachment graph, an epidemic simulation was performed using the same parameters as in Problem 1.1





The preferential attachment model generates a graph with a **heterogeneous degree distribution**, meaning some nodes have significantly higher degrees than others (hubs). This contrasts with the uniform connectivity of k-regular graphs in Problem 1.1.

The plots show a faster and more intense outbreak. The susceptible population declines more rapidly, leaving only 21.1% uninfected by the end, while 78.9% recover, and the infected population drops to nearly zero. The newly infected individuals peak around week 4, higher and later than in Problem 1.1, and decline sharply thereafter, with the epidemic effectively ending by week 10. The faster spread in Problem 2 is driven by the presence of highly connected nodes in the preferential attachment graph, which act as super-spreaders, whereas the uniform connectivity in Problem 1.1 results in a slower, more evenly distributed epidemic.

Problem 1.3: Simulate a pandemic with vaccination
Using the method developed in the previous section, generate a random graph $G = (V, E)$, with $|V| = 500$ nodes. The average degree should be $k = 6$. Let $\beta = 0.25$ and $\rho = 0.6$. With one week being one unit of time, simulate the epidemic with vaccination for 15 weeks, using the vaccination scheme Vacc(t) above. You can choose an initial configuration with 10 infected nodes selected at random from the node set V, or make a different choice of initial configuration (in the latter case, please briefly discuss your motivation).

Do this $N = 100$ times and plot the following:

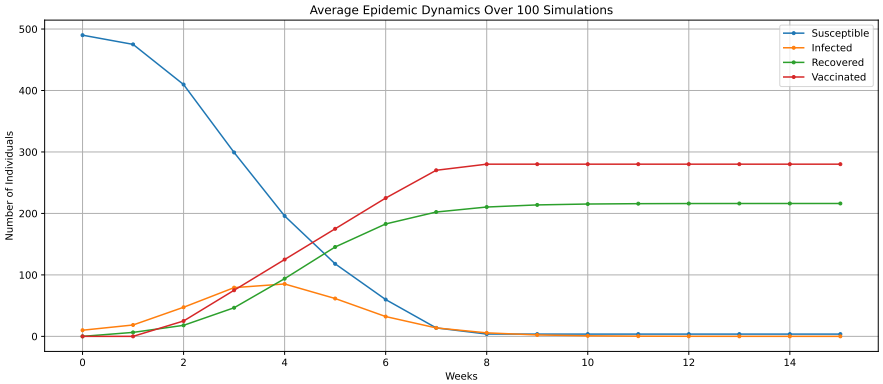
1. The average number of newly infected and newly vaccinated individuals each week.
2. The average total number of susceptible, infected, recovered, and vaccinated individuals at each week.

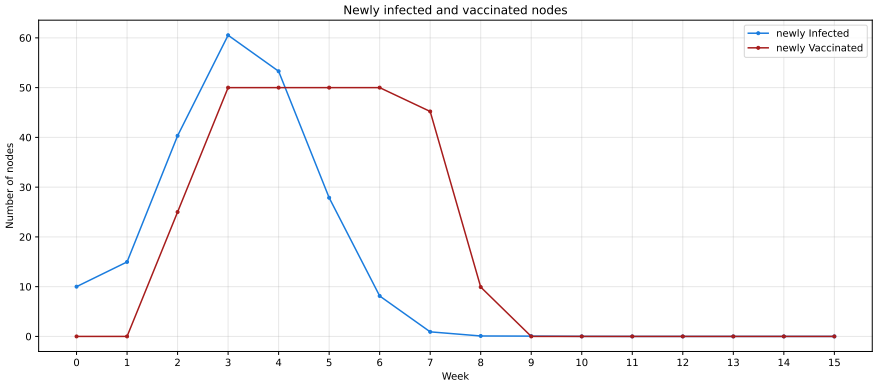
Solution:

To address this problem, we simulated an epidemic on a graph generated using the preferential attachment model with 500 nodes and an average degree of 6. The epidemic followed the SIRV model, which incorporates vaccination. Parameters were set as $\beta = 0.25$, $\rho = 0.6$, and a vaccination schedule Vacc(t) specifying the percentage of the population vaccinated each week. The vaccination schedule is defined as:

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

This means that by week 1, 0% of the population is vaccinated, by week 2, 5%, by week 3, 15%, and so on, until reaching a maximum of 60% by week 8, which remains constant thereafter. This schedule indicates the cumulative percentage of vaccinated individuals over time, with a rapid vaccination rollout in the early weeks to combat the epidemic effectively. The results, averaged over 100 simulations, are summarized in two key plots:





The plots for Problem 3 introduce vaccination into the epidemic simulation on a preferential attachment graph, significantly altering the dynamics compared to Problem 2. By the end of the simulation, only 0.7% of the population remains susceptible, while 43.2% have recovered and 56% have been vaccinated. Vaccination substantially reduces the spread and impact of the epidemic, as evident from the steep decline in infections after week 4. The newly infected individuals peak lower (around 50 nodes at week 4) compared to Problem 2 and drop rapidly as vaccination rates increase weekly. In contrast, in Problem 2 without vaccination, 21.1% of the population remained susceptible, and 78.9% recovered, with a faster and more widespread infection due to the lack of interventions.

The impact of vaccination is further highlighted by the steady rise in the number of vaccinated individuals each week, reaching 56% of the population by week 15, as shown in the second plot. This intervention not only prevents further infections but also indirectly accelerates the epidemic's decline by removing individuals from the susceptible pool. Compared to Problem 2, where the infection relies on highly connected nodes to spread, vaccination effectively neutralizes these super-spreaders, flattening the infection curve and reducing the overall severity and duration of the epidemic. This demonstrates the critical role of vaccination in mitigating the impact of epidemics, particularly in highly connected networks like the preferential attachment graph.

Problem 1.4: Simulate a pandemic with vaccination
Using the gradient-based search algorithm, estimate the average degree k and the disease-spread parameters β and ρ for the pandemic. Once you have found the best estimate, report what parameters you got. You should also show the following plots:

- The average number of newly infected individuals each week according to the model (with your best parameters) compared to the true value of newly infected individuals each week.
- The total number of susceptible, infected, recovered and vaccinated individuals at each week according to the model.

Solution:

In this section, we estimated the average degree k, the infection rate β, and the recovery rate ρ for the H1N1 pandemic in Sweden during the fall of 2009. The scaled-down population used for simulation consisted of $n = 934$ nodes. The observed weekly number of newly infected individuals, $I_0(t)$, over 15 weeks (from week 42, 2009, to week 5, 2010) The cumulative vaccination schedule, $\text{Vacc}(t)$, specifies the percentage of the population vaccinated by the end of each week (just like problem 1.3), are given as:

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

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

To estimate the parameters, we used a gradient-based search algorithm to minimize the root-mean-square error (RMSE) between the observed and simulated number of newly infected individuals per week. The RMSE is defined as:

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

where $I(t)$ is the average number of newly infected individuals per week obtained from $N = 50$ simulations for a given parameter set (k, β, ρ). Starting with initial parameter guesses $k_0 = 10$, $\beta_0 = 0.3$, and $\rho_0 = 0.6$, and initial step sizes $\Delta k = 4$, $\Delta\beta = 0.4$, and $\Delta\rho = 0.4$, the search iteratively refined these values by halving the step sizes at each iteration until convergence was achieved or the step sizes became negligible.

The optimized parameters obtained were: $k = 13, \quad \beta = 0.105, \quad \rho = 0.395$, with an RMSE value of 3.6397. This indicates a close match between the simulated and observed infection data, validating the estimated parameters.

Figure9 . compares the real-world infected data from the H1N1 pandemic with the simulated data generated by the model. The x-axis represents the number of weeks, while the y-axis represents the number of infected nodes (scaled population). The orange line depicts the real data, and the blue line represents the simulation results.

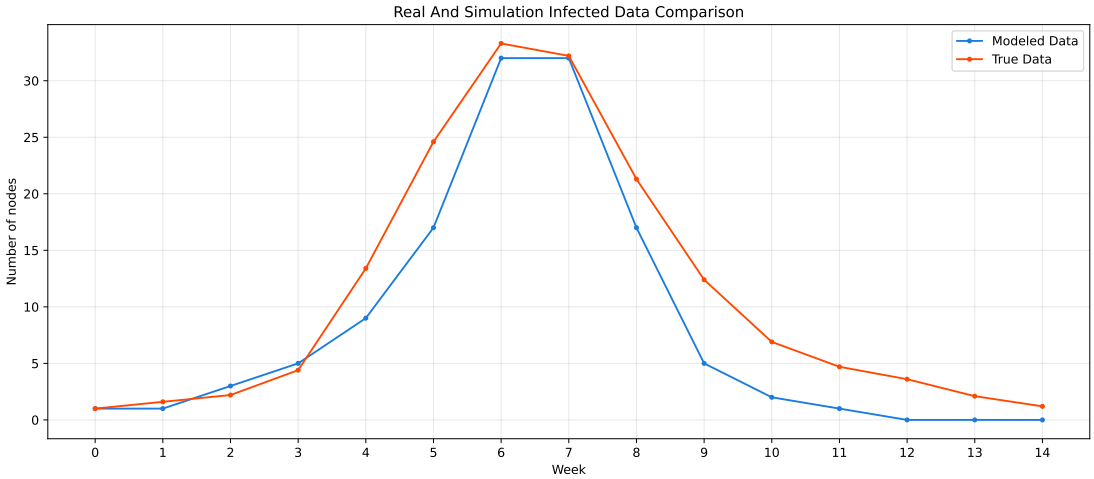


Figure 9: Comparison of Real and Simulated Infected

Both datasets follow a similar trend, with infections rising rapidly to a peak around weeks 5–7 and then declining sharply afterward. However, the simulation appears to lag slightly behind the real data, particularly during the rising phase, suggesting that the model may underestimate the initial spread rate. Despite this lag, the peak infection levels and overall shape of the curve align closely, indicating that the model captures the general dynamics of the epidemic well. Minor discrepancies could be attributed to differences in network assumptions or variations in parameter estimations.

Problem 2.a: Graph Coloring Using Distributed Learning

The graph coloring problem was addressed by simulating a distributed learning algorithm applied to a line graph with 10 nodes. The goal was to assign one of two colors, red or green, to each node such that no two adjacent nodes share the same color. The graph G was represented as a line graph with 10 nodes, connected sequentially such that each node i was linked to $i+1$ and $i-1$, except for the endpoints. The adjacency matrix W was used to represent this connectivity, where $W_{ij} = 1$ if nodes i and j were connected, and $W_{ij} = 0$ otherwise. Each node could take a state X_i from the set $C = \{\text{red, green}\}$. The cost function for coloring conflicts was defined as:

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

This ensures that a higher cost is incurred when adjacent nodes share the same color. The potential function $U(t)$ was introduced to quantify the total conflicts in the graph at time t:

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

The division by 2 avoids double-counting edge contributions. Initially, $U(0) = 7$, reflecting the conflicts when all nodes were colored red.

Distributed Learning Dynamics

At each discrete time step t, one node i was chosen uniformly at random to update its state based on a probability distribution determined by the cost function and a noise parameter $\eta(t)$. The transition probabilities were given by:

$$P(X_i(t+1) = s \mid X(t)) = \frac{\exp\left(-\eta(t) \cdot \sum_j W_{ij} \cdot c(s, X_j(t))\right)}{\sum_{s' \in C} \exp\left(-\eta(t) \cdot \sum_j W_{ij} \cdot c(s', X_j(t))\right)}.$$

Here, $\eta(t) = \frac{t}{50}$ controls the noise level, starting with a higher value to allow for exploratory updates and decreasing over time to stabilize the solution.

Results and Observations

Potential Function Evolution The potential function $U(t)$ decreased monotonically over the iterations, as shown in Figure 10. Initially, $U(0) = 7$ indicated multiple conflicts due to all nodes being assigned the same color (red). By $t = 20$, the system converged to $U(t) = 0$, indicating a conflict-free graph coloring where no adjacent nodes shared the same color. The steady decrease in $U(t)$ demonstrates the effectiveness of the distributed learning algorithm in resolving conflicts.

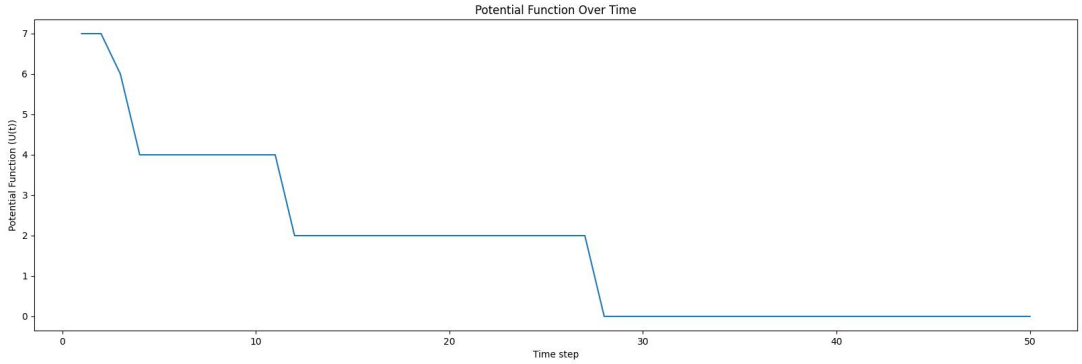


Figure 10: The evolution of the potential function $U(t)$ over time.

Graph Visualization

- **Initial State**: As shown in **Figure 11**, all nodes were red, reflecting the uniform initial condition and resulting in the highest potential $U(0)$.
- **Final State**: The graph achieved a valid coloring solution, alternating between red and green for adjacent nodes. This state is shown in **Figure 12**, where the line graph structure is evident, and the coloring satisfies the conflict-free condition.

Figure 11: Graph visualization in the initial state where $U(0) = 7$.

Figure 12: Graph visualization in the final state where $U(t) = 0$.

The distributed learning algorithm successfully resolved the graph coloring problem on a line graph. The convergence to a conflict-free state demonstrates the effectiveness of the probabilistic approach and highlights the role of the potential function as a measure of progress. The results and plots confirm the theoretical predictions, making this a robust method for solving similar graph-based problems.

Problem 2.b: Distributed Graph Coloring for Wi-Fi Channel Assignment

The goal of this problem was to address Wi-Fi channel assignment as a graph coloring problem using distributed learning. Each node in the graph represented a Wi-Fi router, and the edges denoted interference between routers within range. The task was to assign one of eight channels (represented by distinct colors) to each router while minimizing interference. In this problem, the cost function quantified interference levels based on the relative channel assignments of adjacent routers:

$$c(s, X_j) = \begin{cases} 2 & \text{if } s = X_j, \text{ (severe interference)}, \\ 1 & \text{if } |s - X_j| = 1, \text{ (mild interference)}, \\ 0 & \text{otherwise, (no interference)}. \end{cases}$$

This cost function ensured that the algorithm prioritized avoiding severe interference (same channels) while attempting to minimize mild interference (adjacent channels). The potential function begins at a high value, reflecting significant interference, and decreases steadily as the algorithm progresses. There are fluctuations early in the process due to high noise levels and probabilistic updates, but the potential stabilizes near zero after approximately 600 iterations. This behavior indicates that the distributed learning algorithm converges successfully, achieving a state with minimal interference, validating its effectiveness in resolving the channel assignment problem.

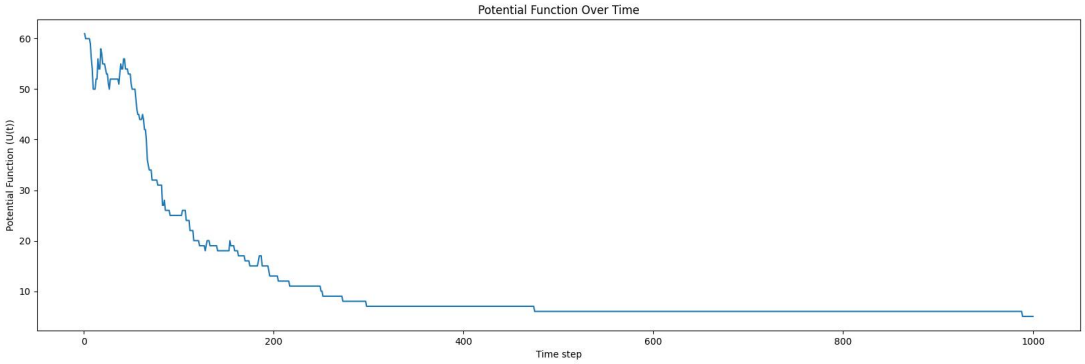


Figure 13: The evolution of the potential function $U(t)$ over time.

The final channel assignments are depicted in **Figure 14**, where nodes represent routers, and edges denote interference links based on the adjacency matrix. Each node is assigned a distinct color (channel) from a set of eight available colors. The algorithm ensures that neighboring nodes have minimal interference by avoiding the same or adjacent channels. The diverse color distribution and minimal color overlap among connected nodes demonstrate the algorithm's ability to reduce interference effectively, achieving a near-optimal channel assignment for the network.

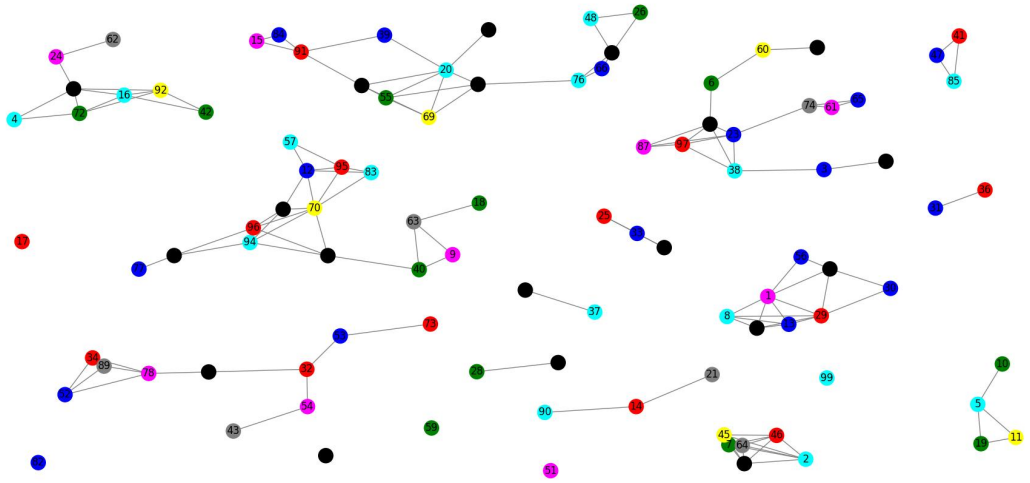


Figure 14: Final graph coloring for Wi-Fi channel assignment.

Conclusion

The distributed learning algorithm successfully addressed the Wi-Fi channel assignment problem by leveraging a graph coloring approach. The results validate the algorithm's ability to minimize interference, achieving an optimal solution with practical implications for real-world wireless communication systems.