POLITECNICO DI TORINO

Master Degree course in Data Science and Engineering

Network Dynamics and Learning

Homework 3

Professors

Prof. Fabio FAGNANI
Prof. Giacomo COMO
PhD. Leonardo CIANFANELLI

Student

Gustavo NICOLETTI ROSA
Claudio FANTASIA

ACADEMIC YEAR 2023-2024

Contents

Chapter 1

Exercise 1: Influenza H1N1 2009 Pandemic in Sweden

The epidemics have been simulated using a SIR model, where S is susceptible, I is infected and R is recovered. Every nodes is represented by a label: $X_i \in \{S, I, R\}$ and the flow pattern between compartments is represented in 1.1

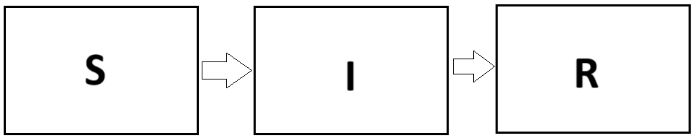


Figure 1.1: Compartimental model for SIR

We are going to discuss how the dynamics of the epidemics have been designed:

It is important to notice that at each time step (i.e. for each week) every node can change using an independent dynamic.

The nodes that are in the susceptible state can only change into the infected state. The single susceptible node can change into an infected one with a probability according to the infection probability β and the number of infected neighbors of the node. While the nodes that are infected can change into the recovered states with a probability ρ. Instead, the recovered nodes can not change into other states and their dynamics are stationary. Given that the epidemic dynamics are governed by these formulas:

$$P(X_i(t+1) = I | X_i(t) = S, \sum_{j \in \mathcal{V}} W_{ij} \delta^I_{X_j(t)} = m) = 1 - (1 - \beta)^m \tag{1.1}$$

$$P(X_i(t+1) = R | X_i(t) = I) = \rho \tag{1.2}$$

where $\sum_{j \in \mathcal{V}} W_{ij} \delta^I_{X_j(t)}$ is the number of infected neighbors for node i.

Notice also that in the following simulations, we have started the initial infected nodes randomly using a shuffle function of the vector injected with initial infections.

1.1 Problem 1

Epidemic on a symmetric k-regular graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ *with* $|\mathcal{V}| = 500$ *and* $k = 4$*,* $\rho = 0.7$*,* $\beta = 0.3$*, 10 initial random infected nodes. With one week being one unit of time and epidemics simulated for 15 weeks.*

It is important to notice that in the k-regular graph, every single node has the same degree equal to k. So the choice of the initial infected nodes does not influence the simulation of the epidemics. So the stochasticity is given by the infection and recovery probabilities, but not by the choice of initial nodes.

For a more accurate result, we ran 100 simulations of the epidemic. Here are shown the results for:

The average number of newly infected individuals for each week in Fig. (1.2).

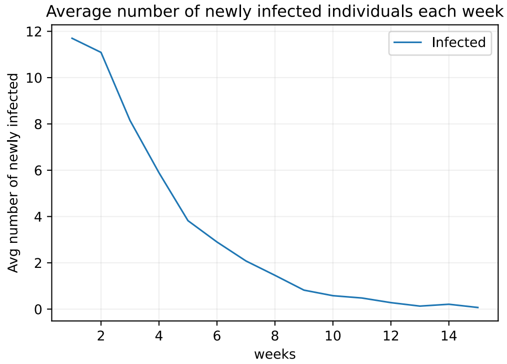


Figure 1.2: Average number of newly infected individuals for each week

The average total number of susceptible, infected, and recovered individuals for each week in Fig. 1.3.

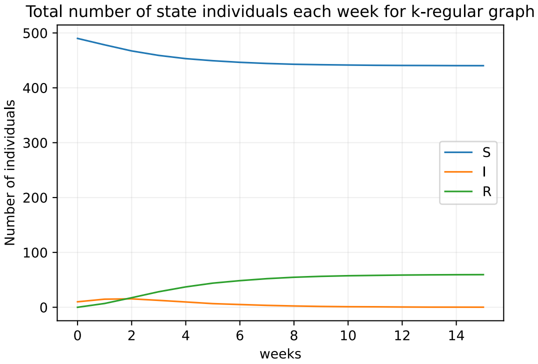


Figure 1.3: Average number of individuals for each week

1.2 Problem 2

Epidemic on a preferential attachment random graph $\mathcal{G} = \{\mathcal{V},\mathcal{E}\}$ with $|\mathcal{V}| = 500$ and average degree per node $k = 6$, $\rho = 0.7$, $\beta = 0.3$, Initializing 10 random infected nodes. With one week being one unit of time and epidemics simulated for 15 weeks.

First of all, it should be noticed that in a preferential attachment random graph, nodes are added to the network over time, and each new node tends to connect to existing nodes with a probability that is proportional to the number of connections those existing nodes already have. Leading to few nodes that have far larger connections with respect to the other nodes in the network.

The procedure for building the graph is to start at time $t = 1$ with a complete graph $\mathcal{G}_1$ with $k + 1$ nodes. Then at every time $t \geq 2$ a new node is added to the graph and it creates $k/2$ links with the other nodes in the network according to the probability distribution with p_i the probability to create a link with node i (Eq. 1.3), being $w_i(t-1)$ the degree of node i at previous iteration:

$$p_i = \frac{w_i(t-1)}{\sum_{j \in \mathcal{V}_{t-1}} w_j(t-1)} \tag{1.3}$$

Notice that if k is an odd number, for each time step t, we add $\lfloor k/2 \rfloor$ links if t is even, otherwise, we add $\lceil k/2 \rceil$ if t is odd We can notice that at the end of graph generation, there would be nodes with a way higher degree than others, as shown in Fig. 1.4.

This implies that the dynamics can significantly change based on the random selection of the initial infected nodes. So the number of simulations should be high enough to obtain an accurate expected value of individuals' dynamics. In our case, the number of simulations has been set up to 100.

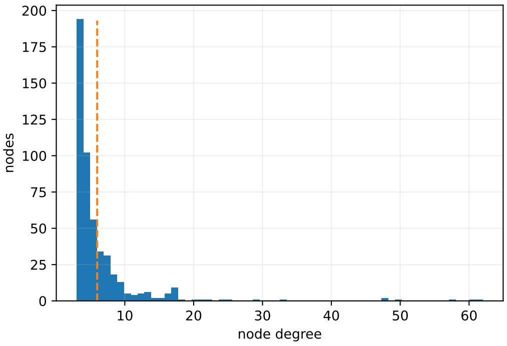


Figure 1.4: Histogram of the degree for each node in the preferential attachment random graph, the orange dashed line represents the average degree $k = 6$

Given the fact that the average degree for each node is $k = 6$, we expect a higher number of infections with respect to the case of Problem 1.1. The results for the average number of newly infected individuals and the average total number of susceptible, infected, and recovered individuals each week are shown in Fig. 1.5 and Fig. 1.6.

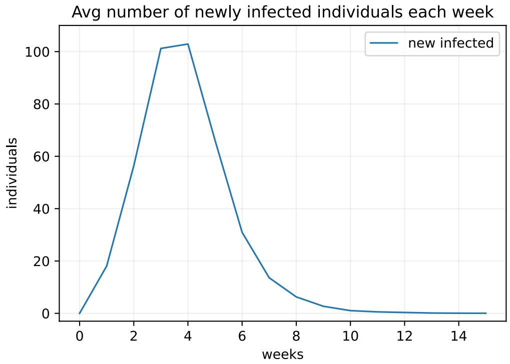


Figure 1.5: Average number of newly infected individuals for each week using preferential attachment random graph with $k = 6$

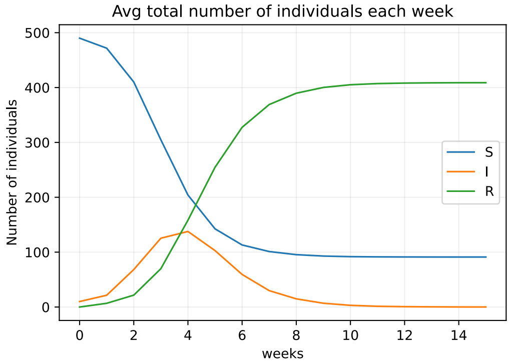


Figure 1.6: Average total number of individuals for each week using preferential attachment random graph with $k = 6$. Final values: $S_{avg}(15) = 90.32$, $I_{avg}(15) = 0.04$, $R_{avg}(15) = 409.64$

1.3 Problem 3

Epidemic on a preferential attachment random graph $\mathcal{G} = \{\mathcal{V},\mathcal{E}\}$ *with* $|\mathcal{V}| = 500$ *and average degree per node* $k = 6$, $\rho = 0.7$, $\beta = 0.3$, *Initializing 10 random infected nodes. With one week being one unit of time and epidemics simulated for 15 weeks. Also using the vaccination scheme* $Vacc(t)$.

$$Vacc(t) = [0, .05, .15, .25, .35, .45, .55, .60, .60, .60, .60, .60, .60, .60, .60]$$

In this section, we are going to face a similar instance of the Problem 1.2. We have the same random graph with preferential attachment with average degree $k = 6$, the same number of nodes and ratios. But we have also added the vaccination intervention in the whole schema.

At the beginning of each time step (i.e. week), a percentage of the population at random is treated with a vaccine, meaning that if a node is vaccinated can not be infected and can not infect others anymore. Notice that every node in every state can become vaccinated, even the recovered one.

It is worth mentioning that we did not introduce a vaccinated compartment, but we just added an attribute *vaccinated* to the node, meaning that if an infected node got vaccinated, it did not pass to the recovered (or vaccinated) at once, but the node had to wait for the succession of dynamics from I to R.

Using the given vector $Vacc(t)$, at each week $t = i$, we take the i-th element from the vaccinated vector and compute the percentage of the population minus the already vaccinated nodes, meaning that nodes can not be vaccinated twice.

Results are shown the Fig. 1.7 and Fig. 1.8. In comparison with the system without vaccination, we have obtained 61% of the final recovered, which corresponds to the cumulative amount of Infected of all weeks, since there are no more Infected by week 15, on average. Even though the epidemic has spread out, the peak of infected people is lower with respect to Problem 1.2 because we also introduced vaccination in the system.

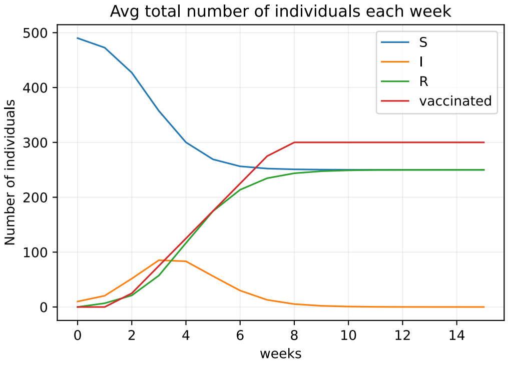


Figure 1.7: Average total number of individuals for each week using preferential attachment random graph with $k = 6$ and vaccination. Final values: $S_{avg}(15) = 249.91$, $I_{avg}(15) = 0.02$, $R_{avg}(15) = 250.17$

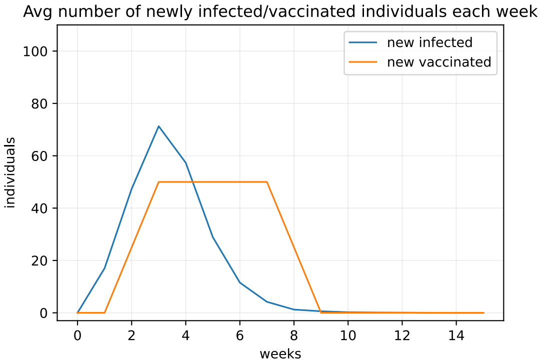


Figure 1.8: Average number of newly infected and vaccinated individuals for each week using preferential attachment random graph with $k = 6$

1.4 Problem 4 - Pandemic in Sweden

Epidemic on an unknown graph with $|\mathcal{V}| = 934$, unknown rates β and ρ. With a known vaccinated percentage of population $Vacc(t)$ and infected individuals per week $I_0(t)$

$$Vacc(t) = [.09, .16, .24, .32, .40, .47, .54, .59, .60, .60, .60, .60, .60, .60, .60, .60]$$

Here the main objective is to model the topology of Sweden's population and the rates of infection β and recover ρ given the real data about the actual vaccinated and infected individuals. We first make some assumptions about the topology of the network, the rates and then we use the **RMSE** measure to understand how close these assumptions are close to reality. First of all, we assumed as the topology of Sweden's population the preferential attachment random graph with a variable average degree k. We applied a grid search using the parameters:

$$k : [7,8,9,10]$$
$$\beta : [0.2,0.3,0.4]$$
$$\rho : [0.3,0.4,0.5,0.6,0.7,0.8,0.9,1]$$

We found out, using **100 simulations** per parameters, that the ones that fitted better with the actual data were around: $k : 10$, $\beta : 0.2$, $\rho : 0.4$ with an RMSE $\sim$ **6.5**.

Using a more fine-grained grid search with values around the optimal ones previously found and increasing the number of simulations to 250, we found out that the best performances were achieved using $k = 10$, $\beta = 0.2$, $\rho = 0.375$ with an RMSE $\sim$ **5.9**. Notice also that the results of rmse and parameters are very sensitive to the simulations made and may vary a bit
The results of using the best parameter are shown in Fig. 1.9 and Fig. 1.10

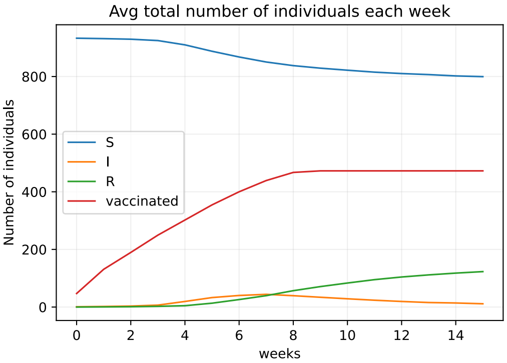


Figure 1.9: Average total number of individuals for each week in Sweden's pandemic.

1.4.1 Challange: Small-World Random Graph

The preferential attachment random graph is typically associated with scale-free networks. Scale-free networks exhibit a heterogeneous degree distribution, with a few highly

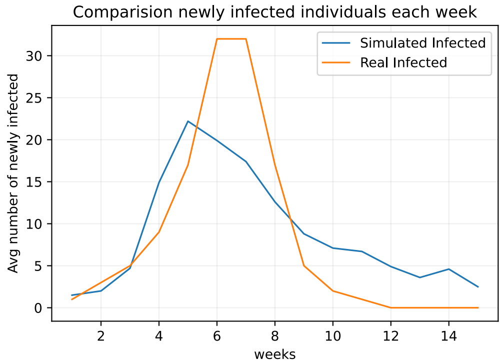


Figure 1.10: Comparision between average newly infected and real newly infected for each week in Sweden's pandemic

connected nodes and many nodes with fewer connections. We wanted to face Sweden's pandemic simulation using a different approach switching our initial guess of a random graph from a scale-free network to a small-world network, which is characterized by short average path lengths between nodes and high clustering coefficients.

For these reasons, we have used a **Watts Strogatz** model.

After some random simulations, we found out that the most promising intervals of value for the Strogatz model were:

$$k : [12{,}14{,}16]$$
$$p : [0.4, ,0.5{,}0.6{,}0.7]$$
$$\beta : [0.2]$$
$$\rho : [0.5{,}0.6{,}0.7{,}0.8]$$

With k links to the nearest neighbors and p the probability of rewiring some links at random.

Using an appropriate grid search we found out that the best values were: $k : 14$, $p : 0.7$, $\beta : 0.2$, $\rho : 0.8$ with an RMSE $\sim$ **5.4**.
We show the results in Fig. 1.11 and Fig. 1.12.

We can make some pretentious assumptions: it is known that *Influenza H1N1* had a length of ~ 1 week, meaning that ρ should be around the value of 1. Given these facts and that we achieved a lower RMSE using small-world properties and a value of ρ closer to 1. It is reasonable to say that a small-world random graph would fit better in our epidemic case.

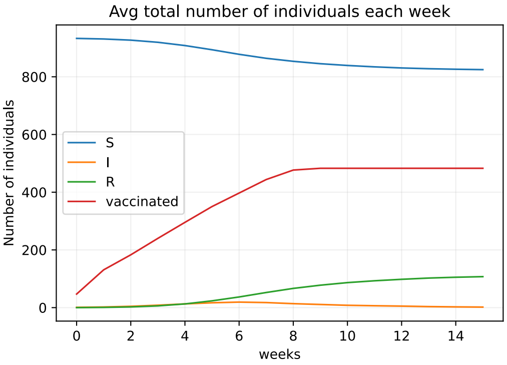


Figure 1.11: Average total number of individuals for each week in Sweden's pandemic using a small-world topology

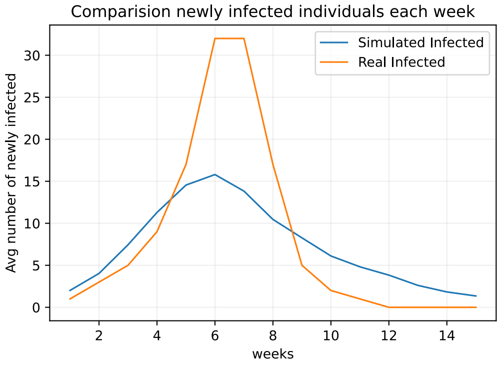


Figure 1.12: Comparision between average newly infected and real newly infected for each week in Sweden's pandemic using a small-world topology

Chapter 2

Exercise 2: Coloring

With distributed learning in a potential game setting, with Noisy Best Response (NBR), we will approximately find a solution for the coloring problem, which belongs to the class NP-complete.

2.1 Part (a)

In this example, study a line graph with 10 *nodes. Denote the* i*-th node state by* $X_i(t)$ *and the set of possible states by* $\mathcal{C} = \{red, green\}$*. At initialization, each node is red, i.e,* $X_i(t) = red$ *for all* $i = 1, ..., 10$*. Every discrete time instance* t*, one node* $I(t)$*, chosen uniformly at random, wakes up and updates its color.*

Firstly, we define some equations that will be used in the NBR. The cost function is defined by:

$$c(s, X_j(t)) = \begin{cases} 1, & \text{if } X_j(t) = s \\ 0, & \text{otherwise} \end{cases} \tag{2.1}$$

The parameter $\eta(t)$ is defined by

$$\eta(t) = \frac{t}{100} \tag{2.2}$$

With $X_i(t) \in \mathcal{C}$, W the adjacency matrix of graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$, $c(a, b)$ the cost function (Eq. 2.1), and $\eta(t)$ a parameter in function of time (Eq. 2.7), when a node wakes up, it chooses the next color given the probability distribution P (Eq. 2.3), and at each time step, we evaluate the graph by using the potential function $U(t)$ (Eq. 2.4), $U(t) = 0$ is a stop criteria.

$$P(X_i(t+1) = a | X(t), I(t) = i) = \frac{e^{-\eta(t) \sum_j W_{i,j} c(a, X_j(t))}}{\sum_{s \in \mathcal{C}} e^{-\eta(t) \sum_j W_{i,j} c(s, X_j(t))}} \tag{2.3}$$

$$U(t) = \frac{1}{2} \sum_{i,j \in \mathcal{V}} W_{ij} c(X_i(t), X_j(t)) \tag{2.4}$$

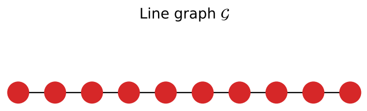


Figure 2.1: Initial condition of $\mathcal{G}$

Initially, we start with all nodes of $\mathcal{G}$ of the color red (Fig. 2.1).

We then applied such an algorithm to $\mathcal{G}$ and obtained the following results. The final state reaches zero potential at time 110 (Fig. 2.3), and the solution is presented in Fig. 2.2, reaching the trivial solution for a line graph and $|\mathcal{C}| = 2$, that is, the alternating of the colors.

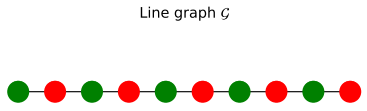


Figure 2.2: Solution of the coloring problem in the line graph $\mathcal{G}$

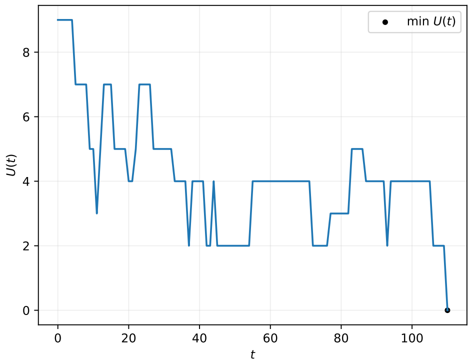


Figure 2.3: Potential of $\mathcal{G}$ in time

2.2 Part (b)

Next, we use the coloring algorithm for the problem of assigning wifi-channels to routers. The adjacency matrix of a network of 100 *routers is given in* `wifi.mat` *and the routers' coordinates are given in* `coord.mat`*. Here, a link between two nodes means that the two routers are able to interfere with each other. The set of possible states is* $\mathcal{C} = \{1 : red, 2 : green, 3 : blue, 4 : yellow, 5 : magenta, 6 : cyan, 7 : white, 8 : black\}$*, where colors represent* *frequency bands, and the cost function $c(s, X_j(t))$ symbolizes that routers that are close by should not use channels with the same frequency band or a frequency band right next to each other.*

Defining the new cost function (Eq. 2.5), we can use the same algorithm as in Part (a) 2.1 to the graph of routers (Fig. 2.4).

$$c(s, X_j(t)) = \begin{cases} 2, & \text{if } X_j(t) = s \\ 1, & \text{if } |X_j(t) - s| = 1 \\ 0, & \text{otherwise} \end{cases} \tag{2.5}$$

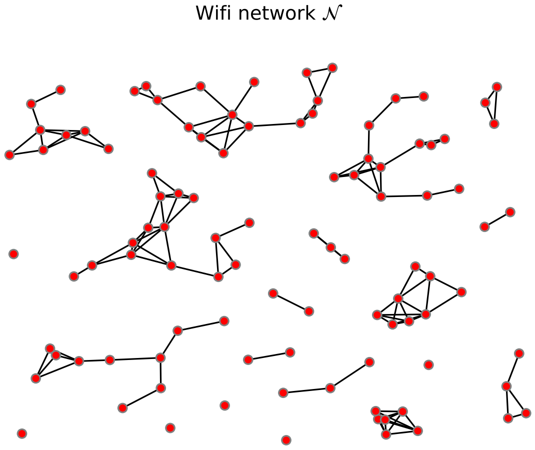


Figure 2.4: Initial condition of $\mathcal{H}$, with all routers using the same frequency band associated with the color red

We observe in Fig. 2.6 and Fig. 2.7 how the potential decreases over time, thanks to the probability distribution of choosing more likely colors with 0 cost associated with it. Because the numerator of Eq. 2.3, represents the unnormalized probability of choosing a specific color, which we plot in Fig. 2.8. We observe how for fixed $\eta(t)$, the probability is always higher for lower new cost, however, for our choice of $\eta(t)$ (Eq. 2.7), we have at time 1000, $\eta(1000) = 10$, hence an unnormalized probability distribution that will almost nullify the probability of choosing a color with a new cost greater than 0, making it unlikely to increase the potential $U(t)$, and limiting the search space of solutions, which is compatible with the concept of reducing the noise since η is its inverse.

In all our results we obtained $\min U(t) = 4$, now we shall discuss why we cannot get any better than that. In this problem, we must consider that the objects of interest are the maximum cliques. But before going into that, let us analyze the minimum potential for different complete graphs given $\mathcal{C}$, with cardinalities from 2 to 6, since 6 is the maximum cardinality clique found in graph $\mathcal{H}$:

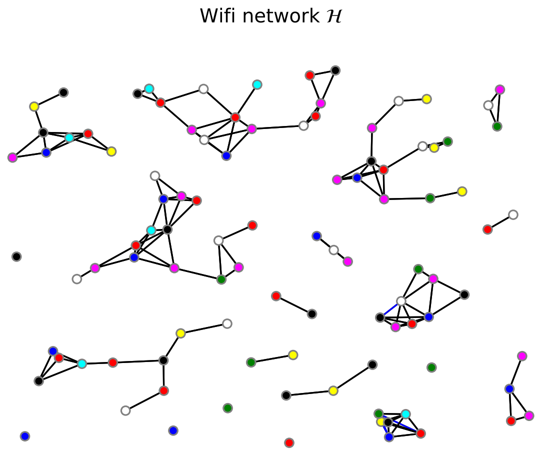


Figure 2.5: Approximate solution of the coloring problem in $\mathcal{H}$, with $U = 4$; edges colored red have cost $c(X_i, X_j) = 2$, colored blue when $c(X_i, X_j) = 1$ or colored black when $c(X_i, X_j) = 0$.

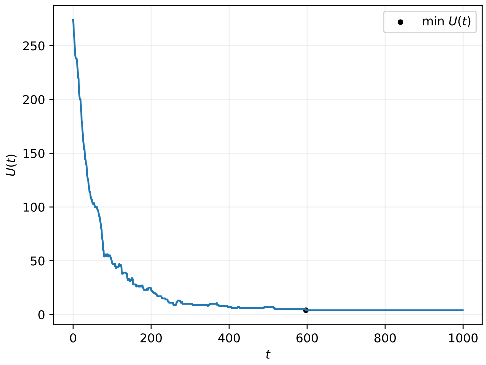
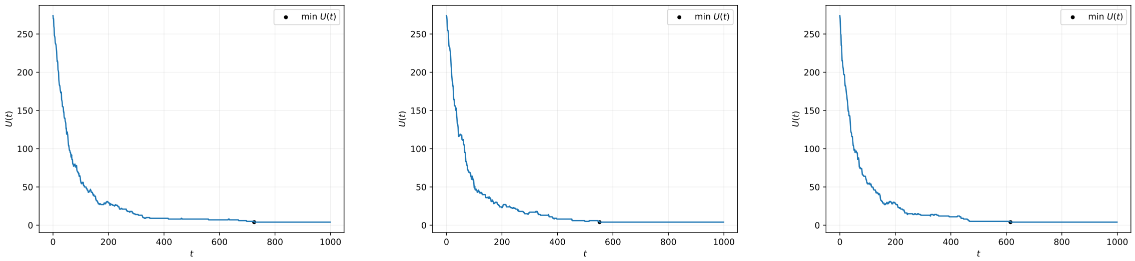


Figure 2.6: Potential of $\mathcal{H}$ in time, $\min U(t) = 4$

In Fig. 2.9 we notice how the minimum potentials of complete graphs of sizes 5 and 6 are, respectively, 1 and 3. One can assume that if there are maximum cliques of size 5 and 6 in a graph, each one would add its potential cost in an optimal solution. But as we will discuss now, such values are but upper bounds of $\min U(t)$, in our particular graph they add exactly such costs, but we will understand why.

To discuss if the costs we found to $\min U_{\mathcal{H}}(t)$ related to the max cliques of size 5 and 6 would contribute as lower bounds, exact values, or upper bounds, we test the interaction among max cliques with potentially additional cost, since a max clique of size smaller than 5, connected to a max clique of size 6 would at most cause it to change the optimal



(a) Potential of run #2 (b) Potential of run #3 (c) Potential of run #4

Figure 2.7: Potential in time of different runs of the algorithm over $\mathcal{H}$, all reached $\min U(t) = 4$, but other simulations reached Nash equilibria with potential equal to 5 or 6

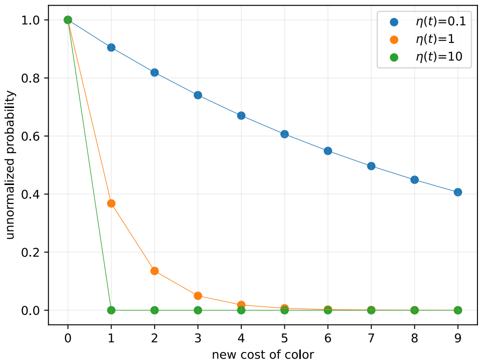


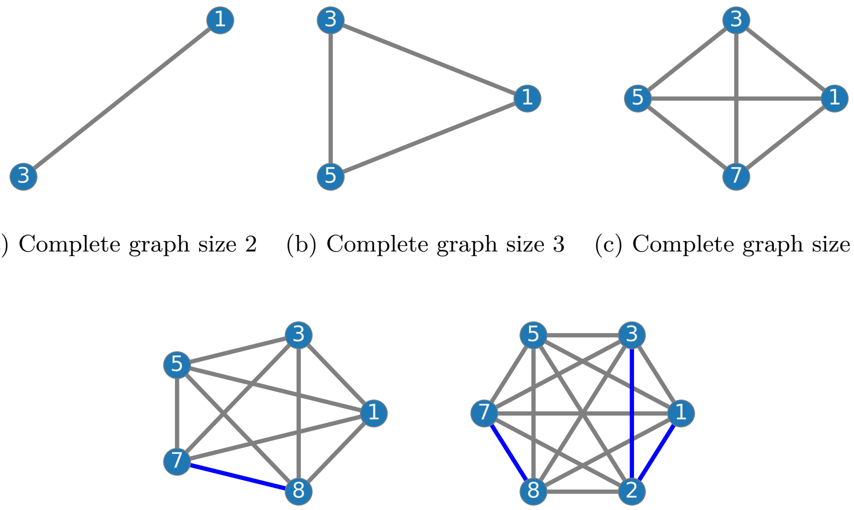
Figure 2.8: Unnormalized probabilities of choosing the new color based on the new cost of the node, for fixed values of $\eta(t)$

solution, likely by swapping colors, or changing *odd colors* with *even colors*, keeping the same cost. Instead, if we have a max clique of size 6 and 5 that have node intersection, the $\min U(t)$ of the graph is no longer the sum of what would be their contributions to the cost ($\min U(t) \neq 1 + 3$), but, optimally, it can reach $\min U(t) = 3$, as we observe in Fig. 2.10, the max cliques share the cost.

We define α_i as the minimum cost (Eq. 2.5) a complete graph of size i assumes, being $\mathcal{Q}$ the set of maximum cliques, and $\beta \in \mathcal{Q}$. For each max clique in any network of size $i = |\beta|$, it can contribute to the $\min U(t)$ as described in Eq. 2.6.

$$\min U_{\mathcal{G}}(t) \leq \sum_{\beta \in \mathcal{Q}} \alpha_{|\beta|} \tag{2.6}$$

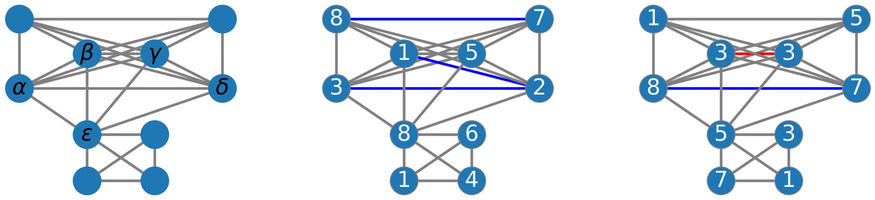
Finally, we can compute our upper bound for the minimum potential, and we obtained



(a) Complete graph size 2 (b) Complete graph size 3 (c) Complete graph size 4

(d) Complete graph size 5 (e) Complete graph size 6

Figure 2.9: Complete graphs with different sizes. Blue edges have cost 1. Given $|\mathcal{C}| = 8$ we have (a)(b)(c) with zero cost, (d) with cost 1 and (e) with cost 3. Numbers in white represent the color a node would assume to minimize costs.



(a) Test graph (b) Solution 1 (c) Solution 2

Figure 2.10: (a) Test graph with size 6 max clique above, a max clique of size 4 below, and a max clique of size 5 connecting the nodes belonging to the previous cliques $[\alpha, \beta, \gamma, \delta, \epsilon]$. (b)(c) Blue edges have cost 1, and red edges have cost 2. Numbers in white represent the color a node would assume to minimize costs. Both solutions found $\min U(t) = 3$

4:

$$\min U_{\mathcal{H}}(t) \leq \sum_{\beta \in \mathcal{Q}} \alpha_{|\beta|} = \alpha_5 + \alpha_6 = 1 + 3 = 4$$

Observing Fig. 2.5, the two max cliques of size 5 and 6 do not share an intersection of nodes, hence the upper bound is the exact minimum value that the potential may reach because they act as the complete graphs in Fig. 2.9. Notice that our algorithm may reach Nash equilibria with a potential higher than 4 because when $\eta(t)$ is large, it behaves similarly to a pure Best Response, and it may reach local minima solutions, as we will discuss in the next section.

2.3 Part (c)

Evaluate what happens for different choices of $\eta(t)$, i.e., constant (with small and large values), or other increasing functions $\eta(t)$ etc.

Our algorithm makes use of NBR dynamics, because our nodes will likely choose an action that optimizes heuristically the cost, but not with probability 1, however, when $eta(t)$ gets large, it behaves as so. Since $\eta(t)$ represents the inverse of the noise.

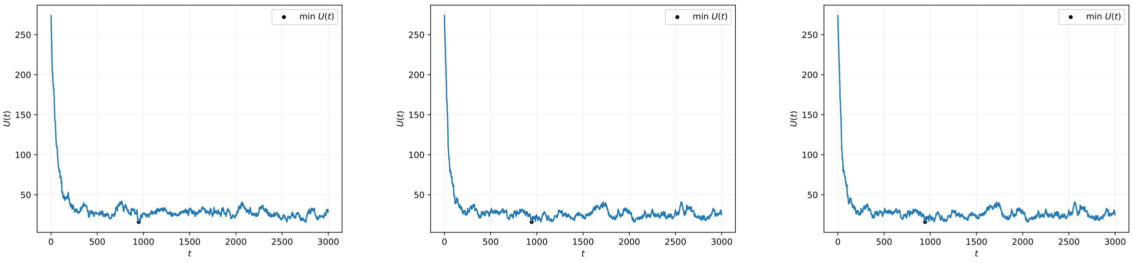
We will now test four different functions for $\eta(t)$.

2.3.1 Constant $\eta = 1$

First, we test our algorithm with $\eta(t)$ constantly equal to 1:

$$\eta(t) = 1 \tag{2.7}$$

This means that our algorithm will run with relatively high noise, with respect to our previous choice of $\eta(t)$ (Fig. 2.8). Indeed, we get from our results that the potential oscillates with high noise (Fig. 2.11), never reaching the optimal solution of potential equal to 4.



(a) Potential of run #1

(b) Potential of run #2

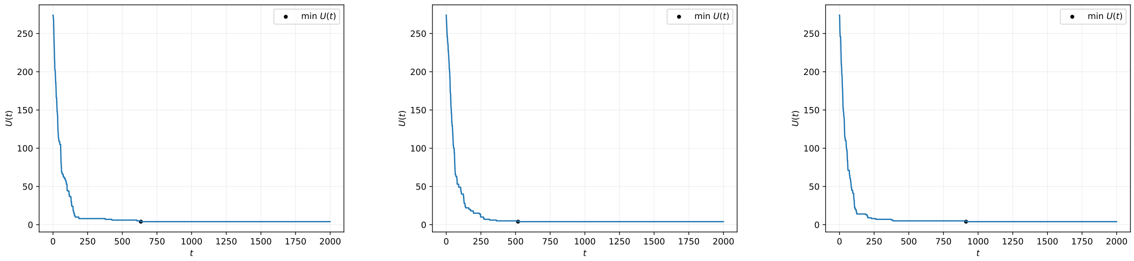
(c) Potential of run #3

Figure 2.11: Potential in time of different runs of the algorithm over $\mathcal{H}$ with $\eta(t) = 1$. (a) $\min U(t) = 16$, (b) $\min U(t) = 16$, (c) $\min U(t) = 15$.

2.3.2 Constant $\eta = 200$

On the opposite of the previous constant, 200 makes so as our algorithm behaves similarly to Best Response, hence we notice a straight descent and a straight plateau on $U(t) = 4$. on all simulations, for it was in the Nash equilibrium (Fig. 2.12).

$$\eta(t) = 200 \tag{2.8}$$



(a) Potential of run #1

(b) Potential of run #2

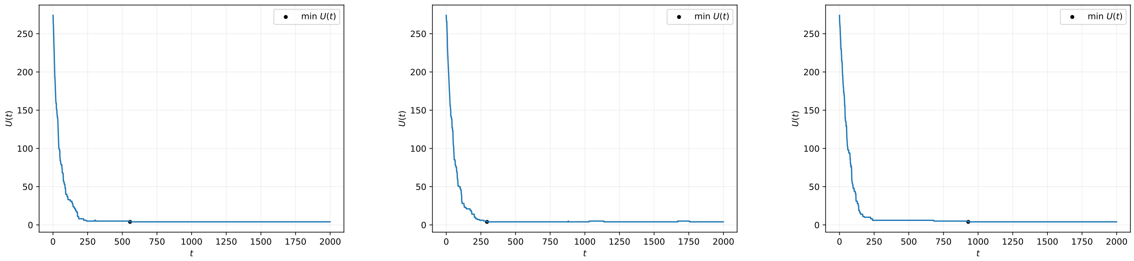
(c) Potential of run #3

Figure 2.12: Potential in time of different runs of the algorithm over $\mathcal{H}$ with $\eta(t) = 200$. All reached $\min U(t) = 4$.

2.3.3 Logarithmic

Using an increasing function of $\eta(t)$, but increases very slowly, mostly for large t, therefore keeping a reasonable amount of noise to avoid getting stuck to local minima solutions (Fig. 2.13),

$$\eta(t) = \ln t \tag{2.9}$$



(a) Potential of run #1

(b) Potential of run #2

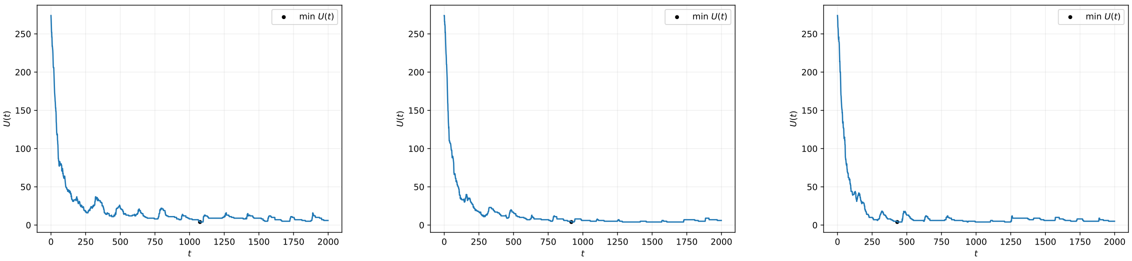
(c) Potential of run #3

Figure 2.13: Potential in time of different runs of the algorithm over $\mathcal{H}$ with $\eta(t) = \ln t$. All reached $\min U(t) = 4$.

2.3.4 Sine behavior

In our case, we have reached optimal solutions many times, but for more complex graphs, it is possible that finding local minima is more of an issue. In such cases, a valuable metaheuristic is the Iterated Local Search (ILS), on which we will base our choice of $\eta(t)$ to behave similarly, specifically on the sin part of the formula, and the increasing of the amplitude of the sine will reduce noise over time, except for when t/τ_1 is close to multiples of π, where $|\sin(k\pi)| = 0 \Rightarrow \eta = 0$, hence the noise is maximum, and the choice of the new color is uniform, independently of the new cost (Eq. 2.3). The choice of parameters τ_1 and τ_2 depends on the problem, in Fig. 2.14 we plot three combinations of parameters, all runs reached $\min U(t) = 4$.

$$\eta(t) = \frac{t}{\tau_1} |\sin \frac{t}{\tau_2}| \tag{2.10}$$



(a) $\tau_1 = 100$, $\tau_2 = 50$

(b) $\tau_1 = 50$, $\tau_2 = 100$

(c) $\tau_1 = 50$, $\tau_2 = 50$

Figure 2.14: Potential in time of different runs of the algorithm over $\mathcal{H}$ with ILS-like algorithm. All reached $\min U(t) = 4$.