

**Activity A4 : Epidemic spreading on complex
networks**
Hajar Lachheb

1 Introduction

The spread of infectious diseases has always been an important topic in epidemiology, and in recent years, the use of network theory has become a valuable tool in the study of disease dynamics. In this report, we present the results of our Monte Carlo simulation of an epidemic spreading dynamics in complex networks, using the SIS model. This model represents each node in a network as an individual, which can be in two possible states: Susceptible (S), i.e., healthy but can get infected, or Infected (I), i.e., has the disease and can spread it to its neighbors.

The aim of our simulation is to calculate the fraction of infected nodes, i , in the stationary state, as a function of the infection probability of the disease b , for different values of the recovery probability u . We have tried different undirected networks, such as Erdős-Rényi, scale-free, and real networks, with different sizes, average degrees, and exponents.

Additionally, we have compared the results of our Monte Carlo simulations with the theoretical predictions provided by the Microscopic Markov Chain Approach (MMCA) model, which is an analytical method used to describe the dynamics of epidemics in networks.

In the following sections, we will present a detailed explanation of the software used, the decisions made, and the results obtained from our simulations. We will also provide plots of i for different parameter combinations.

2 Work Description

2.1 Description of the functions used

In this task, it was very exciting to learn more about how to model the spread of epidemic diseases especially after the latest events related to the global pandemic situation and virus COVID19. It indeed helped me learn about the spread of epidemics in complex networks and how it can be modeled using the SIS (Susceptible-Infected-Susceptible) model. So to begin with, i had to develop the SIS Model following the documentation given and searching online for ressources.

Before explaining the work done in every function, we need to mention that we mainly used NetworkX as a software for our work.

The purpose of my function SIS, is to The purpose of this code is to simulate the spreading of the epidemic in a network. The function needs three parameters. First of all, we have the network G , then the infection probability b , and finally the recovery probability u .

The network's nodes are initially put into a Susceptible state, which then randomly infects selected nodes based on the specified infection probability b . The simulation is then run for a predetermined amount of time, with each node's state being updated at each timestep based on its present state and the states of its neighbor nodes. The simulation then moves forward by repeatedly iterating over the nodes and their neighbors; if the neighbor is infected, the node has a chance of becoming infected based on the provided infection probability b . Additionally, based on the provided recovery probability u , the node has a chance of recovering if it has already contracted the infection. Finally, the simulation function calculates the percentage of infected nodes in the network in the stationary state based on the number of infected nodes, and a parameter was also added so as to plot the percentage of infected nodes over time.

Then, using the function SIS, we were able to simulate the dynamics of an epidemic spreading, in which each node could be in one of two possible states: susceptible (S), i.e., healthy but susceptible to infection; or infected (I), i.e., carrying the disease and capable of spreading it to nearby nodes. In order to estimate the average fraction of infected nodes in the stationary state during numerous iterations (Nrep) of the SIS function with the provided parameters u and b , we had to build the function MC.

The graph G , the number of repetitions Nrep (100 or 50 if 100 is too many), the infection probability u , and the recovery probability b are the parameters that the MC function was developed to take into account. After that, the function determines the percentage of infected nodes p in the stationary state for various values of the disease's infection probability (b) and its recovery probability (u). Our code returns a dictionary that contains the values of p for various values of u , which is used to plot the results using the matplotlib library.

Finally and to be able to compare between the results of the MC and MMCA, we have to develop a MMCA function and then plot a figure to show the difference in the prediction. The MMCA was developed again using some online documentation around the Microscopic Markov Chain Approach. The function MMCA was created to calculate the theoretical prediction for the fraction of infected nodes in the stationary state. This model takes into consideration the likelihood that an infected node would recover from the illness and the likelihood that susceptible nodes will get the disease from their diseased neighbors.

Then, we created the MMCAcombinations function. In fact, the importance of this function is highlighted in the fact that we will be able to plot different combinations and compare the results. In fact, we will be able to create a dictionary of theoretical predictions for the percentage of infected nodes for any combination of u and given a complex network G and a list of recovery probability u list.

To end with the functions we developed, we added a plot comparison function that will help us plot the comparison between the MC and MMCA predictions for each network of our choosing.

2.2 Plotting the networks

Before plotting the different networks we worked on. We need to start by emphasizing the fact that we mainly worked on three different types of networks: ER, SF, and real networks.

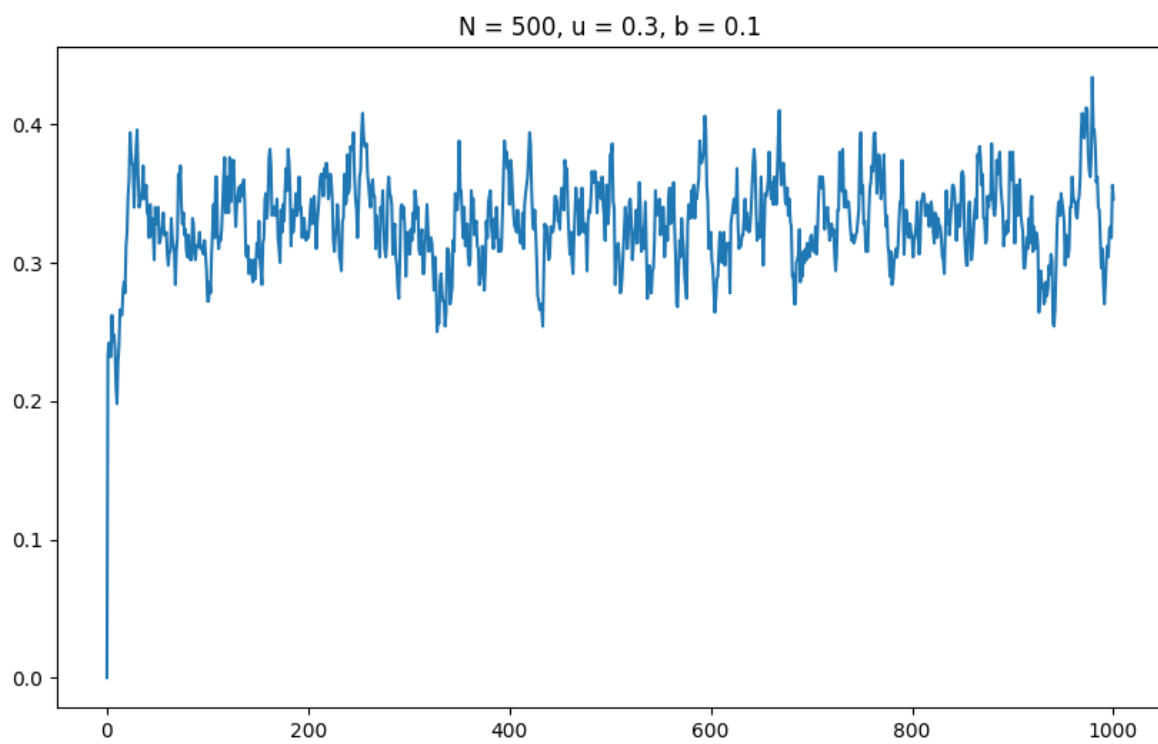
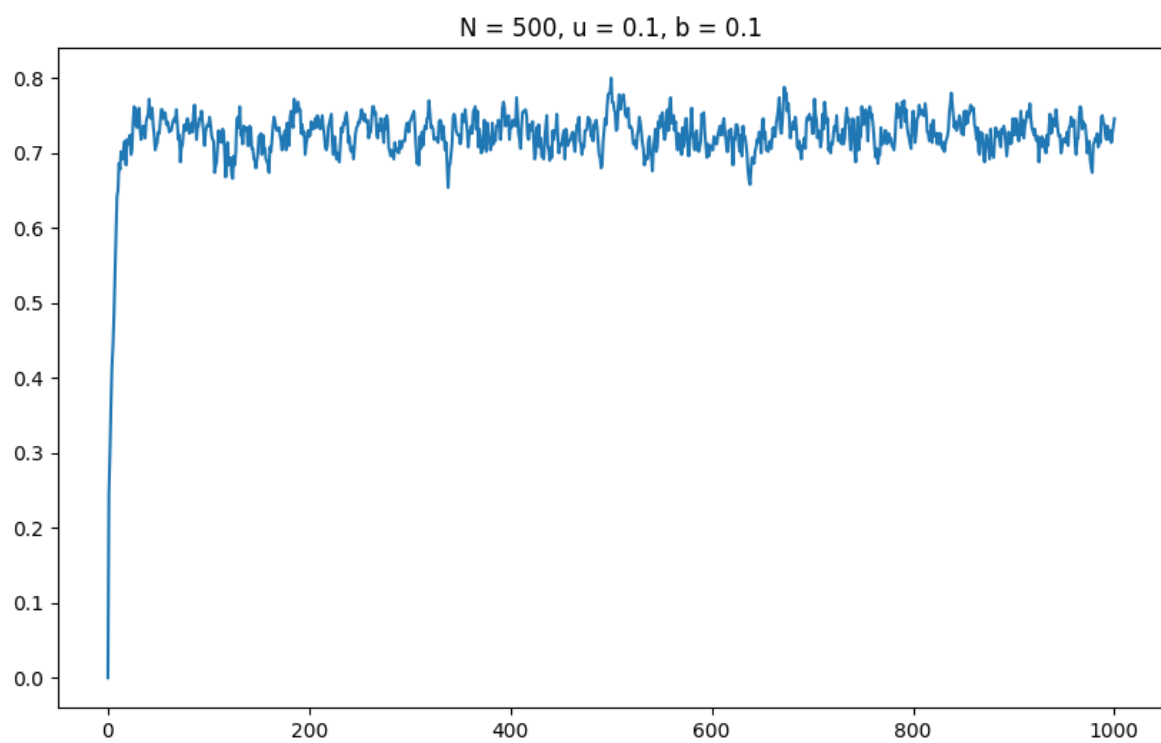
For the ER and SF networks, we mainly used the same functions we implemented in Activity 2.

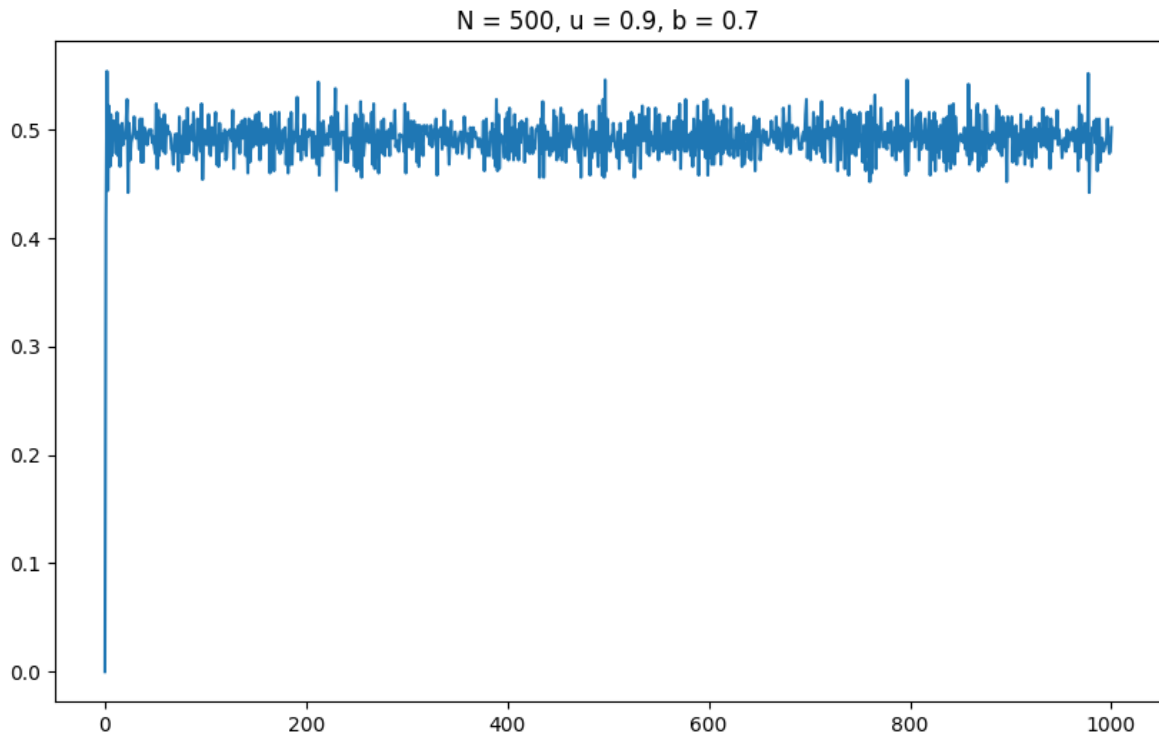
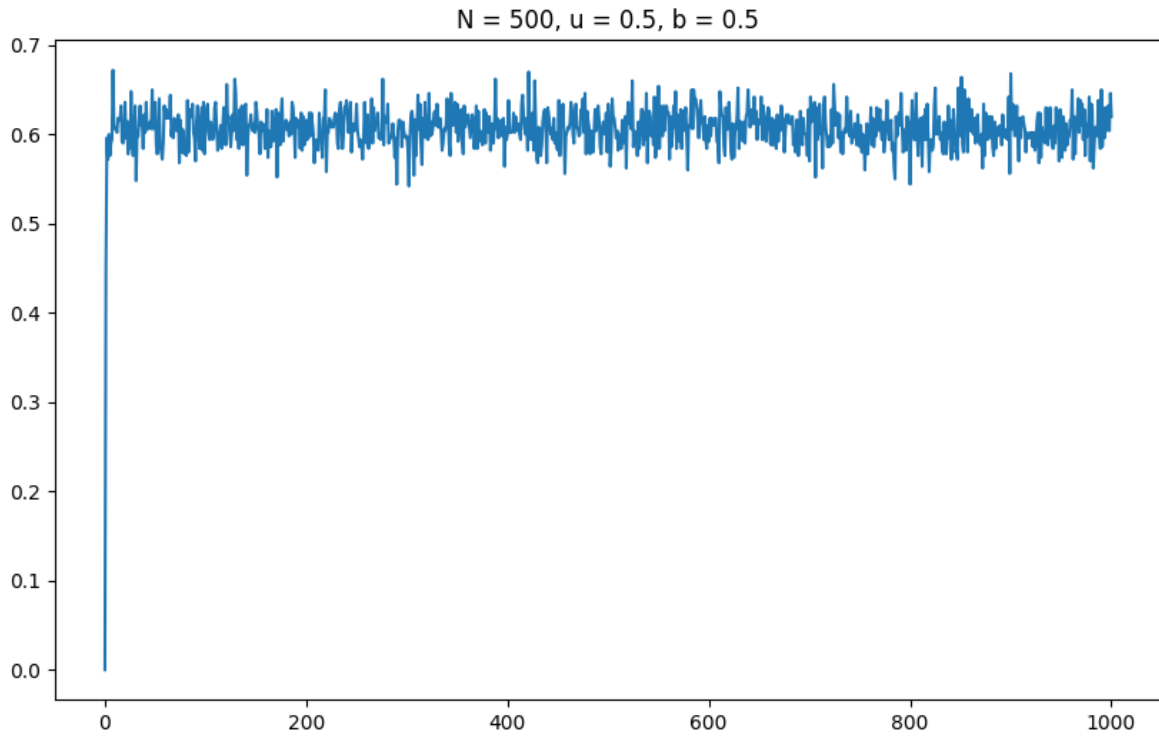
We also need to mention that the Microscopic Markov Chain Approach (MMCA) model's theoretical predictions were compared with the results of each network's Monte Carlo simulation. Some simulations were run for 100 repetitions to reduce computational time, whereas others were conducted for 50 because 100 would have been too long. Before starting the whole simulation, we plotted the simulated evolution of the fraction of infected nodes over time for several b values to know approximately when the transition stage came to an end and the stationary state was reached. According to these results, T_{trans} and T_{max} were changed.

Parameters used here : $N = 500$, Average Degree = 5, $T_{max} = 1000$, $T_{trans} = 900$ and u list = 0.1, 0.3, 0.5, 0.7, 0.9 and $b = 0.1, 0.3, 0.5, 0.7, 0.9$.

The combination plots can be found in the notebook, we have only shared four plots to show the results.

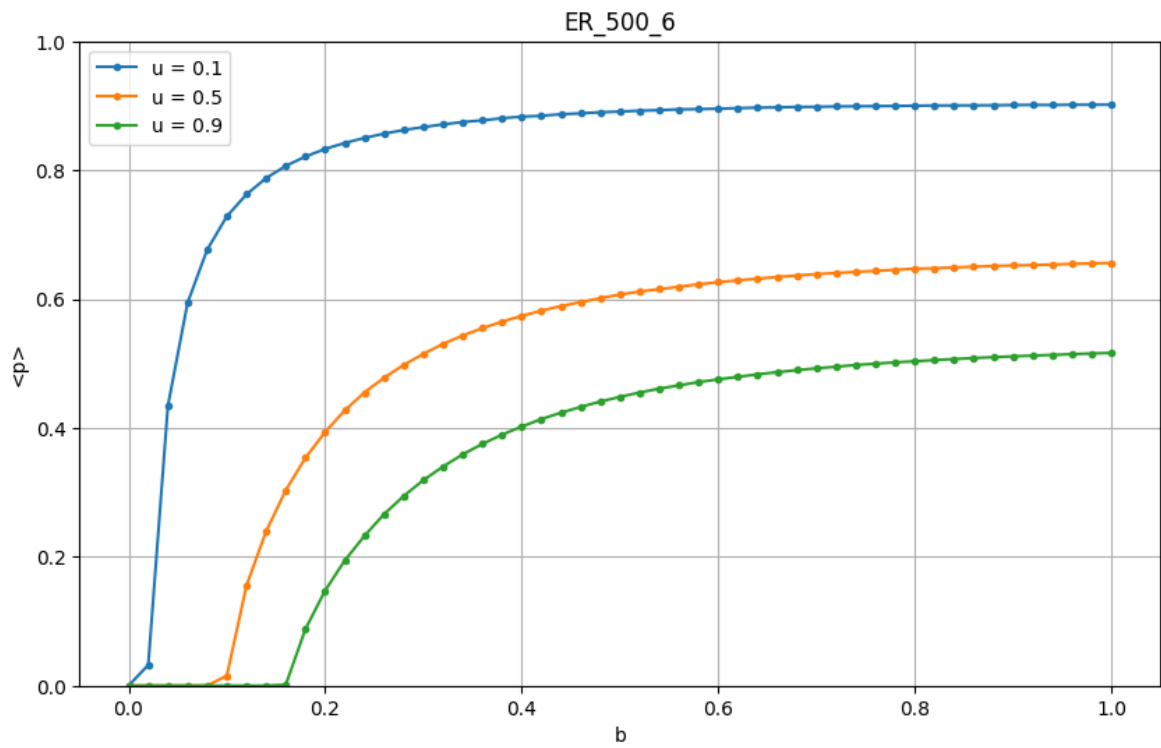
Combination plots of the network ER500



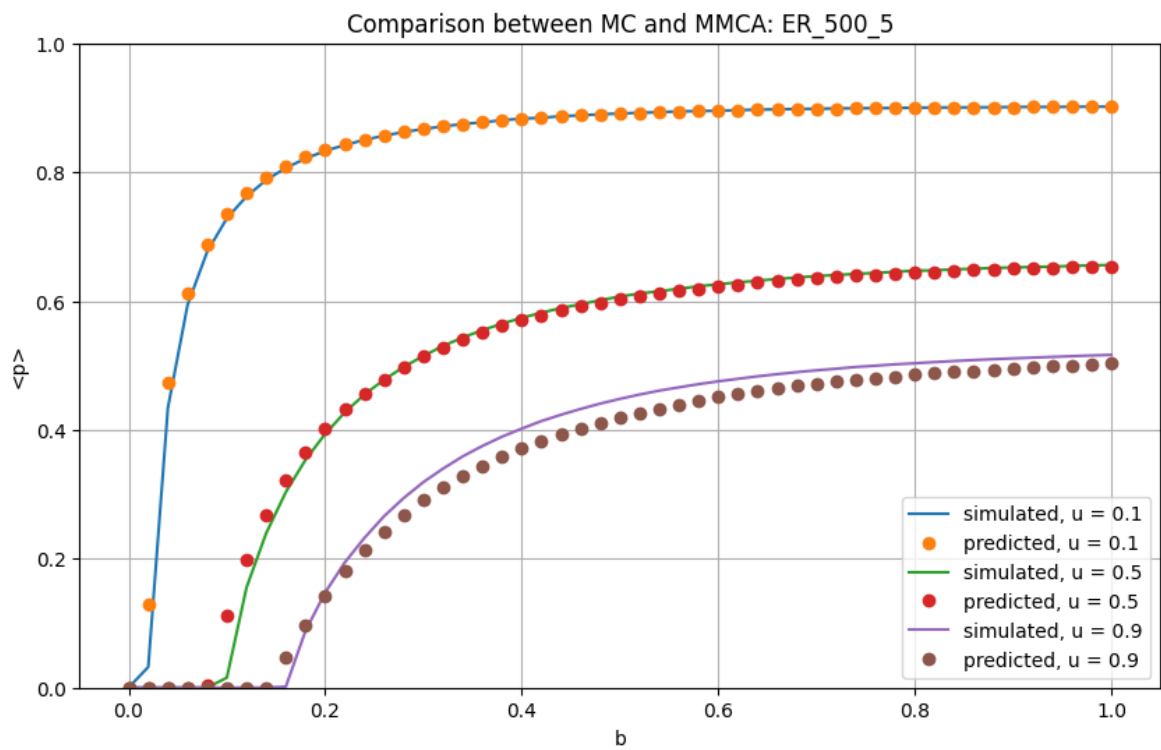


To reduce the computational cost, the ulist became 0.1, 0.5 and 0.9. And we choose a $T_{trans} = 200$ since the stationary state starts before reaching 200. So we estimated it.

Plot of the network ER5005 after choosing the adequate Ttrans



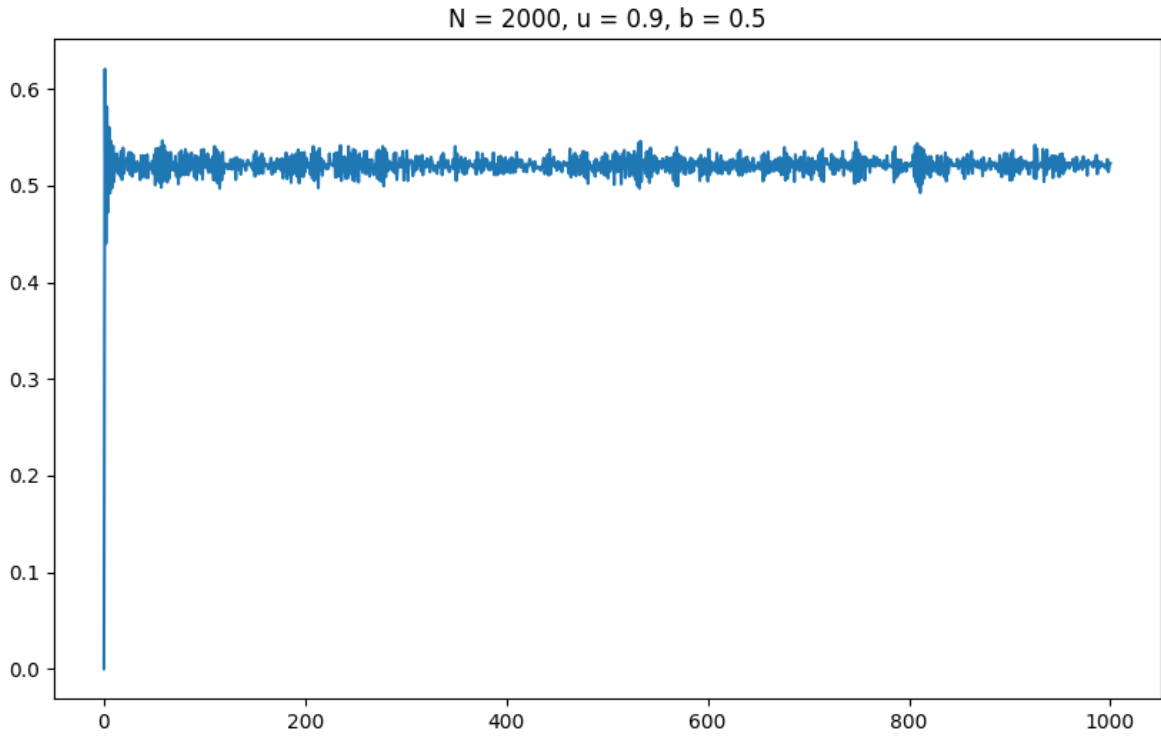
Plot of comparison MC MMCA for the network ER5005

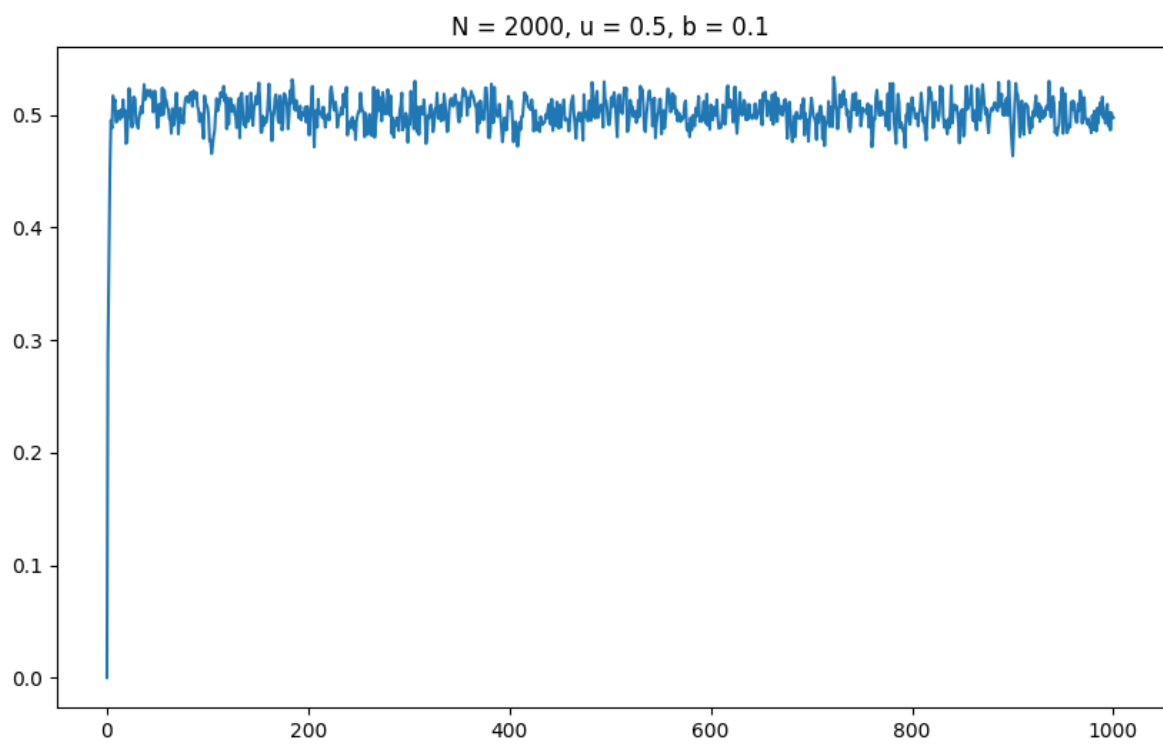
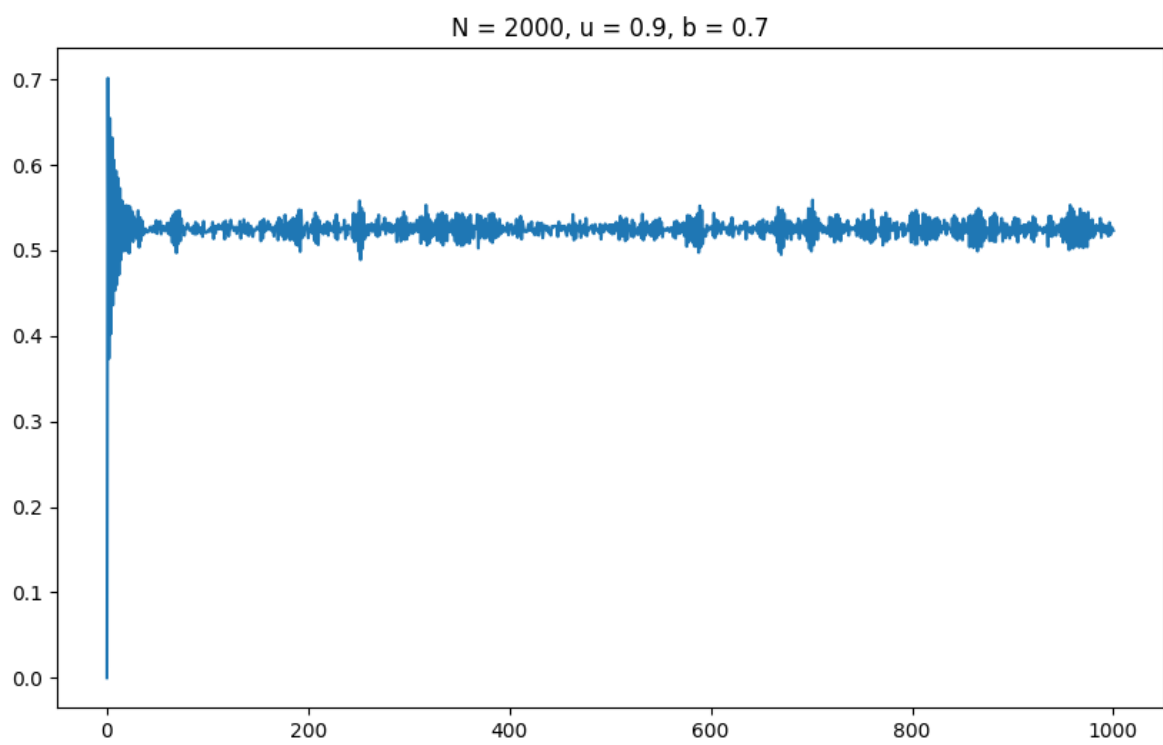


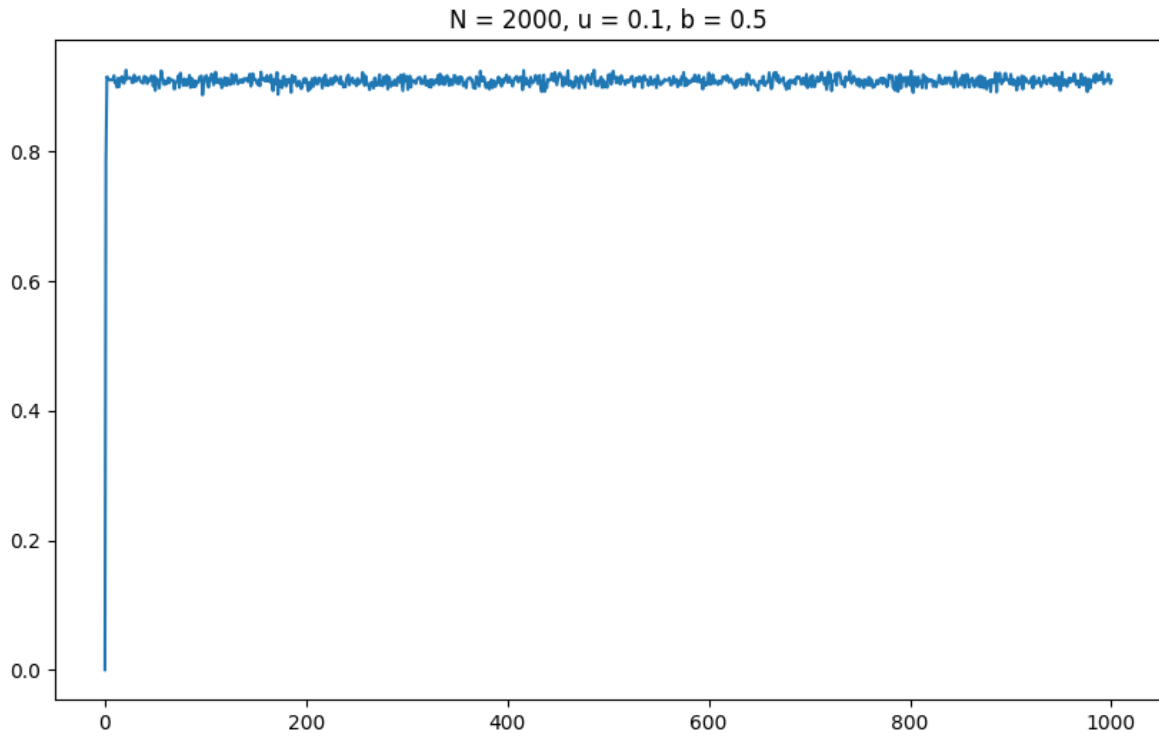
Parameters used here : $N = 2000$, Average Degree = 14, $T_{\max} = 1000$, $T_{\text{trans}} = 900$ and $u_{\text{list}} = 0.1, 0.3, 0.5, 0.7, 0.9$ and $b = 0.1, 0.3, 0.5, 0.7, 0.9$.

The combination plots can be found in the notebook, we have only shared four plots to show the results.

Combination plots of the network ER200014

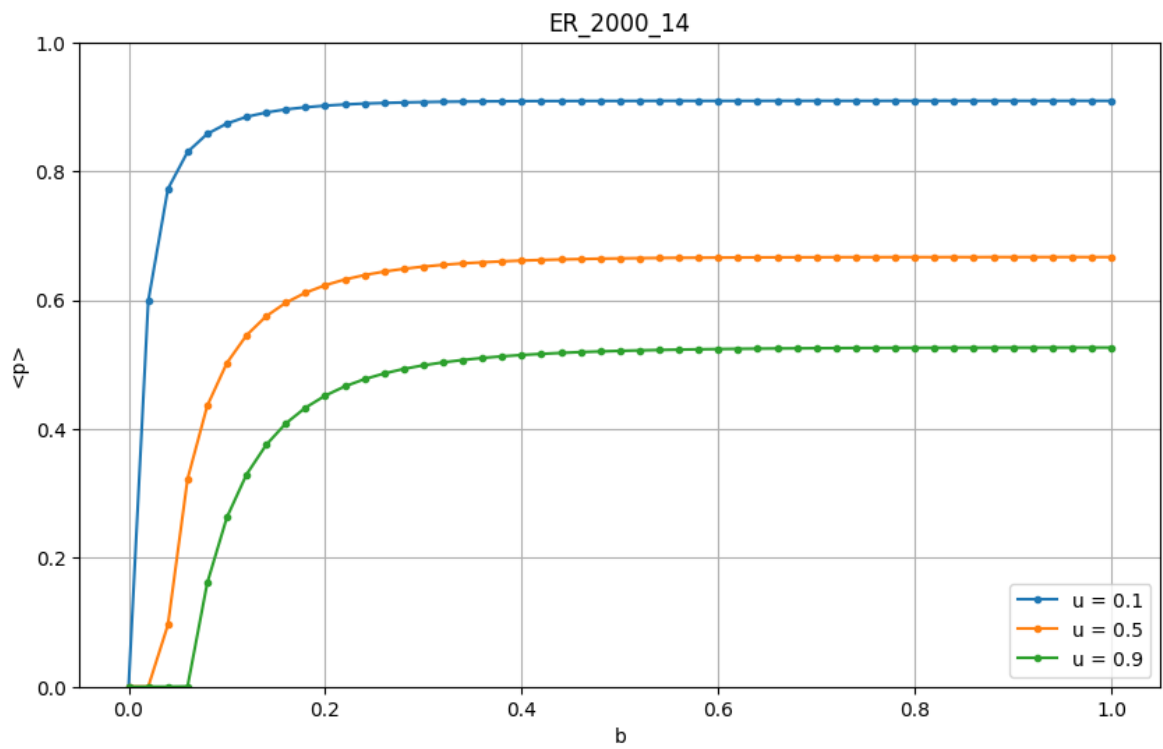




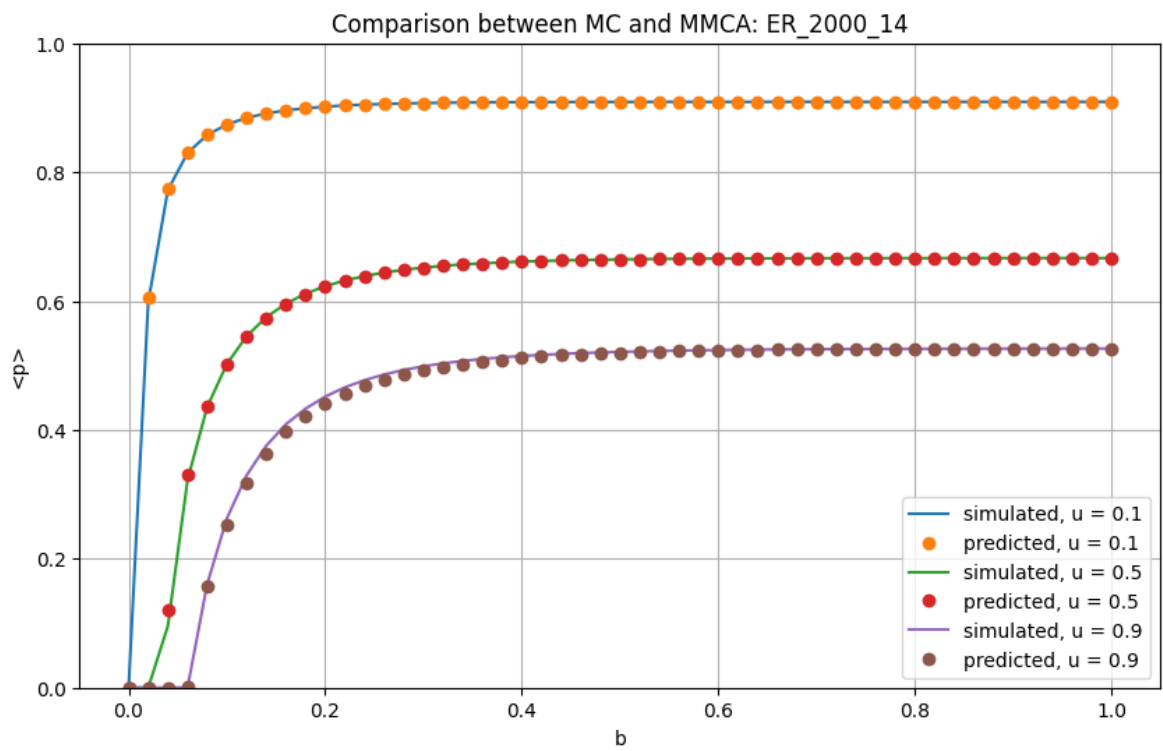


To reduce the computational cost, the ulist became 0.1, 0.5 and 0.9. And we choose a $T_{trans} = 100$ since the stationary state starts before reaching 100. So we estimated it.

Plot of the network ER200014 after choosing the adequate T_{trans}

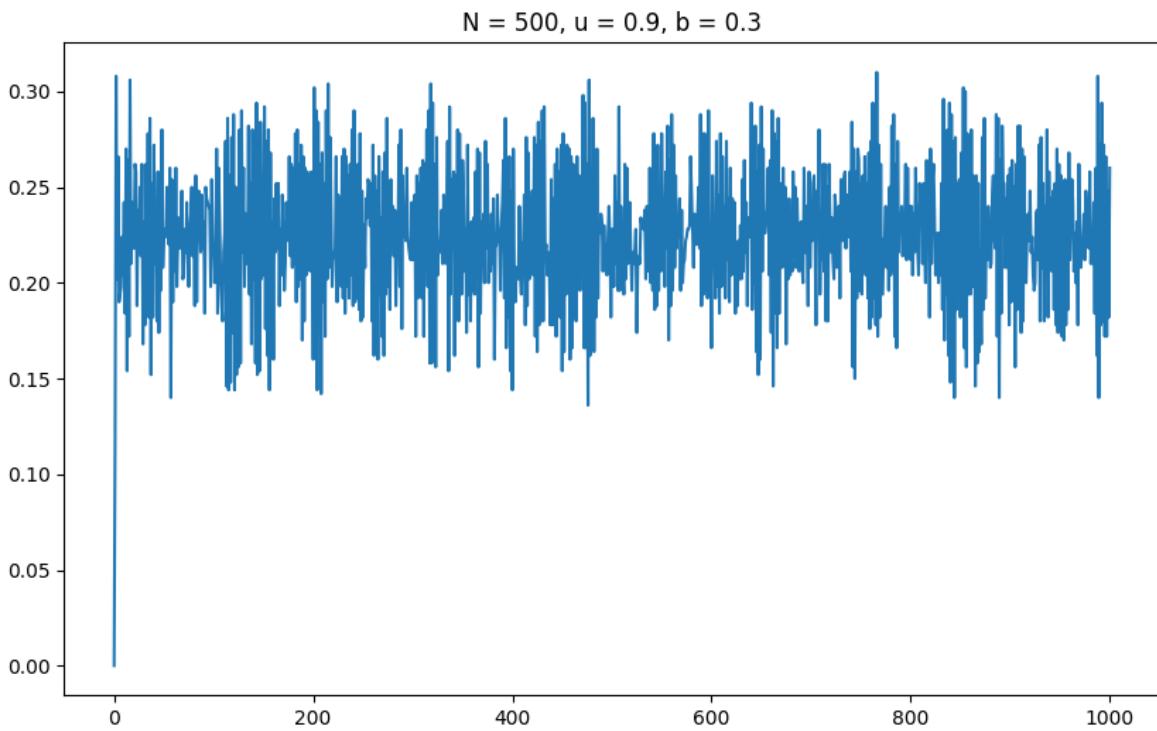
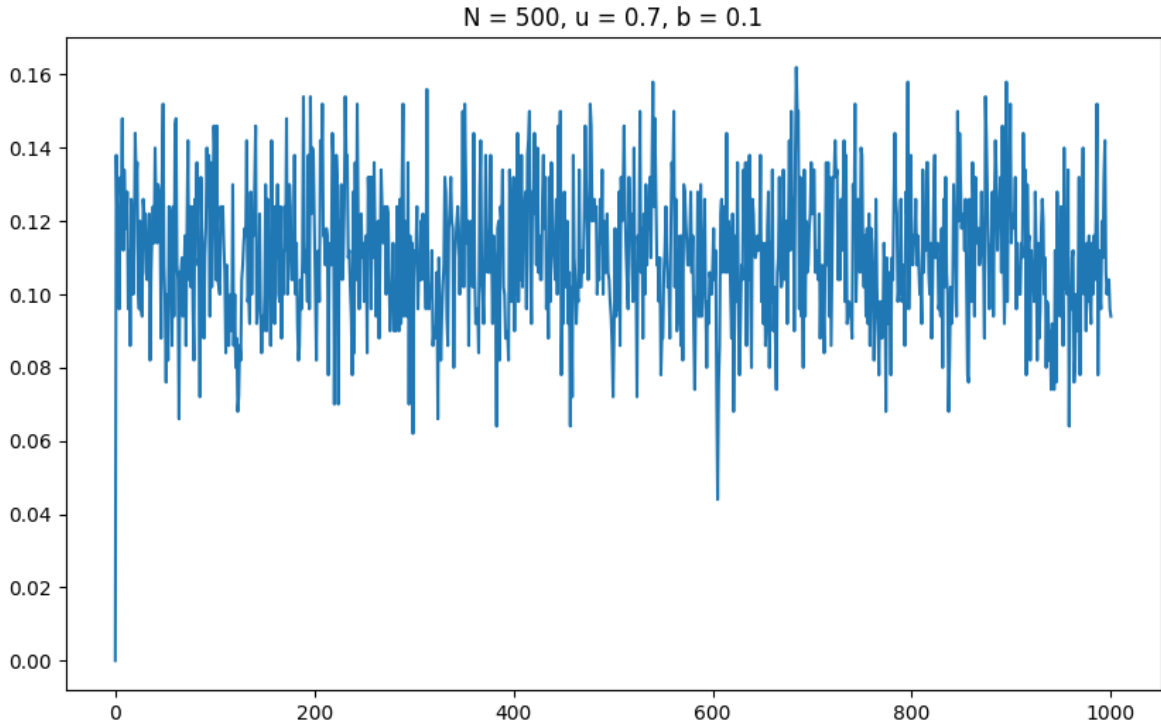


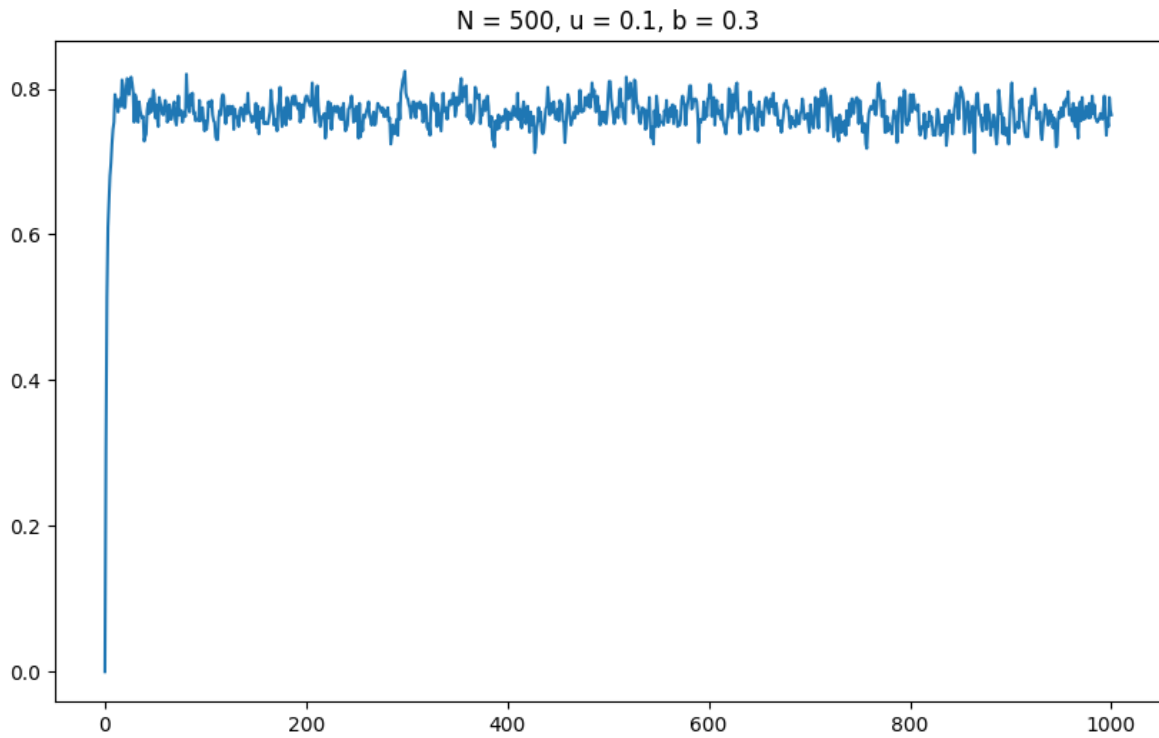
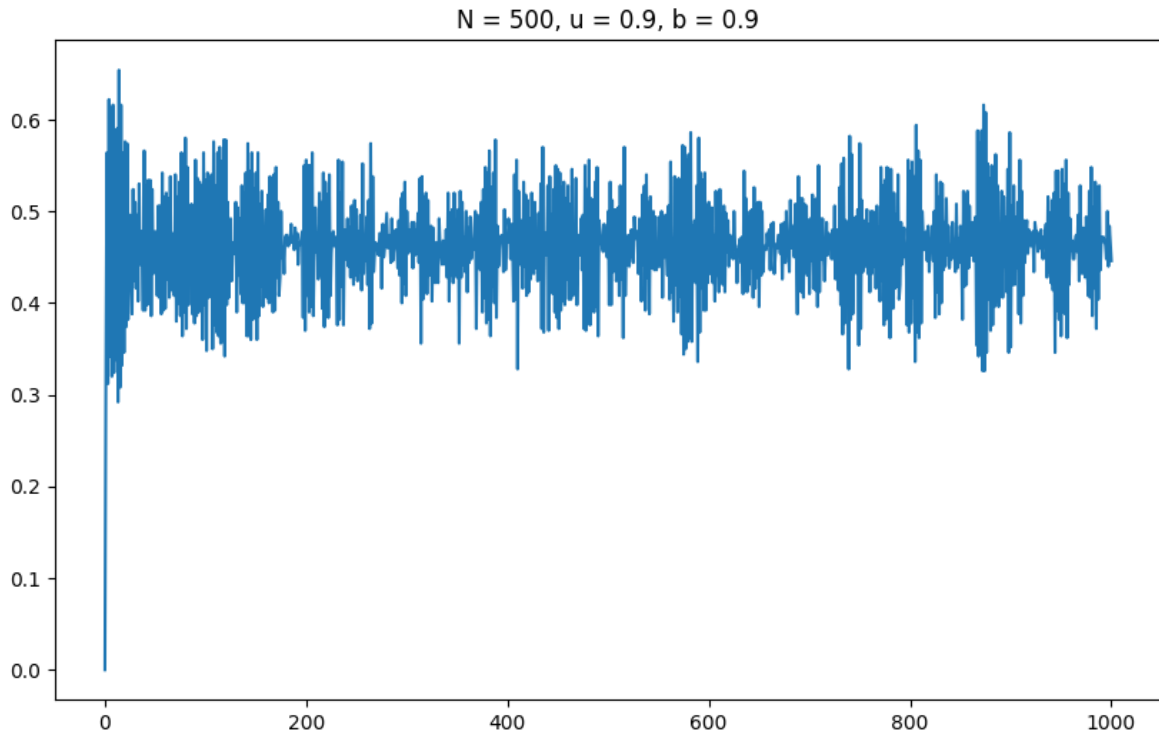
Plot of comparison MC MMCA for the network ER200014



Parameters used here : $N = 500$, $\text{Gamma} = 2.1$, $T_{\text{max}} = 1000$, $T_{\text{trans}} = 900$ and $u_{\text{list}} = 0.1, 0.3, 0.5, 0.7, 0.9$ and $b = 0.1, 0.3, 0.5, 0.7, 0.9$.
The combination plots can be found in the notebook, we have only shared four plots to show the results.

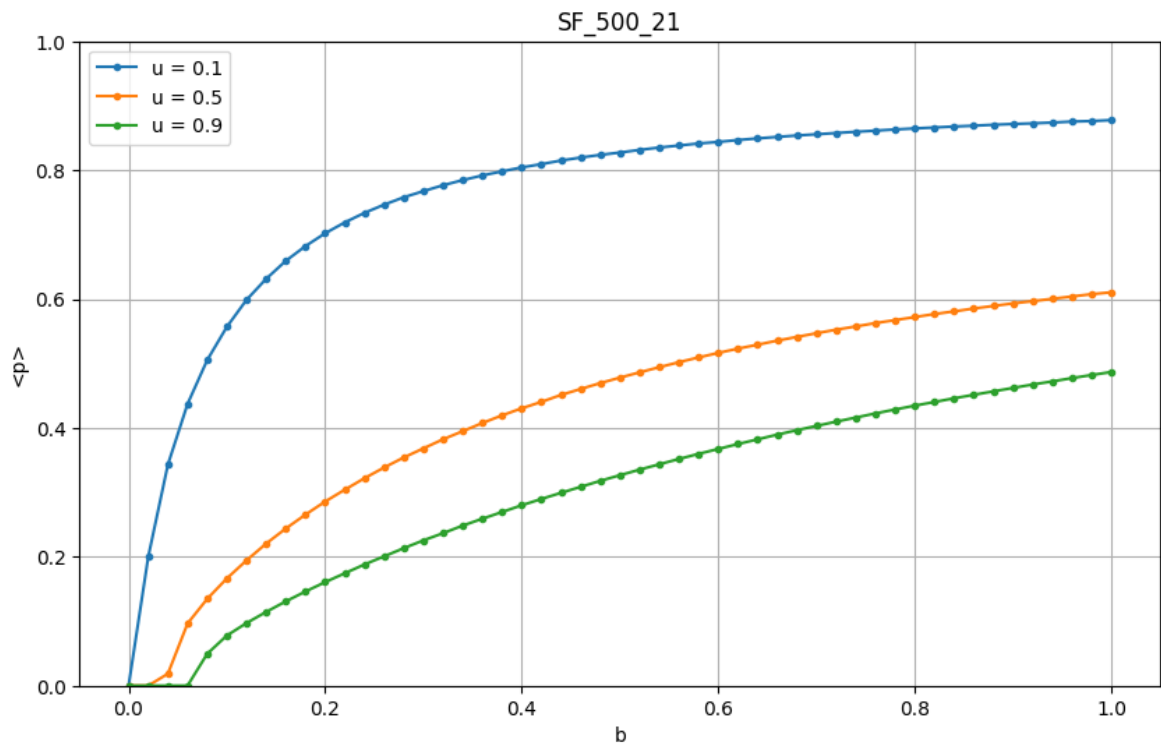
Combination plots of the network CMSF50021



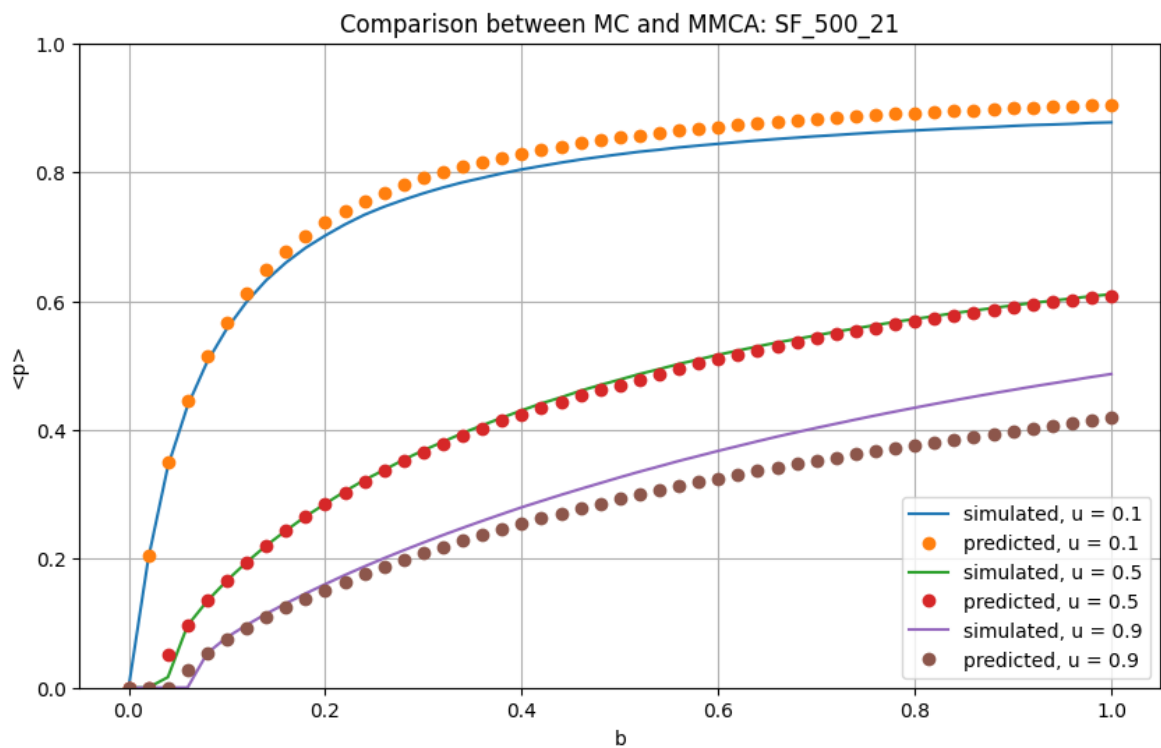


To reduce the computational cost, the ulist became 0.1, 0.5 and 0.9. And we choose a $T_{\text{trans}} = 200$ since the stationary state starts before reaching 200. So we estimated it.

Plot of the network CMSF50021 after choosing the adequate Ttrans



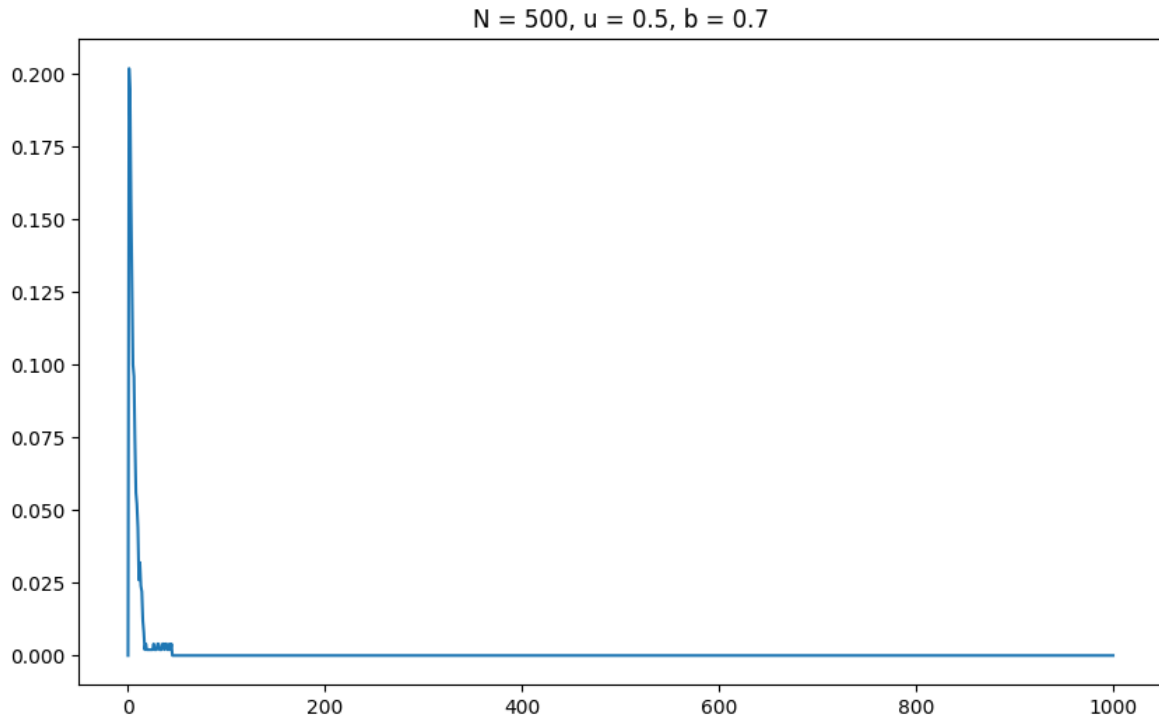
Plot of comparison MC MMCA for the network CMSF50021

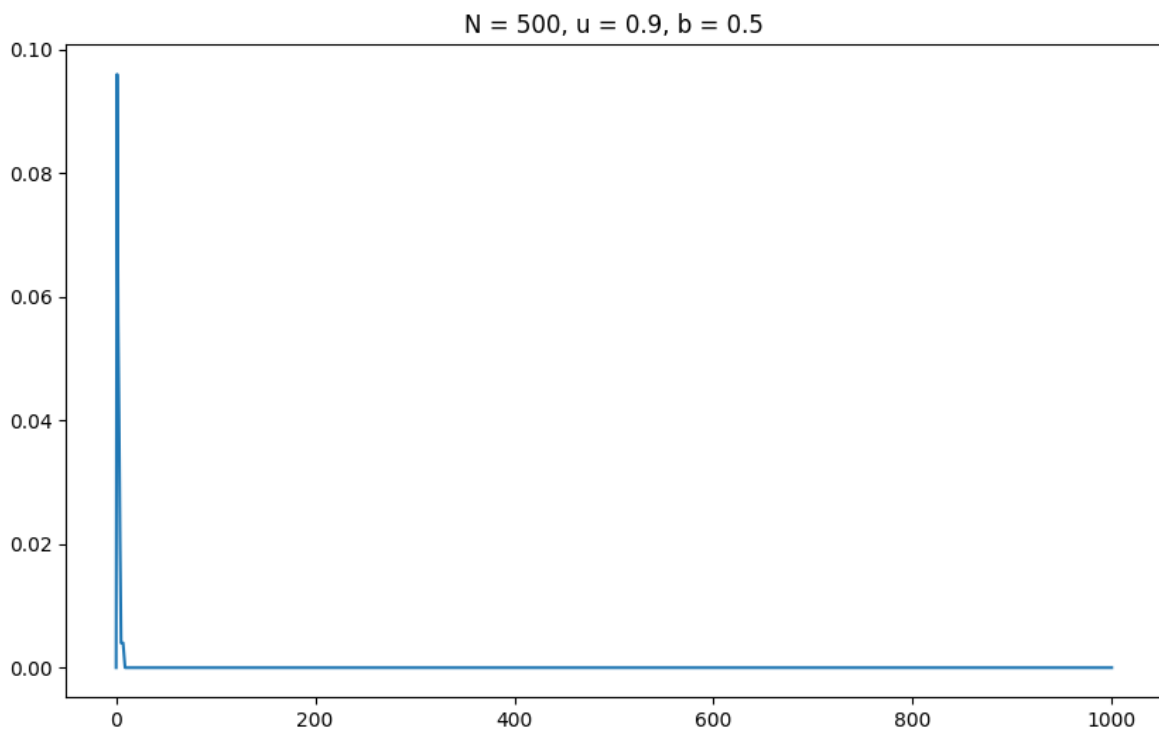
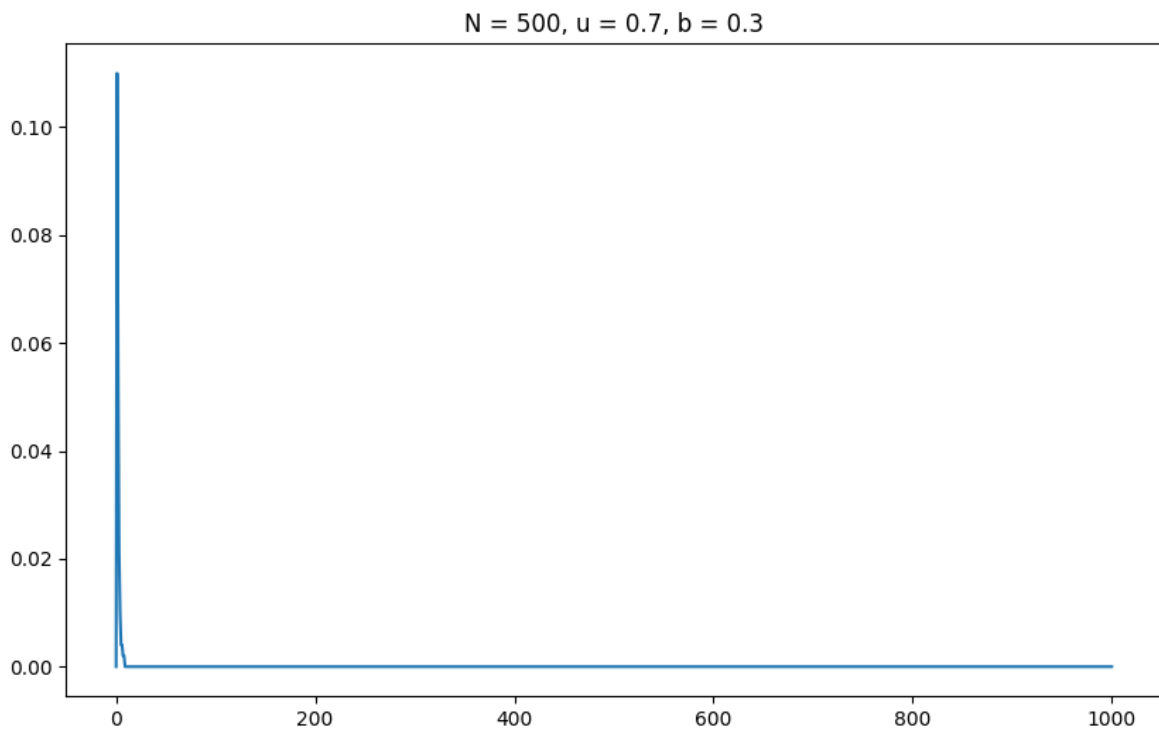


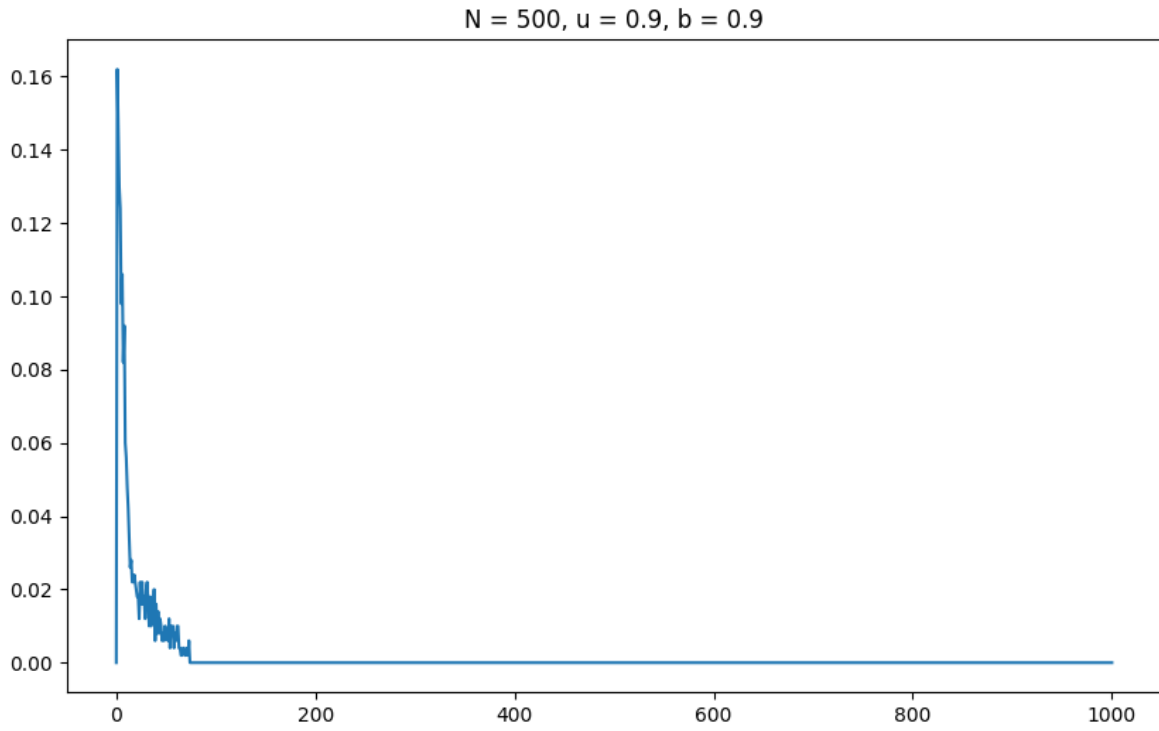
Parameters used here : $N = 500$, $\text{Gamma} = 5.5$, $T_{\text{max}} = 1000$, $T_{\text{trans}} = 900$ and $u_{\text{list}} = 0.1, 0.3, 0.5, 0.7, 0.9$ and $b = 0.1, 0.3, 0.5, 0.7, 0.9$.

The combination plots can be found in the notebook, we have only shared four plots to show the results.

Combination plots of the network CMSF50055

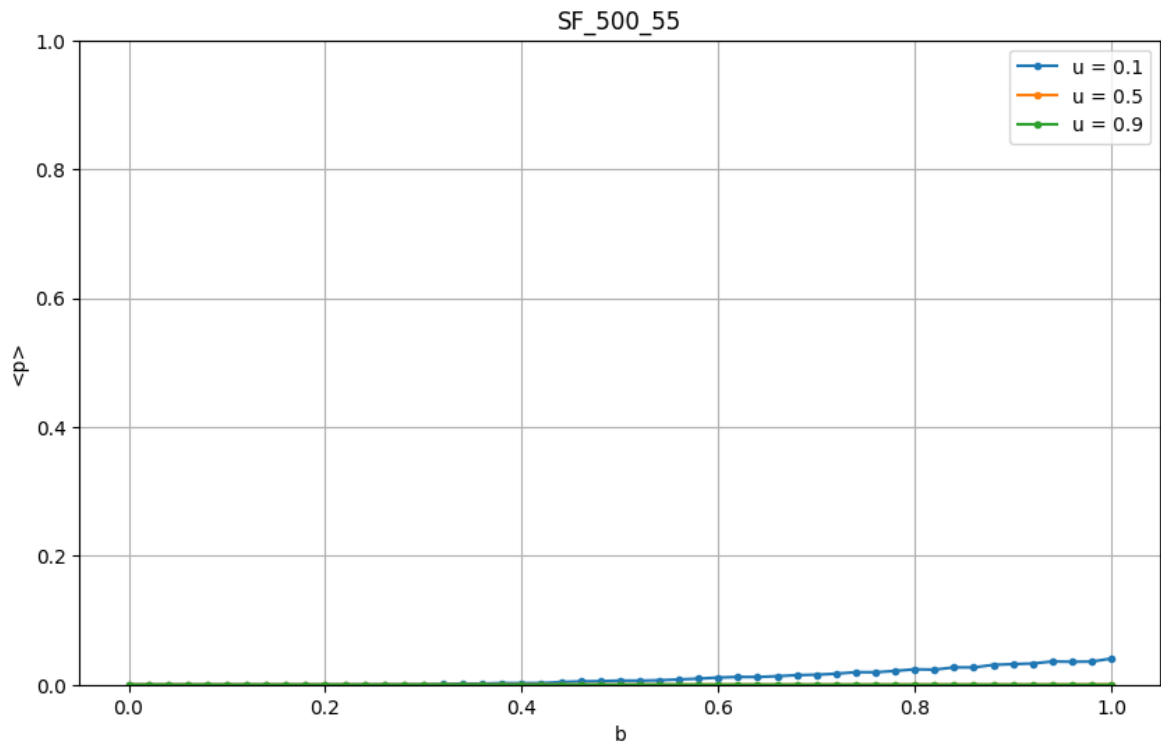




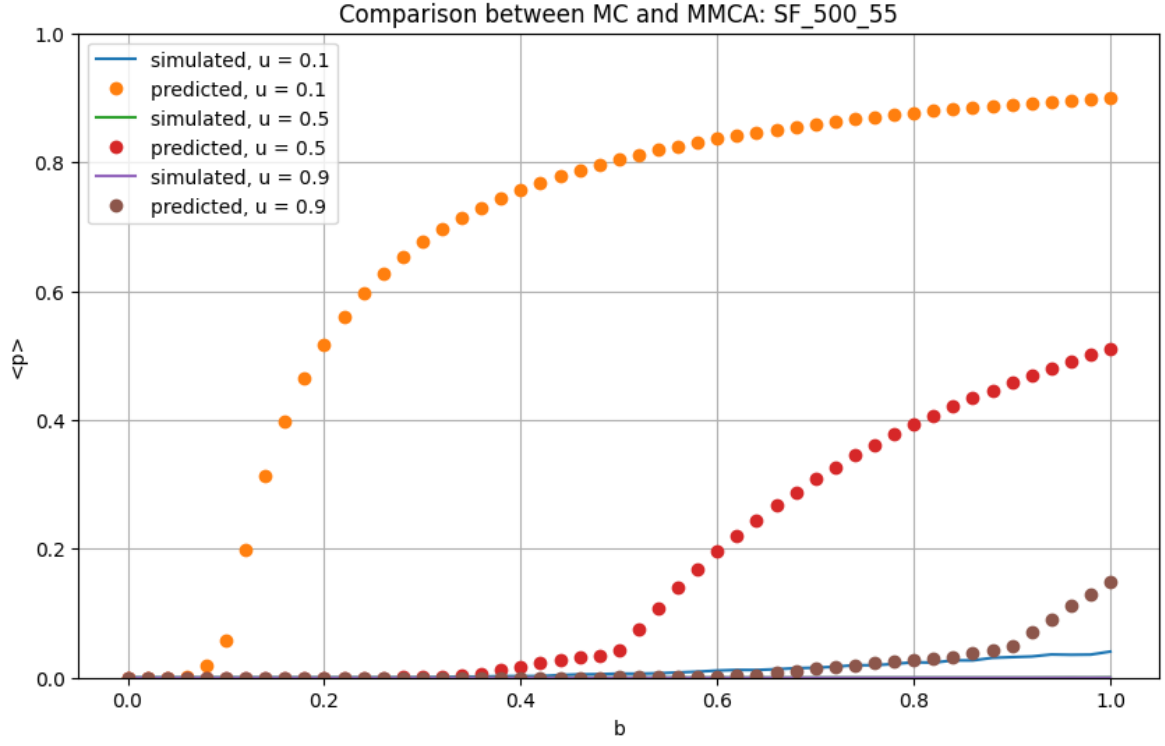


To reduce the computational cost, the ulist became 0.1, 0.5 and 0.9. And we kept a $T_{trans} = 900$.

Plot of the network CMSF50055 after choosing the adequate T_{trans}



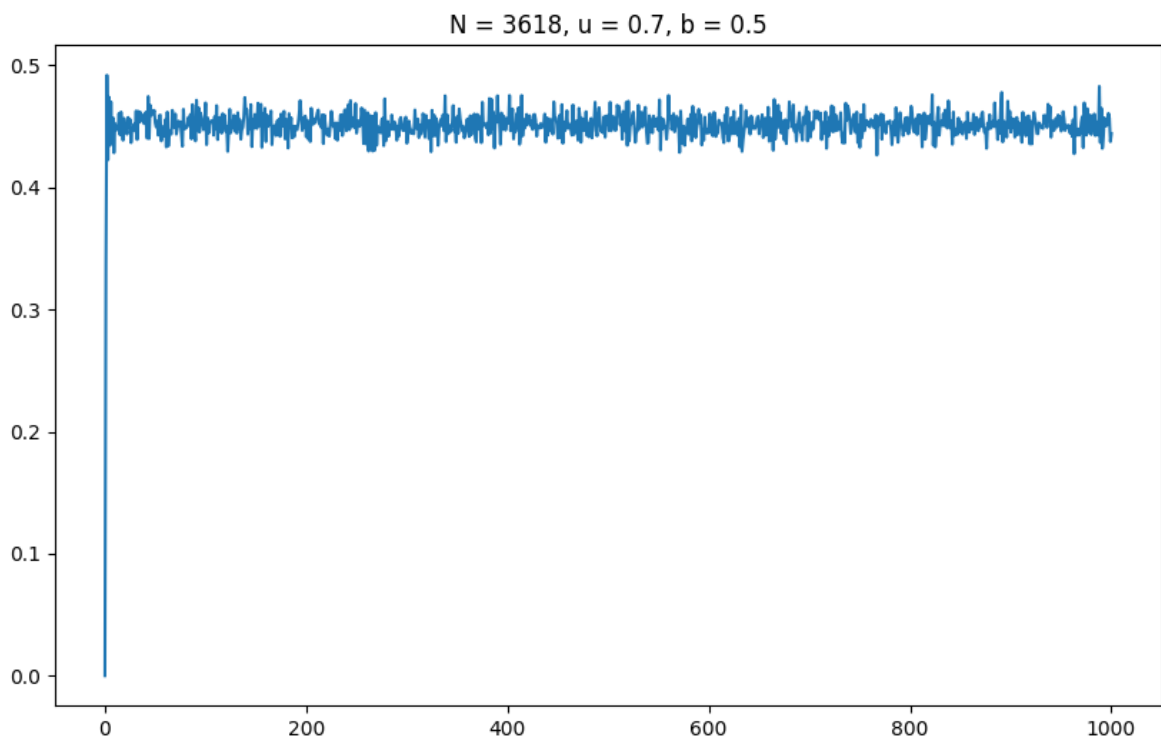
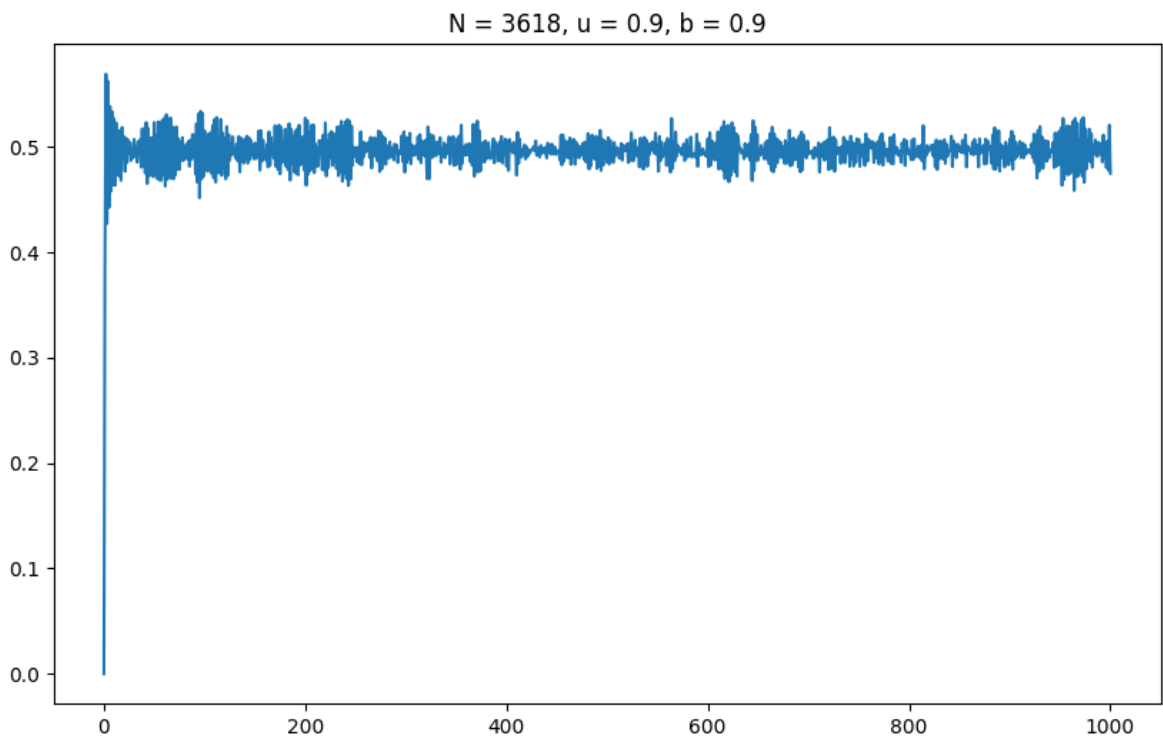
Plot of comparison MC MMCA for the network CMSF50055

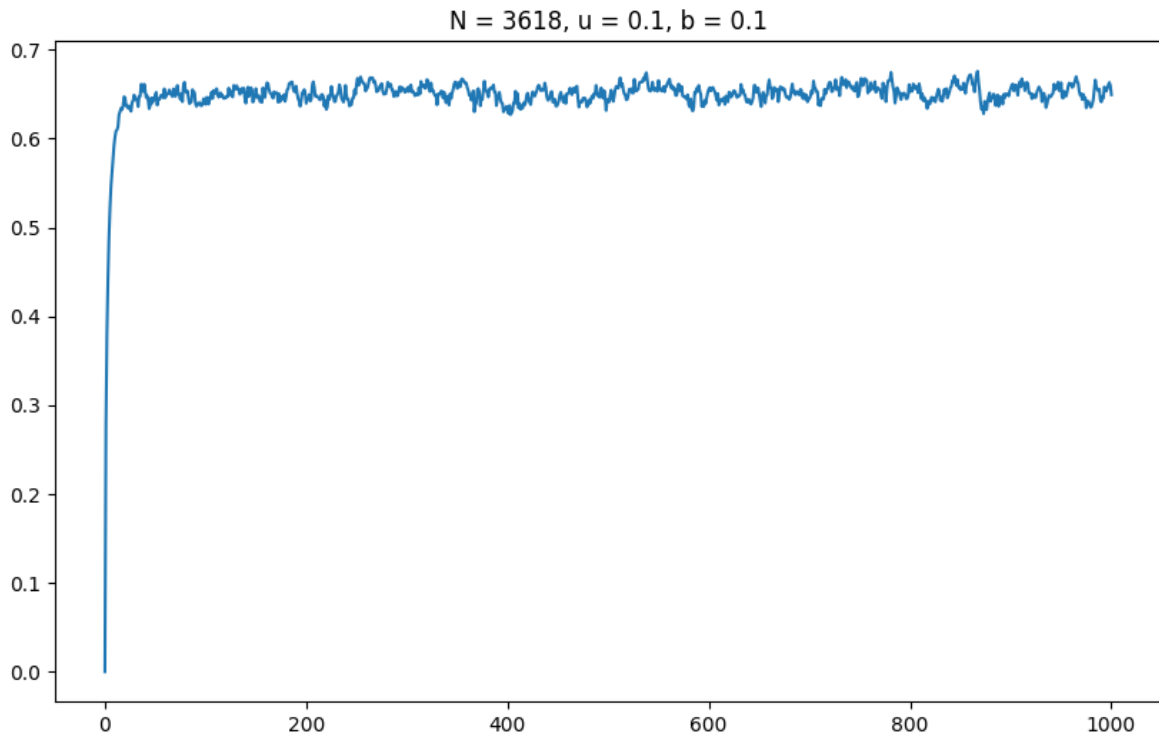
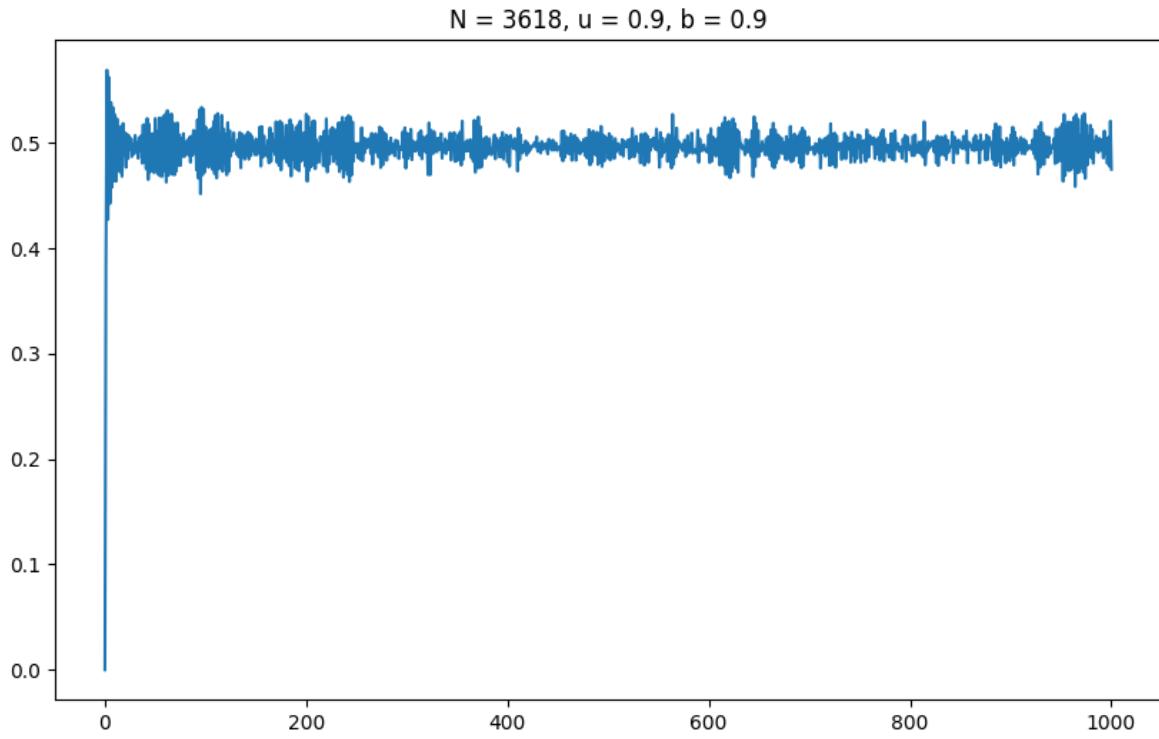


Parameters used here : $N = 3618$, $T_{\max} = 1000$, $T_{\text{trans}} = 900$ and $u_{\text{list}} = 0.1, 0.3, 0.5, 0.7, 0.9$ and $b = 0.1, 0.3, 0.5, 0.7, 0.9$.

The combination plots can be found in the notebook, we have only shared four plots to show the results.

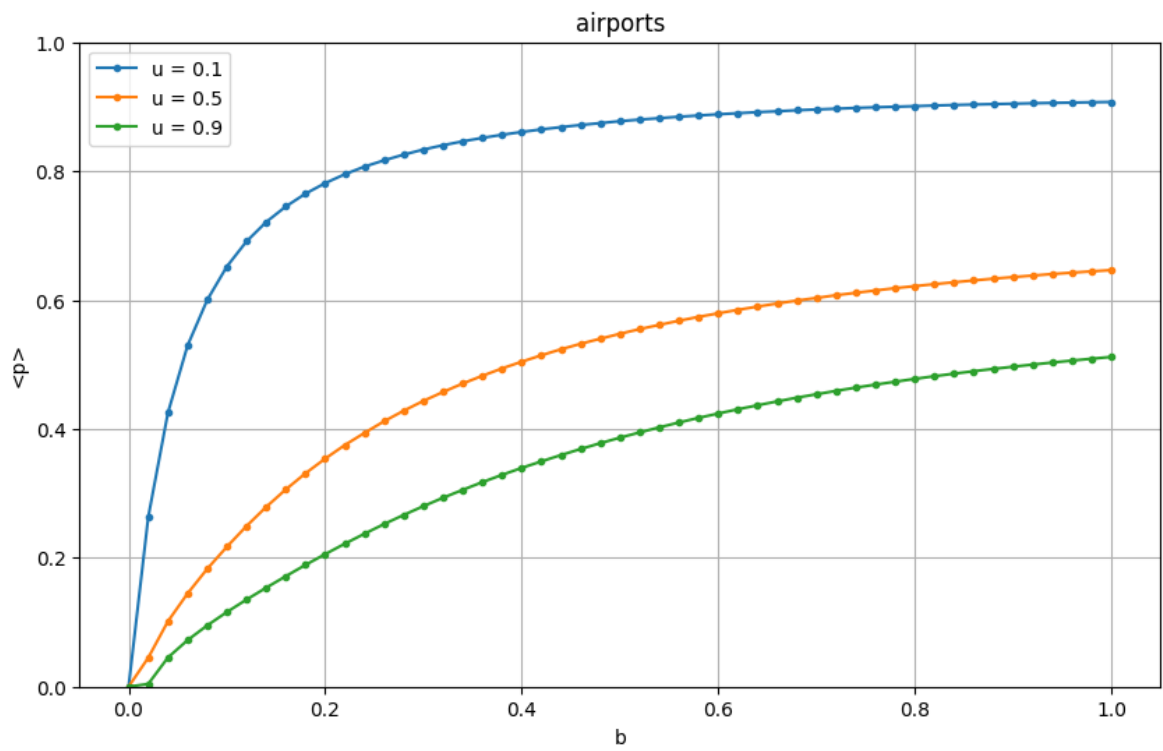
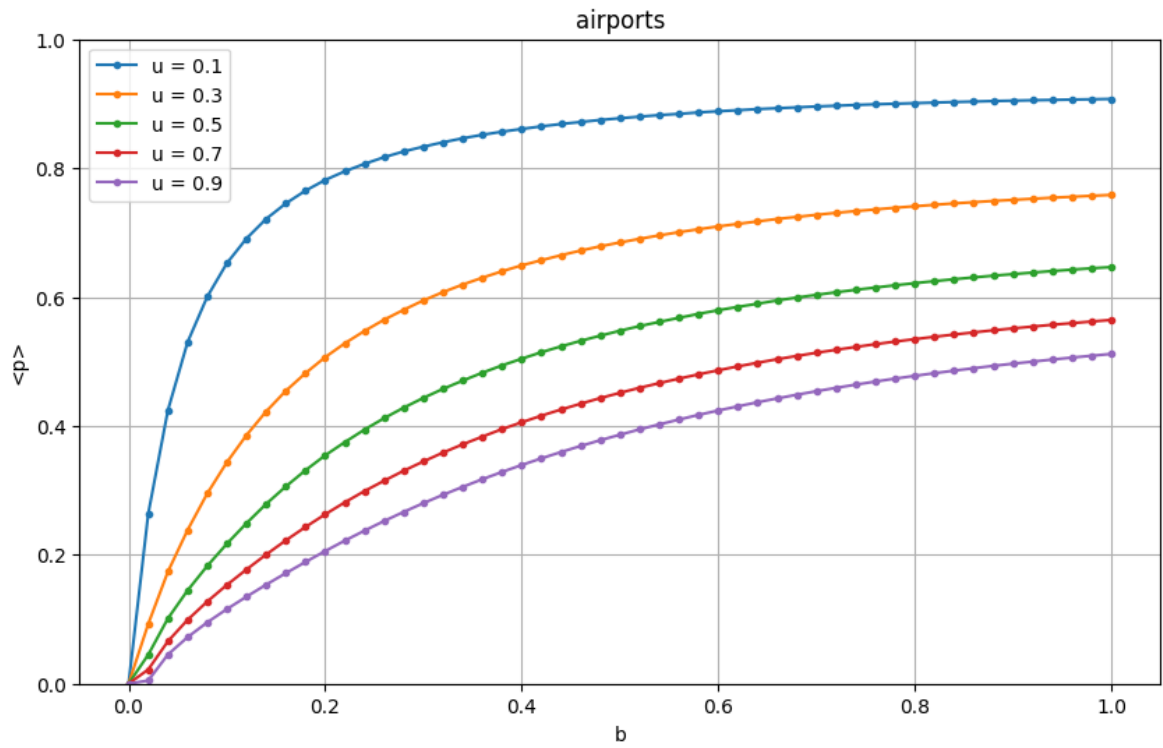
Combination plots of the network Airport



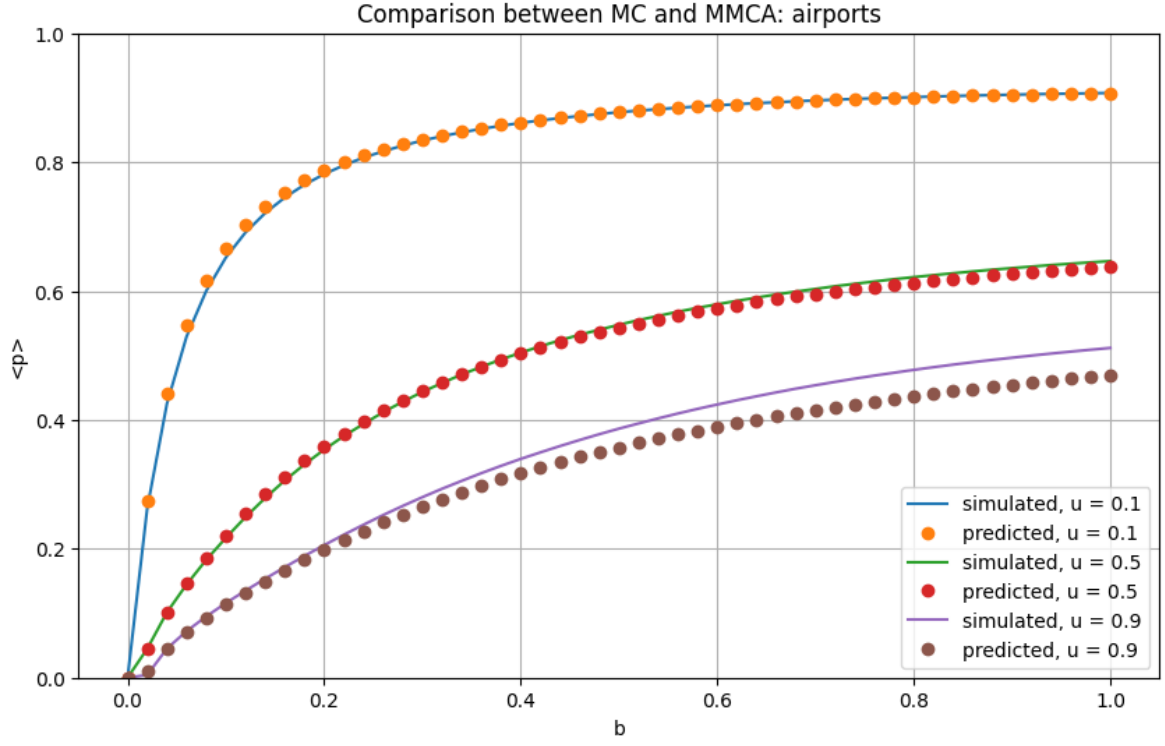


To test the computational cost, the ulist stayed 0.1, 0.3, 0.5, 0.7 and 0.9. And we choose a $T_{\text{trans}} = 100$ since the stationary state starts before reaching 100. So we estimated it.

Plot of the network Airport after choosing the adequate Ttrans



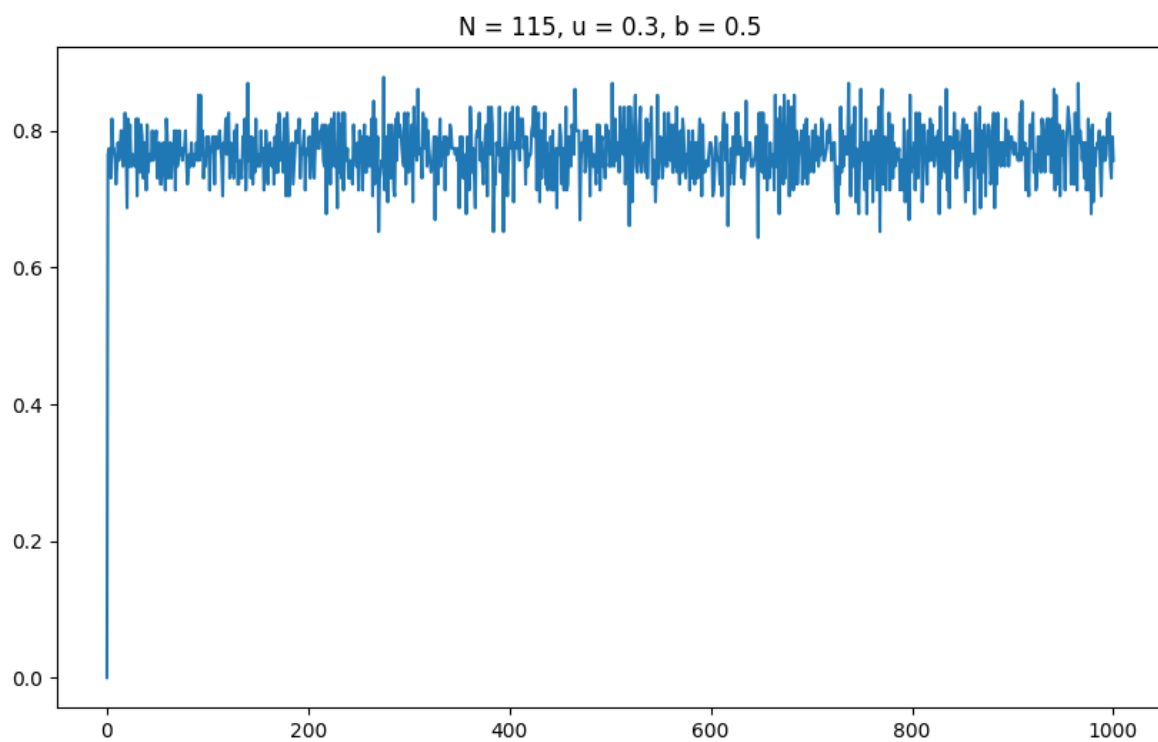
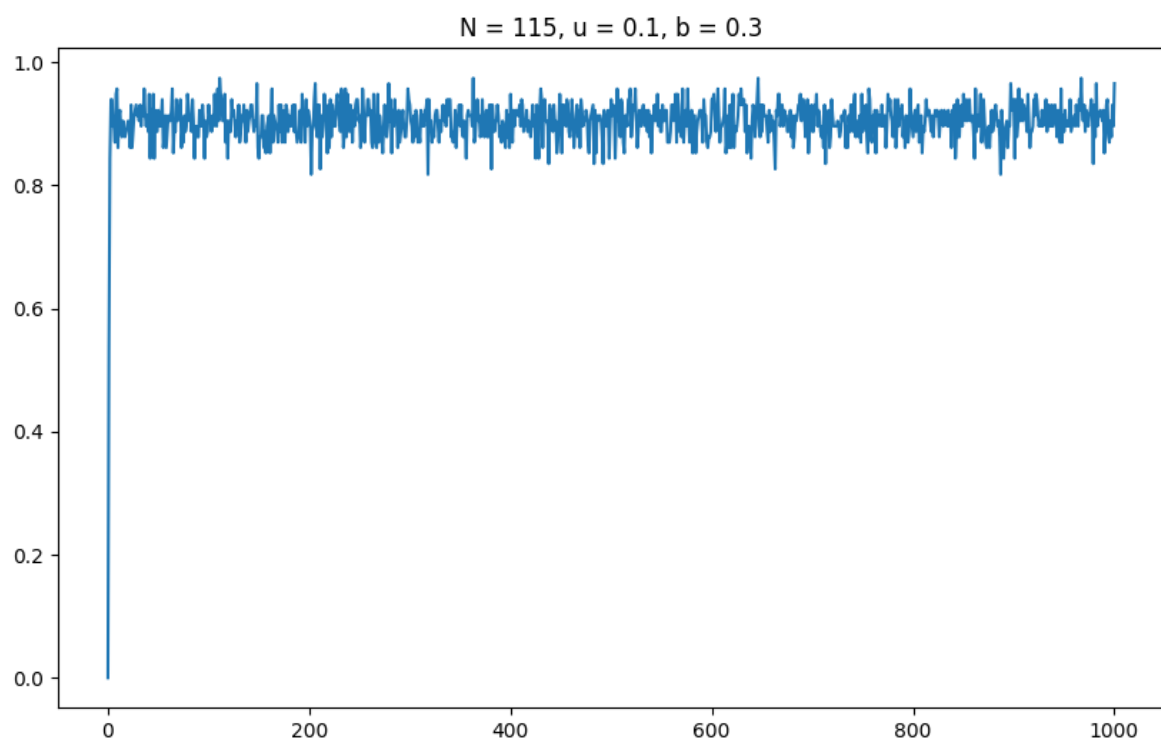
Plot of comparison MC MMCA for the network Airport

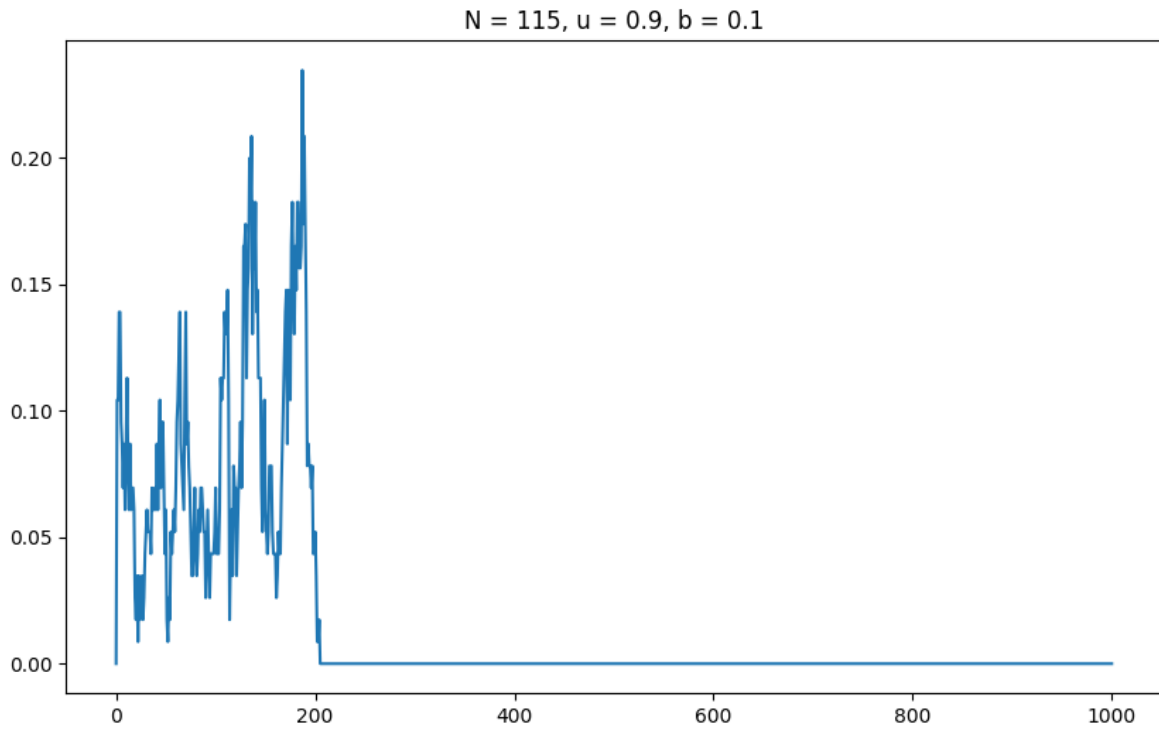
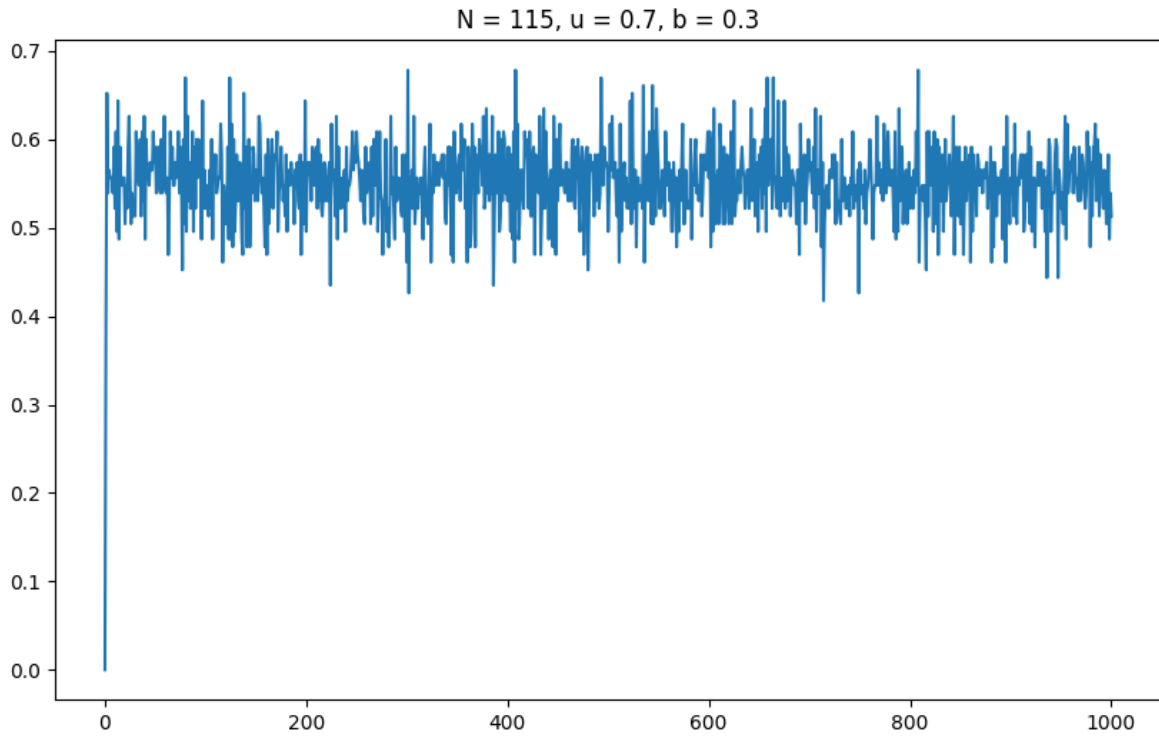


Parameters used here : $N = 115$, $T_{\max} = 1000$, $T_{\text{trans}} = 900$ and $u_{\text{list}} = 0.1, 0.3, 0.5, 0.7, 0.9$ and $b = 0.1, 0.3, 0.5, 0.7, 0.9$.

The combination plots can be found in the notebook, we have only shared four plots to show the results.

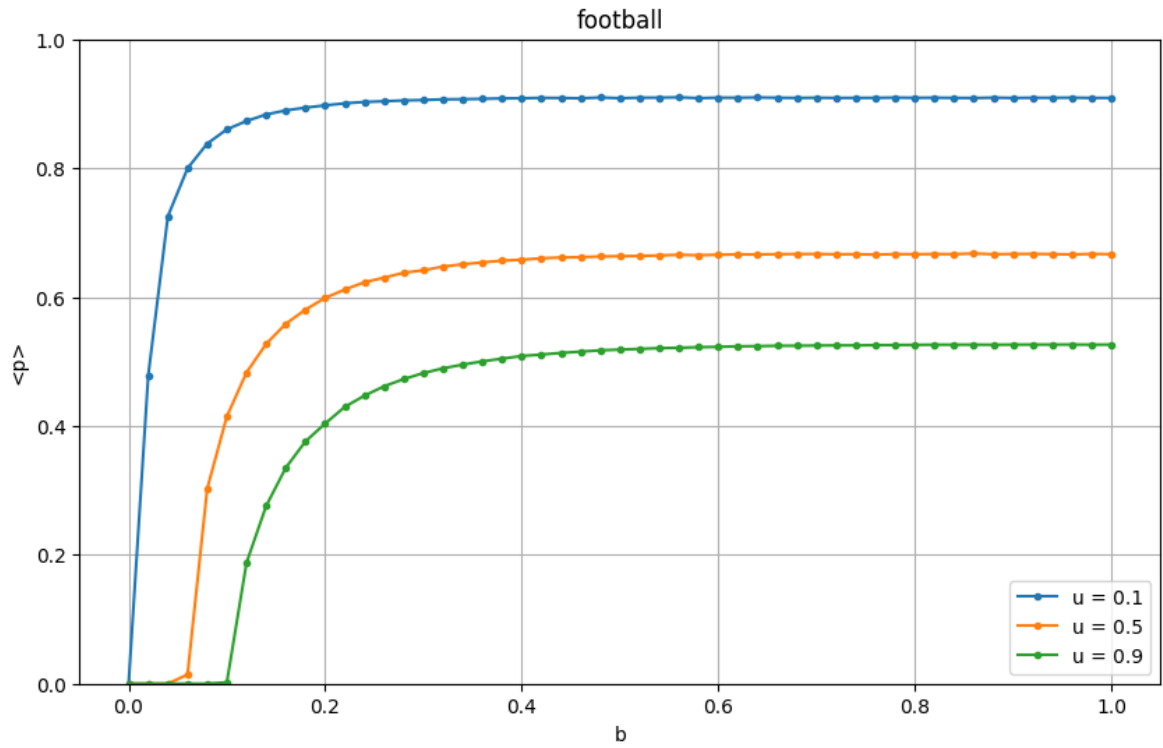
Combination plots of the network Football



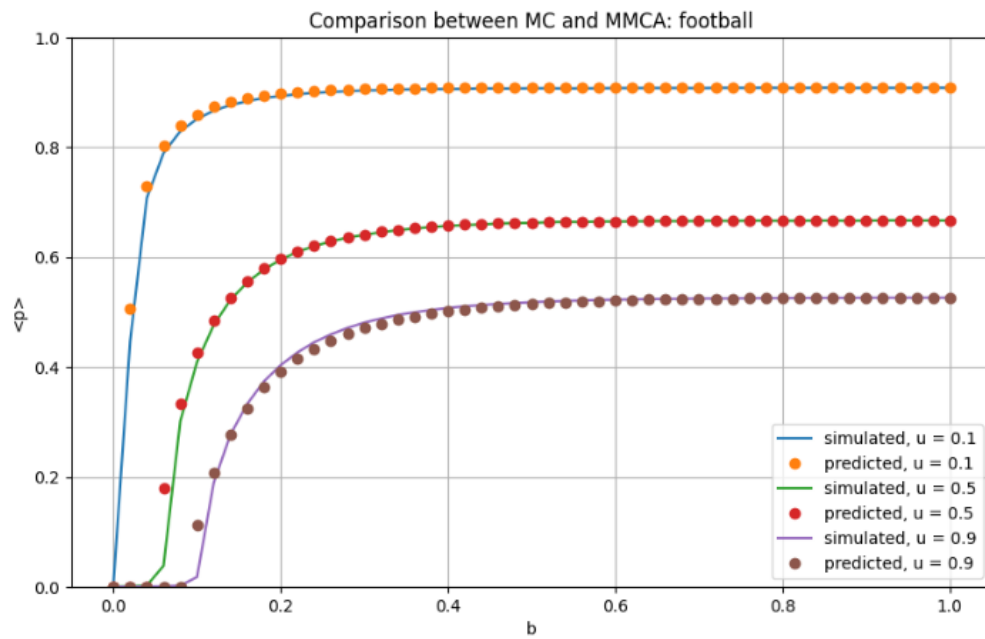


To test the computational cost, the ulist stayed 0.1, 0.5, and 0.9. And we choose a $T_{\text{trans}} = 200$ since the stationary state starts later. So we estimated it.

Plot of the network Football after choosing the adequate Ttrans

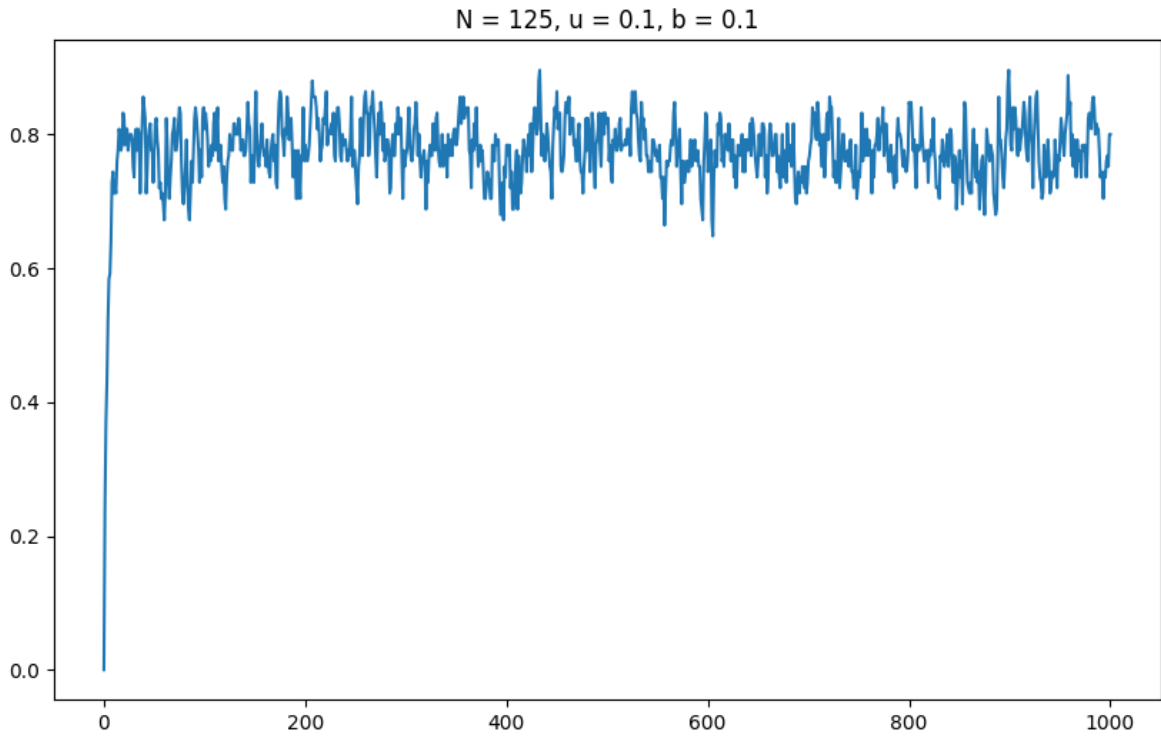


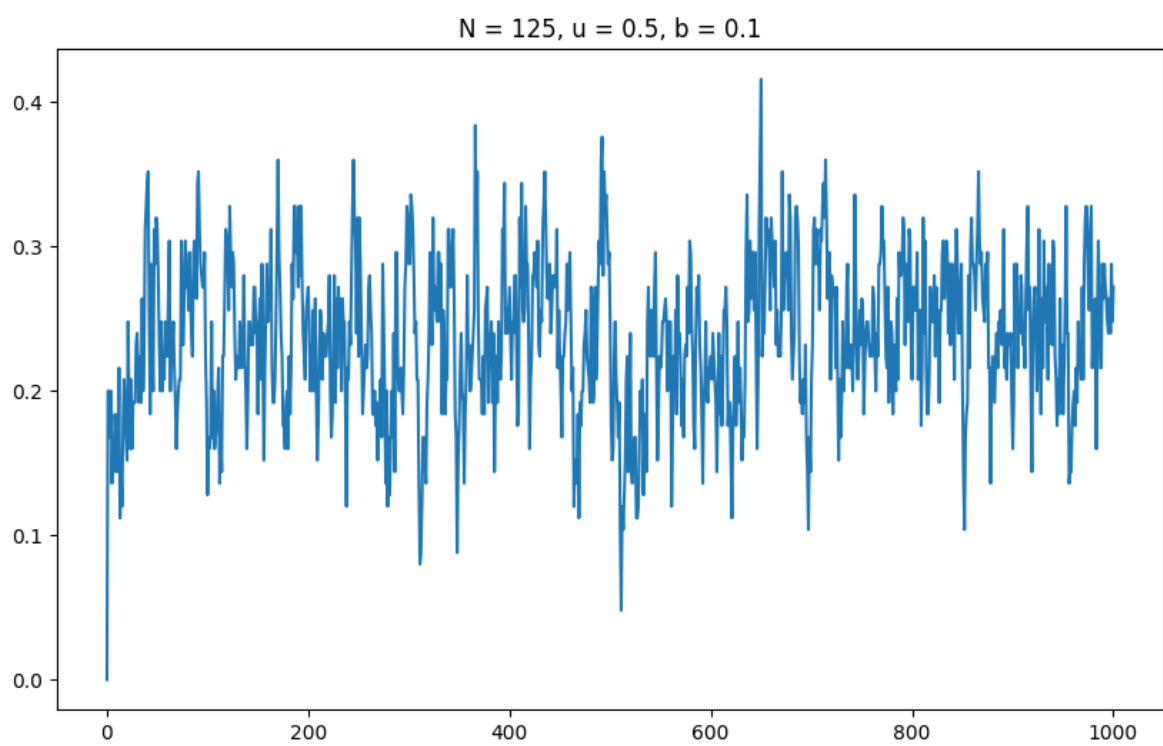
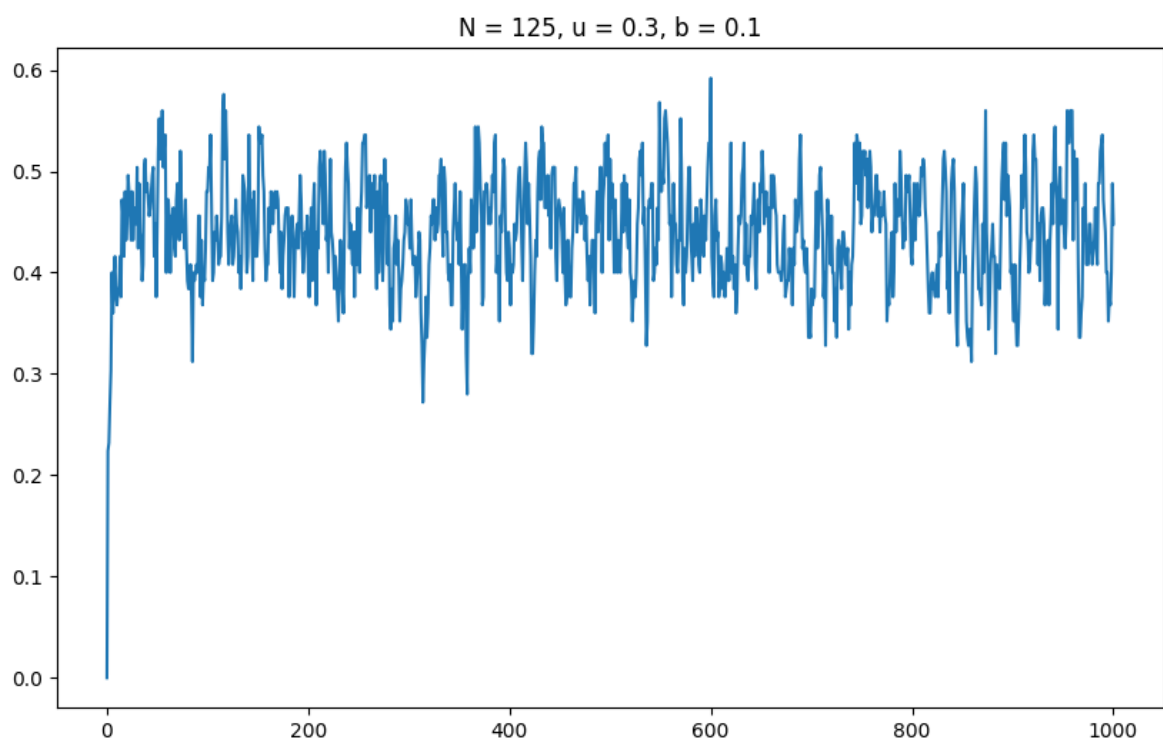
Plot of comparison MC MMCA for the network Football

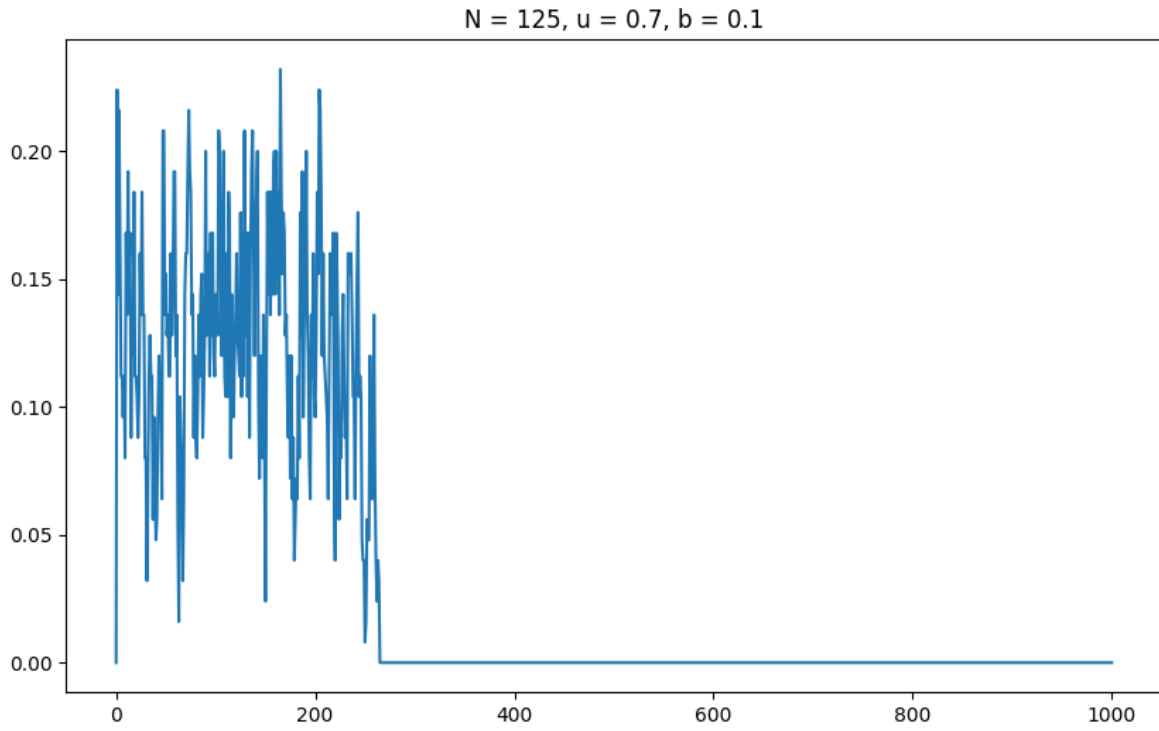


Parameters used here : $N = 125$, $T_{\max} = 1000$, $T_{\text{trans}} = 900$ and $u_{\text{list}} = 0.1, 0.3, 0.5, 0.7, 0.9$ and $b = 0.1, 0.3, 0.5, 0.7, 0.9$.
The combination plots can be found in the notebook, we have only shared four plots to show the results.

Combination plots of the network RB125

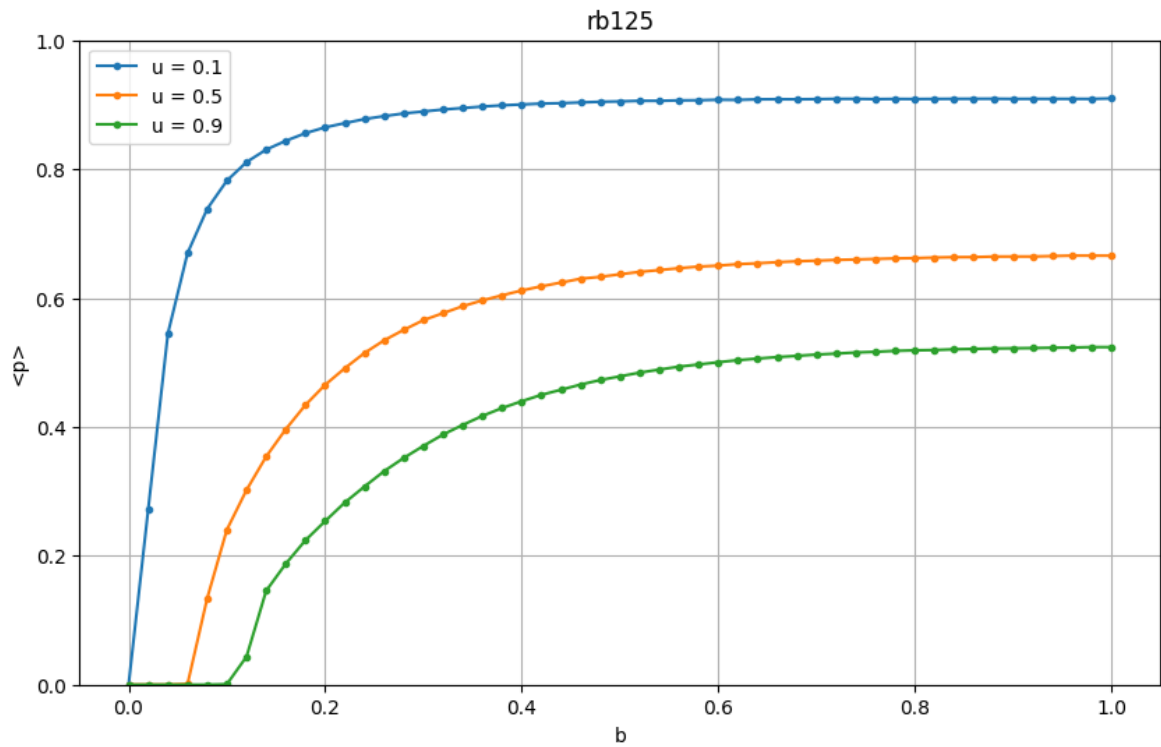




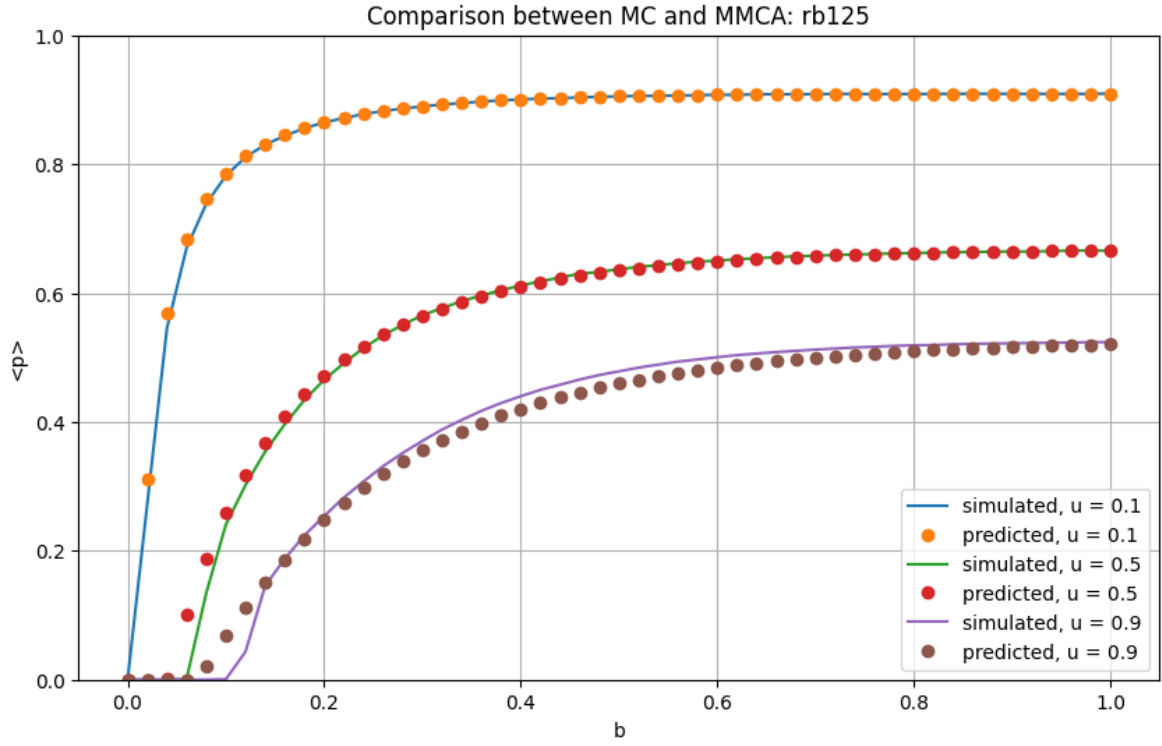


To test the computational cost, the ulist stayed 0.1, 0.5 and 0.9. And we choose a $T_{\text{trans}} = 200$ since the stationary state starts later. So we estimated it.

Plot of the network RB125 after choosing the adequate T_{trans}



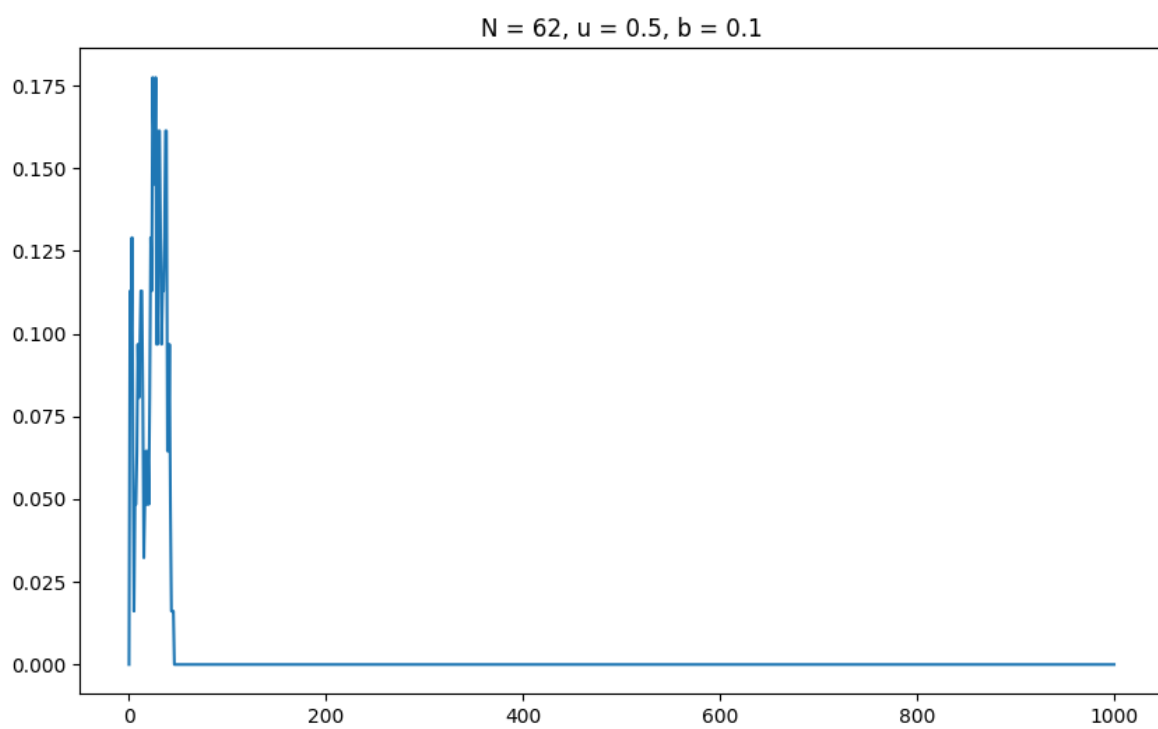
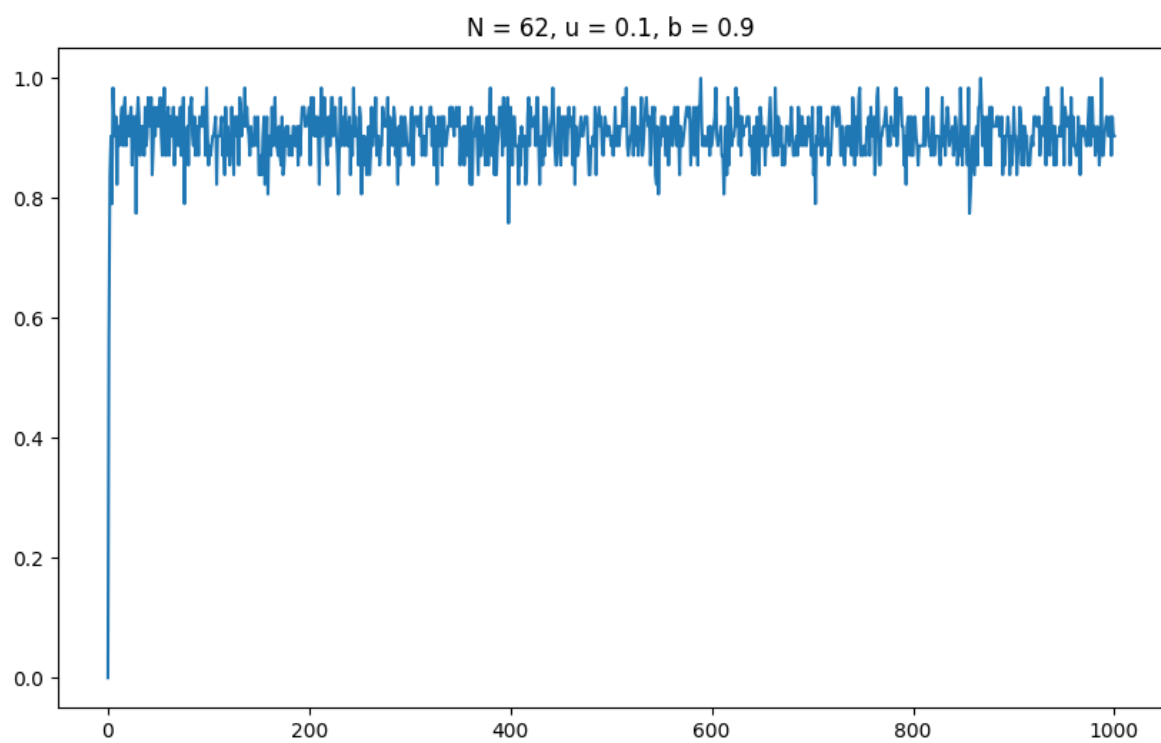
Plot of comparison MC MMCA for the network RB125

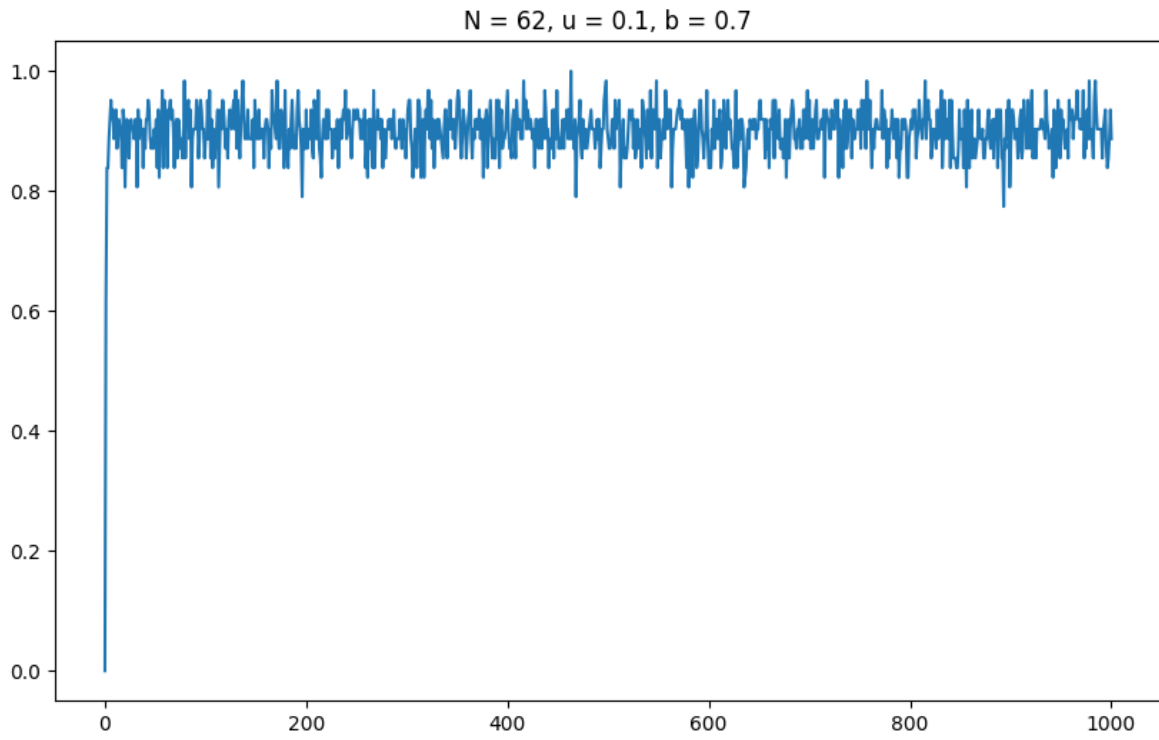
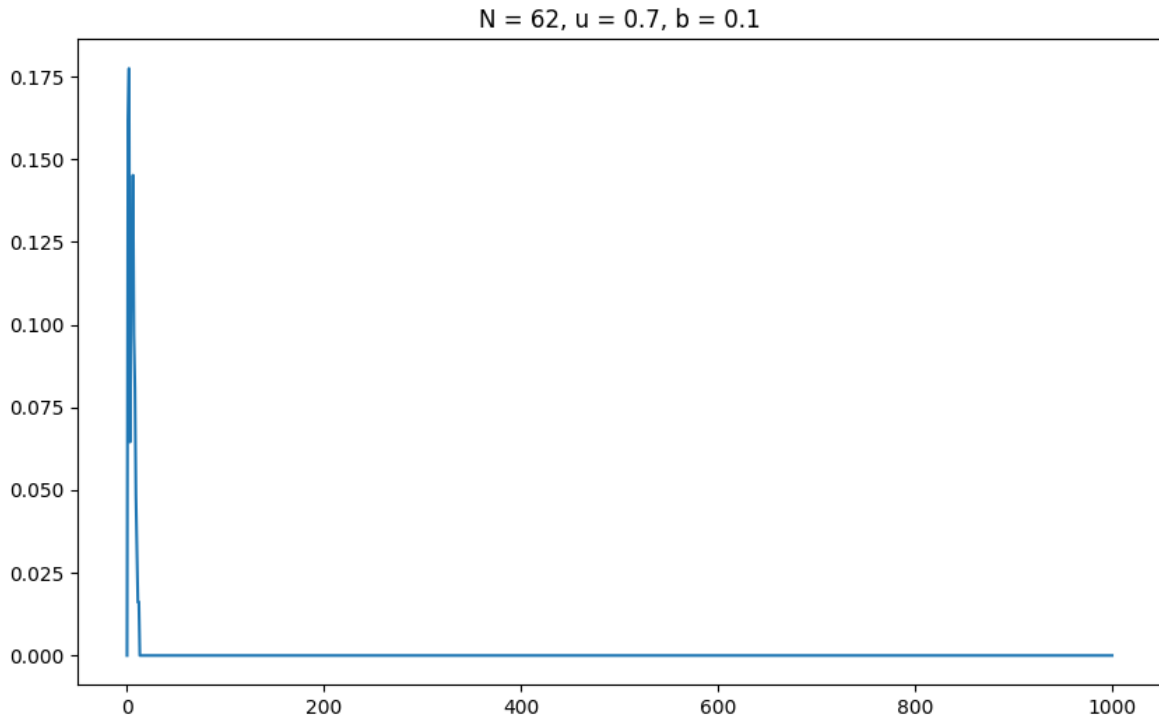


Parameters used here : $N = 62$, $T_{\max} = 1000$, $T_{\text{trans}} = 900$ and $u_{\text{list}} = 0.1, 0.3, 0.5, 0.7, 0.9$ and $b = 0.1, 0.3, 0.5, 0.7, 0.9$.

The combination plots can be found in the notebook, we have only shared four plots to show the results.

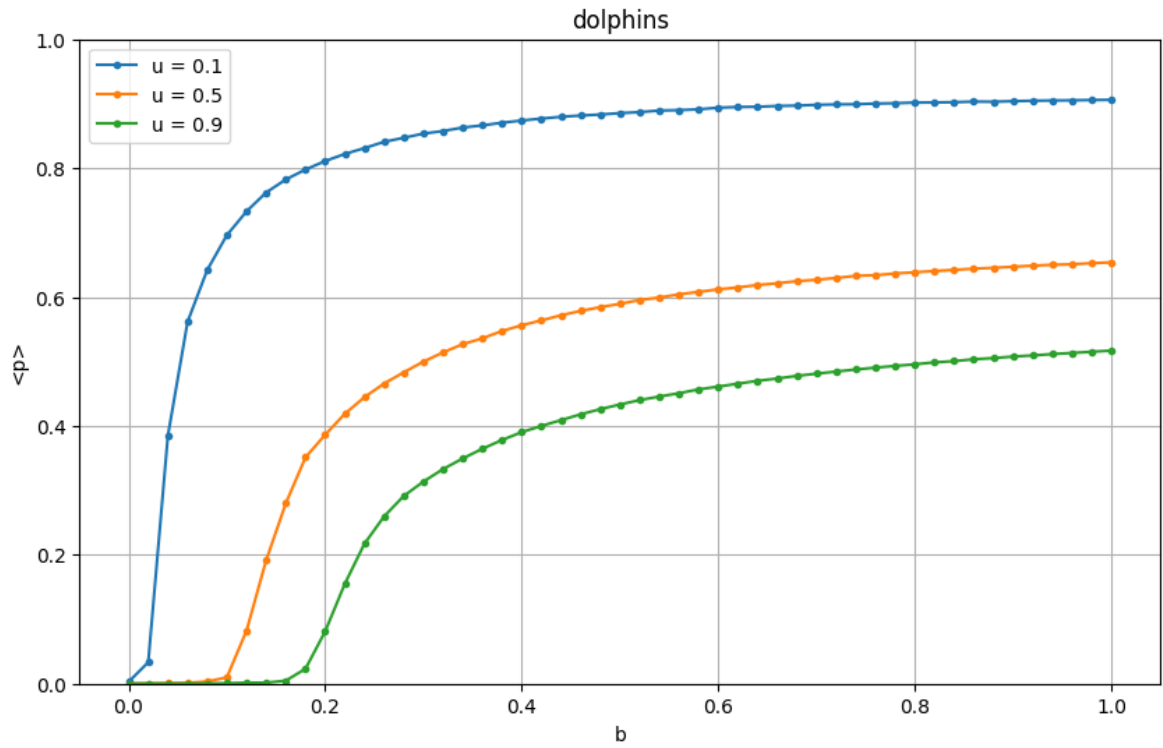
Combination plots of the network Dolphins



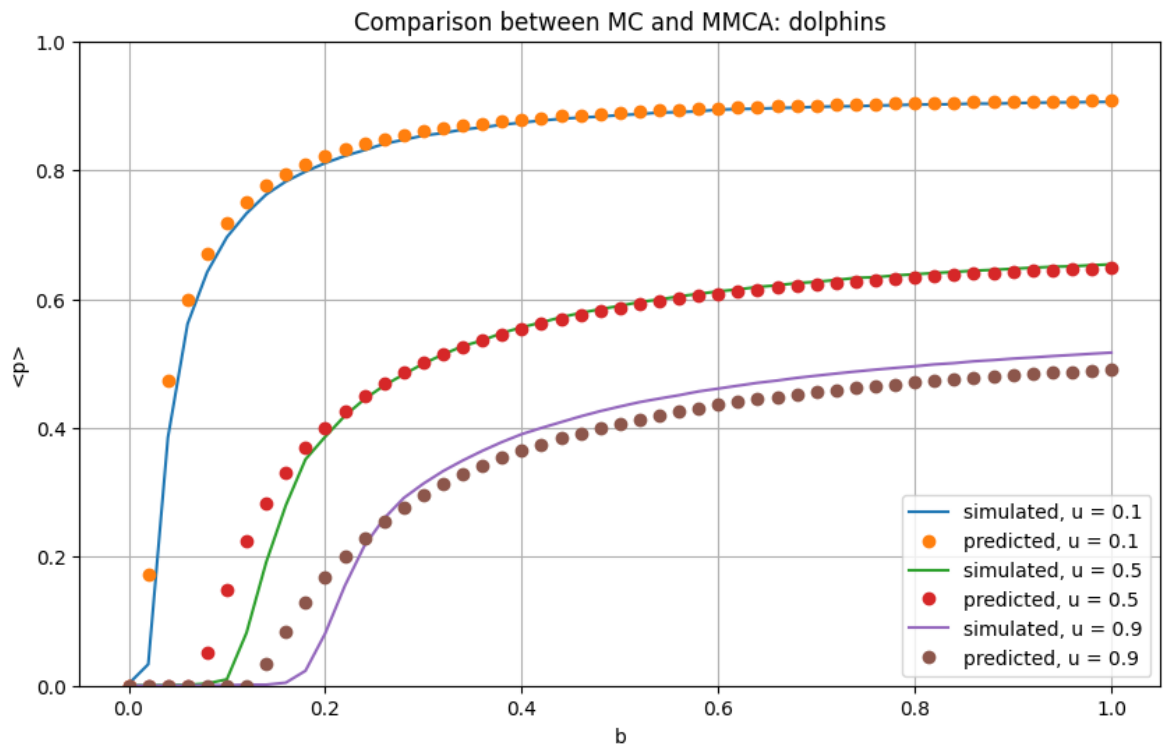


To test the computational cost, the ulist stayed 0.1, 0.5 and 0.9. And we choose a $T_{\text{trans}} = 200$ since the stationary state starts after it immediately. So we estimated it.

Plot of the network Dolphins after choosing the adequate Ttrans



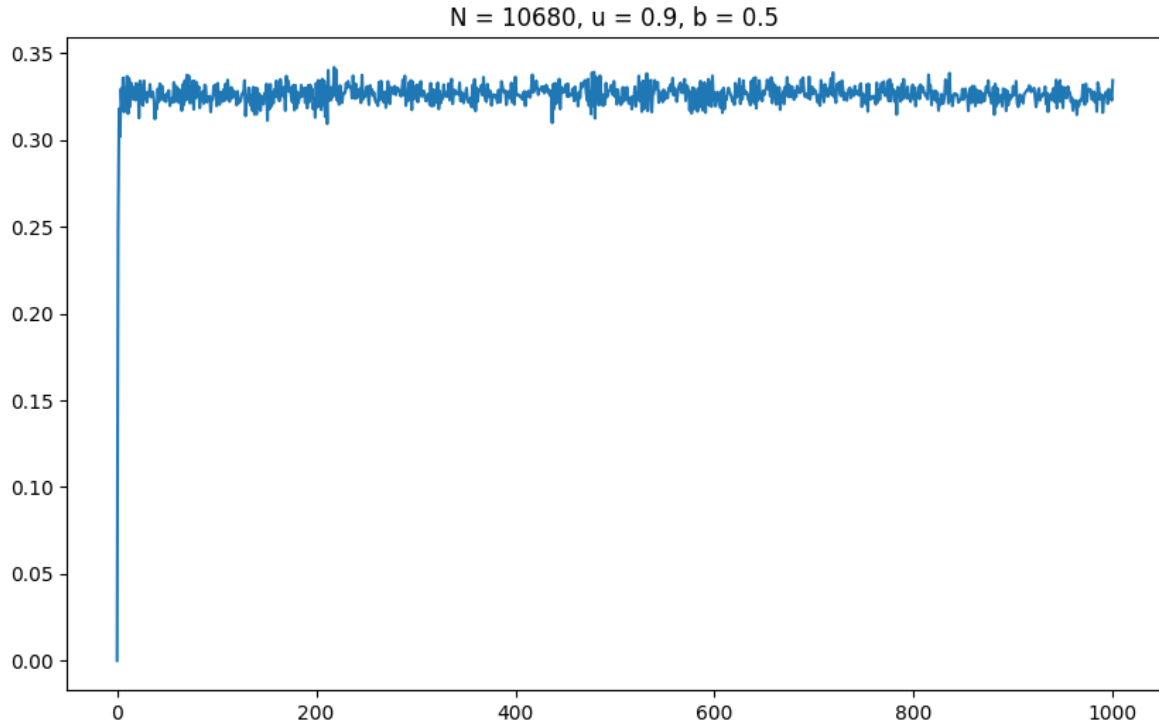
Plot of comparison MC MMCA for the network Dolphins

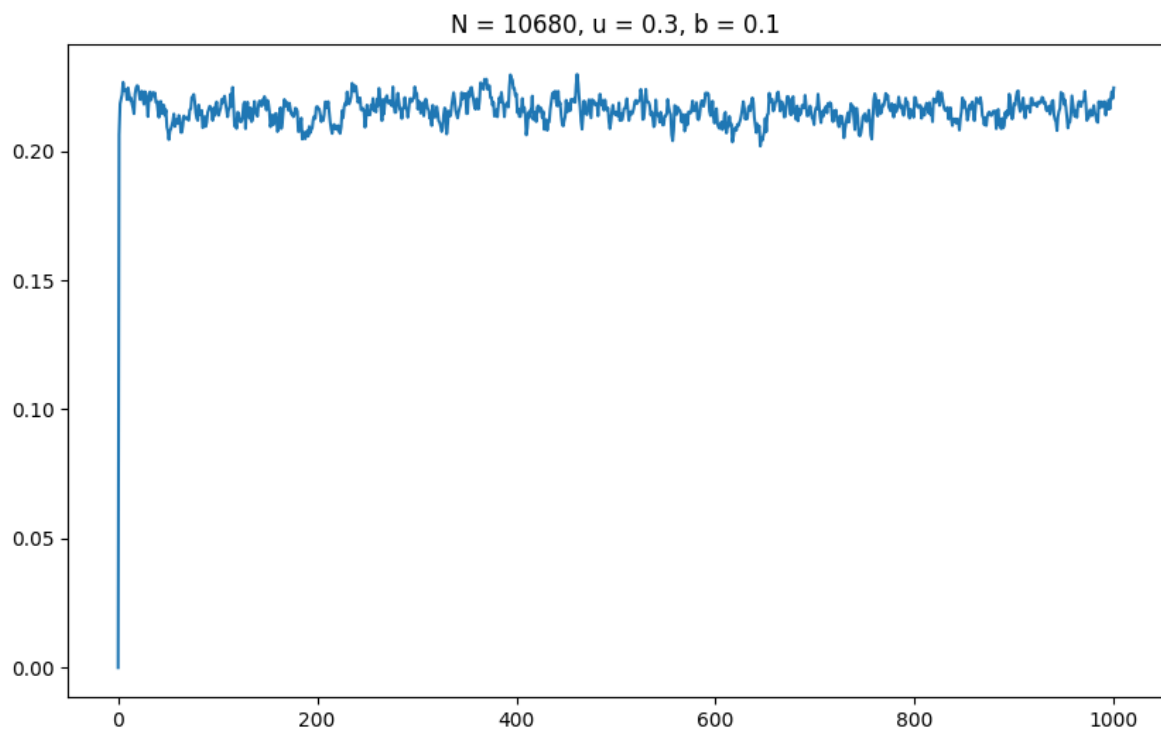
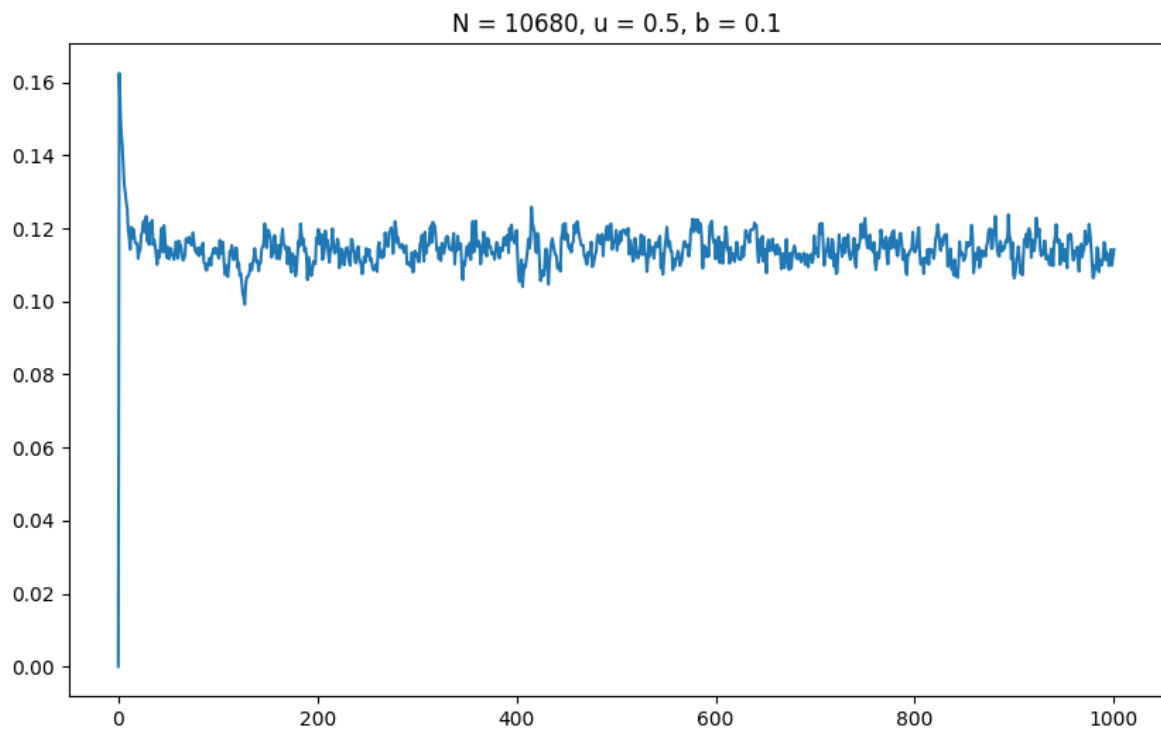


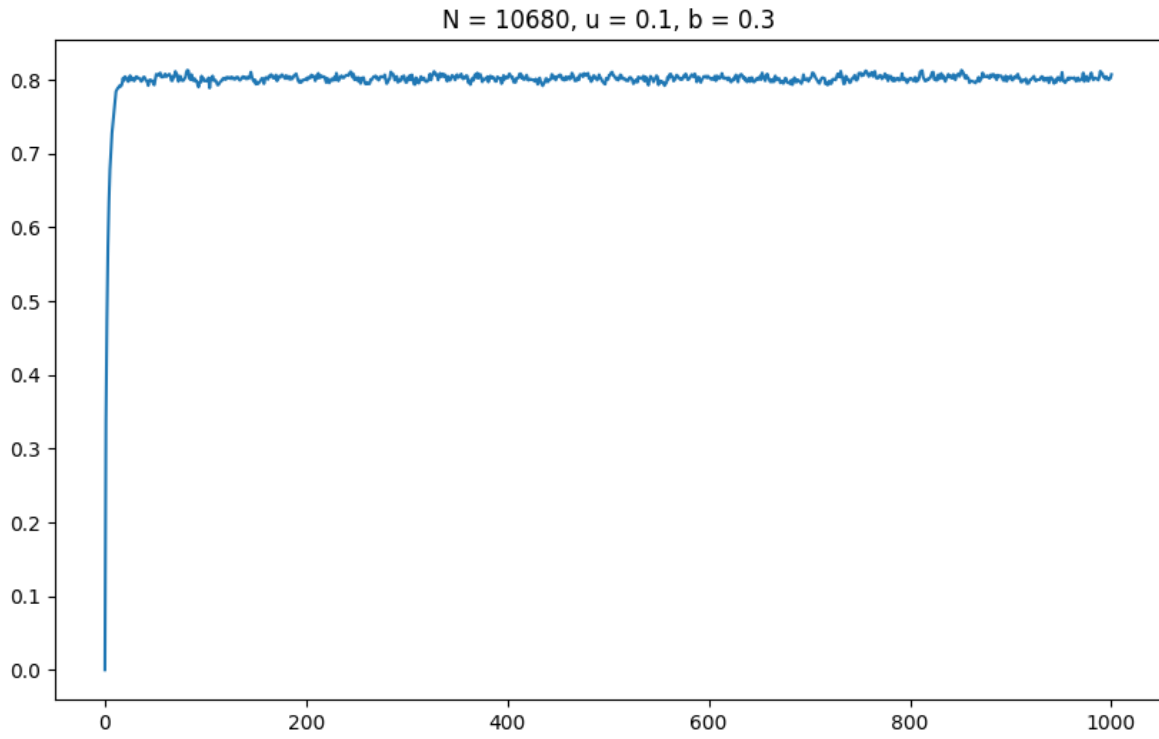
Parameters used here : $N = 10680$, $T_{\max} = 1000$, $T_{\text{trans}} = 900$ and $u_{\text{list}} = 0.1, 0.3, 0.5, 0.7, 0.9$ and $b = 0.1, 0.3, 0.5, 0.7, 0.9$.

The combination plots can be found in the notebook, we have only shared four plots to show the results.

Combination plots of the network PGP

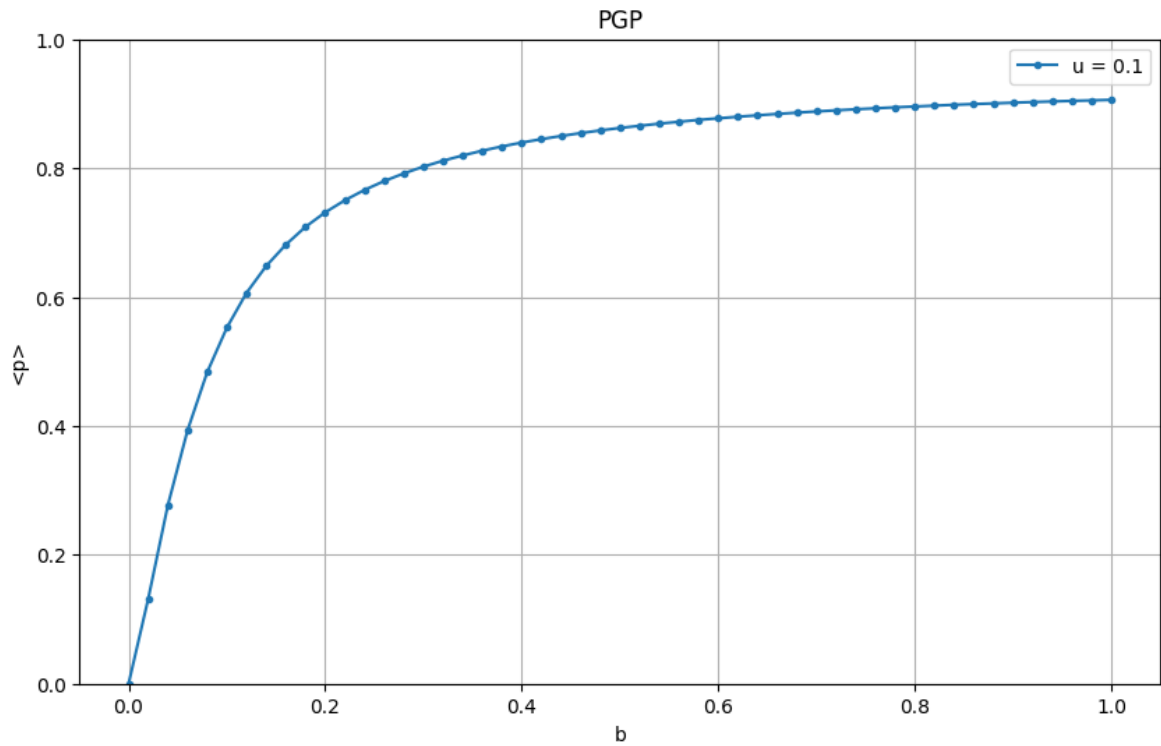




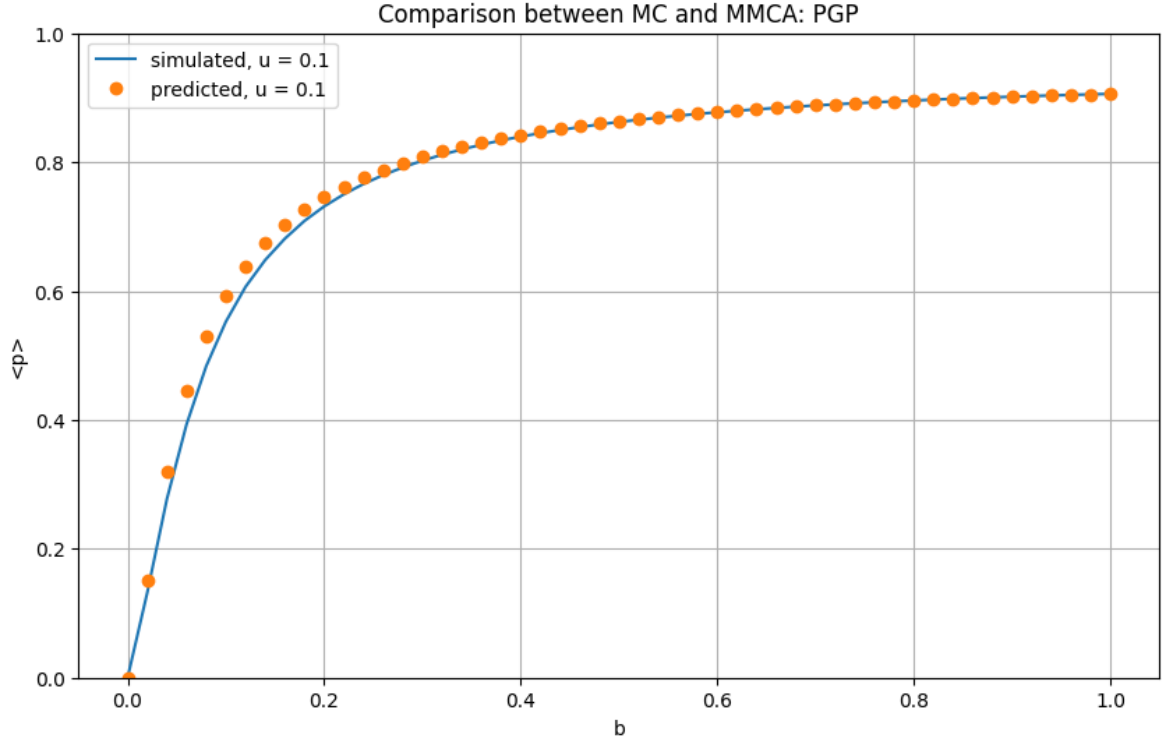


To test the computational cost, the ulist stayed 0.1, 0.5 and 0.9. And we choose a $T_{trans} = 100$ since the stationary state starts after it immediately. So we estimated it.

Plot of the network PGP after choosing the adequate T_{trans}



Plot of comparison MC MMCA for the network PGP



2.3 Conclusions about the work

First, we can observe that the SIS model has been successfully used to perform a Monte Carlo simulation of epidemic spreading dynamics in complex networks on the ER, SF, and real-world networks.

The simulation was then run to determine the percentage of infected nodes in the stationary state as a function of the disease's infection probability for various recovery probabilities. The results we got demonstrate that the network structure affects how a disease spreads. For example, compared to the ER network, scale-free networks exhibit a more robust epidemic spread to the removal of highly connected nodes.

We also can compare the time of execution that will be indeed a struggle if we have a network with more than 10 000 nodes. We have tested for small networks as well to see if the performance is better. In our case, we got some of the MC Simulation results after a few hours of running which is a challenge if we ever want to analyze a whole network belonging to a city.

We then continued our analysis by adding a comparison part. Overall, the Microscopic Markov Chain Approach (MMCA) model provides a good approximation of the percentage of infected nodes across a wide range of values according to our comparison of the results of Monte Carlo simulation with the MMCA model. Our results also demonstrate the importance of taking into account the network architecture when researching the dynamics of an epidemic spreading and the applicability of the MMCA model for estimating the proportion of infected nodes in large networks.