Agent-Based Simulation on different approaches efficacy to contrast wrong rumors

Lorenzo Chicchi
https://github.com/Lorenzochicchi/Rumors-Spreading-ABS

April 5, 2019

1 Introduction

In these years, the growing interest in opinion spreading is greater than ever. A lot of previous works tried to understand how opinions and rumors were born and how they spread across the society [1]. A first challenge to model the opinion spreading phenomenon consists in being able to represent the environment (i.e., the society). We use a network to model the social environment. Indeed, social interactions, both the short-range and long-range ones, can be represented by a network [2, 3].The network nodes represent people and the edges represent the interaction among them. However, a good environment model is not enough. Another key aspect we must take into consideration is the individual complexity in spreading opinions. The tool of ABS (Agent-Based Simulation) can help us in this sense [4]. In this work we use both the network theory and the ABS to simulate the spreading of a wrong rumor into a society. Our model is based on the model [1] and we use the same rules to model both the interactions among agents and the rumors distortions.

Our goal is to find the best way to contrast the wrong rumors. We insert some incorruptible agents (repairers) in the network who cannot modify their own opinion and they try to spread the “right” rumor to the other agents. We evaluate three different strategies to insert the incorruptible agents: random attachment, preferential attachment by degree centrality and preferential attachment by betweenness centrality. It is important to say that both the right rumor and the wrong one are exchangeable, that is, we evaluate the best way to contrast wrong information inside a social network and the best way to infect it with wrong information at the same time.

2 The model

2.1 Overview

The opinions

We use binary strings to represent the rumors. The strings length L is one of the parameters of the model and we use $L = 5$ in agreement with [1]. The wrong rumor is represented by the string '00000', the right rumor is represented by the string '11111'. The other strings represent rumors included betweenn the wrong one and right one. Since the model is so general, the strings can alternatively be observed as a piece of news, an opinion, or a (potentially distorted) fact. From now onward the strings will be called 'opinions', as it is clearer. The attributes "right" and "wrong" can have different meanings, too, so they will be replaced by general attributes *true* and *false*.

Agents

There are two different types of agents: the *spreaders* and the *repairers*. The first ones simply represent neutral people who interact with one another and spread opinions. They have a memory with a maximum capacity in which they store the opinions they accept; the opinion that appears the most in a spreader memory is his "major opinion", that is the opinion he will try to spread. Morever,they are likely to distort their major opinion when they try to spread it. The distortion is more likely to happen if the agent has lots of different types of opinions in his memory. The second type of agents, the repairers, try to correct the opinions inside the network. They have no memory and they only try to spread the true opinion to the other agents inside the network.

Organization of the simulation

First of all, we create a network using the Albert-Barabasi model [2] to obtain a scale-free network, that is a good model of real social networks. At the beginning of the simulation we associate spreaders agents at all the network nodes. Then we randomly choose a node which begins to spread the false opinion and we put it in the front of his memory.

The spreaders distort and spread their opinion to the agents in the nearest nodes all along the first half of the simulation. Halfway through the simulation we insert a number R of repairers in the network. We evaluate three different rules to insert the repairers. The first one consists in choosing randomly a number R of spreaders and replacing them with repairers. The second rule consists in choosing the first most connected R nodes (i.e., the nodes with the highest degree) and replacing the occupying spreaders with repairers. The third rule is similar to the second one, but instead of choosing the most connected nodes, we choose the highest betweenness centrality nodes.

Distortions

Each spreader can distort his major opinion before transferring it to other agents. The probability of distortion depends to the variety in the spreader memory. If a spreader has a single type of opinion in his memory, he has less probabilities to distort his opinion. We use the entropy as a variety measure in the spreaders memory:

$$H = \sum f_i log(f_i)$$

where f_i is the frequency of opinion type i. Thus, the probability of distortion is:

$$P_d = \frac{1}{\exp(\frac{H_{max}-H}{H_{max}}K) + 1}$$

where H_{max} is the maximum value of the entropy and K is a parameter that quantifies the resistance to distortion. The more the value of K is high, the less is the probability of distorting the opinion.

The distortion consists of changing one of the elements of the string. For example, the string '00100' can be distort in '00110'. Each string has a subset of strings in which it can be distort. In this sense, we can define a kind of distance between two strings. The maximum distance is between the string associated with the true opinion and the one associated with the false opinion.

Spreading phase

In the spreading phase agents try to pass his major opinion to their neighbors: if the neighbor is a repairer he rejects the new opinion, if the neighbor is a

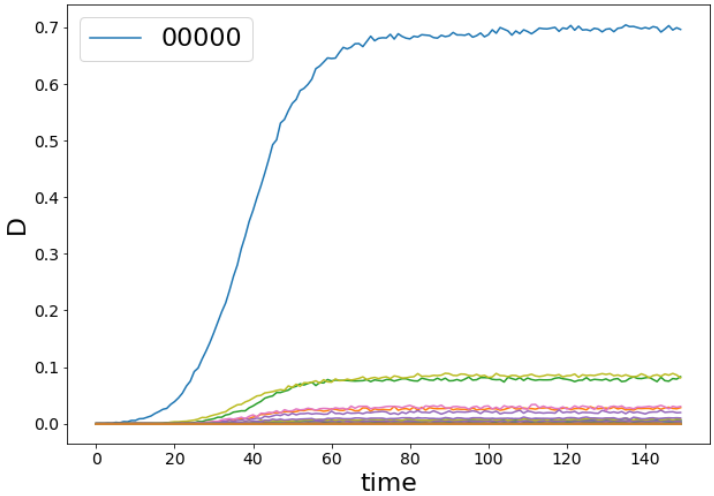

Figure 1: Evolution of opinion types concentration with no repairers entered. $D = N_i/N$, with N_i number of agents with major opinion of type i.

spreader he accepts the new opinion with a probability:

$$P_{ij} = \frac{k_i^{\beta}}{\max\limits_{l \in \text{Neigh(j)}} (k_l^{\beta})}$$

Where k is the degree and the parameter β quantifies the confidence in the most connected nodes. If $\beta > 0$ the agents prefer to accept opinions from nodes with a hight degree, if $\beta < 0$ the agents prefer to accept opinion from nodes with a low degree. A model with a high value of β can represent a society where people accept opinions only from the media; on the other hand, a low value of β can represent a society that has lost its confidence in the media.

If a spreader accepts an opinion, he puts it in his memory. If the number of opinions in the spreader memory is bigger than the capacity M the oldest opinion is removed.

2.2 The Code

Parameters

The parameters of the model are:

- β: confidence in most connected nodes.

- K: propensity to change spontaneously the major opinion.
- N: network size.
- L: lenght of binary sequences (i.e., opinion). L defines also the total number of opinion types, that is 2^L.
- M: lenght of spreader memory.
- R: number of repairers inserted in the network.

We have chosen to fix $N = 3000$, $L = 5$ and $M = 320$ in agreement with [1]. Moreover, if $\beta = -3$ e $K = 3$, until we insert repairers into the network, the opinions are clustered around the initial opinion '00000', which remains the most popular opinion (fig.1). This situation is good for our goals. It can represent well a social system where a wrong opinion has already been spread. The values of β and K can be explain by imagining a society that has lost confidence in the most connected agents and where people rarely change their ideas spontaneously. Therefore, we have chosen to fix these values for β and K. The number of repairers R is a free parameter in the model. We evaluate different values of R for each investigeted strategy.

The agents

Spreaders. The spreaders inizialization is show in the code below:

```
class Spreader:
        def __init__(self,spreader, List_rumor,M,L):
                self.H_max=-math.log(1/M,2)
                self.spreader= spreader
                self.memory =[[]]*M
                self.major_rumor=[]
                self.entropy=0.0
                self.count=[0]*pow(2,L)
```

- The **"spreader"** attribute is a boolean variable that is always *True* for spreader agents. It easly lets us know if an agent is a spreader or not.
- The **"memory"** attribute is a list of lists with fixed first dimension. First dimension is equal to the memory capacity M.
- The **"major opinion"** is the most frequent opinion in the memory.

- The **“entropy”** attribute is a measure for agent uncertainty, as we have defined it before.

- The **“count”** attribute is an hidden attribute that allows us to count the different opinion types inside the agent memory. It is a list of length 2^L as the number of opinion types.

Repairers. The repairers inizialization is:

```
class repairer:
        def __init__(self,spreader,TrueOpinion):
                self.spreader= spreader
                self.major_rumor=TrueOpinion
```

- The **“Spreader”** attribute is *False* to the repairers.

- The **“major rumor”** for a repairer is fixed equal to the true opinion. We have to pass on it when we create a new instance of class repairer.

Repairers are simpler than spreaders. They have no memory and do not need to have any of the attributes that allow spraders to calculate their main opinion and their level of uncertainty.

Time organization

After entering the parameters, the program runs like this:

```
List_of_rumors=MakeListRumors(L)
G=CreateGraphBarabasi(n)
InitGraph(G,List_of_rumors,Mem, L)
Tstar=item//2
for t in range (0,item):
  if t==Tstar:
        if Pref==0:
            InsertRepairersRandom(G,Rep,List_of_rumors[-1])
        if Pref==1:
            InsertRepairersPref(G,Rep,List_of_rumors[-1])
        if Pref==2:
            InsertRepairersBetw(G,Rep,List_of_rumors[-1])
  for i in G.nodes():
        if G.agent[i].spreader == True:
                G.agent[i].CalMaxRumor(L,List_of_rumors)
```

```
                    G.agent[i].CalEntropy(L, List_of_rumors)
                    G.agent[i].ValutaDistorsioni(K,L)
Interaction(G, beta)
```

First of all, we initialize the network and create a list with all the possible types of opinion. The *MakeListRumors* function takes as input an integer and returns a list of binary strings with a lenght equal to the integer. The *CreateGraphBarabasi* function returns a Barabasi-Albert graph of size n. The *InitGraph* function takes a graph as an input and associates a spreader agent to each node of the graph. Then we define an entering time (*Tstar*), that is the time step in which the repairers are entered in the network.
After that, the time loop begins. It each step we ask all sprader agents to calculate their major opinion (*CalMaxRumor* function), to calculate the entropy of their memory (*CalEntropy* function) and to assess possible distortions of their major opinion. Eventually, the spreaders interact with each others by the *Interaction* function.
When time index is equal to $Tstar$, R repairers are entered in the network following one of the three rules presented before.

3 Results

We found some interesting results. In particular, we observe a very strong resistence of the false opinion against the attacks. In (fig. 2) we report the evolution of opinions concentrations with the three different entering rules and for a number of repairers $R = 300$, that is 10% of the network size (a very large fraction). The entrance of the repairers into the network at the time step $T^* = 200$ initially does not involve any change in the density of the opinions . After a response time (that depend on the memory capacity) the concentration of the true opinion begin to increase faster than a linear function (we didn't exactly fit it). However, the concentration of the true opinion does not grow indefinitely and it sharply stops its growth. From that moment on, the concentrations keep constant values and they only show small fluctations.

This behavior is shared by all the three entering rules. it is possible that the discontinuity is due to the fact that the true opinion diffuse in a community of the network that is not well connected with the rest of the network. Thus, when the true opinion has finished infecting the whole community in which the seed was placed, its concentration stops growing abruptly. However, to confirm this idea a more detailed analysis with spatial definition is needed.

In order to consider a possible diffusion by step we looked at later time

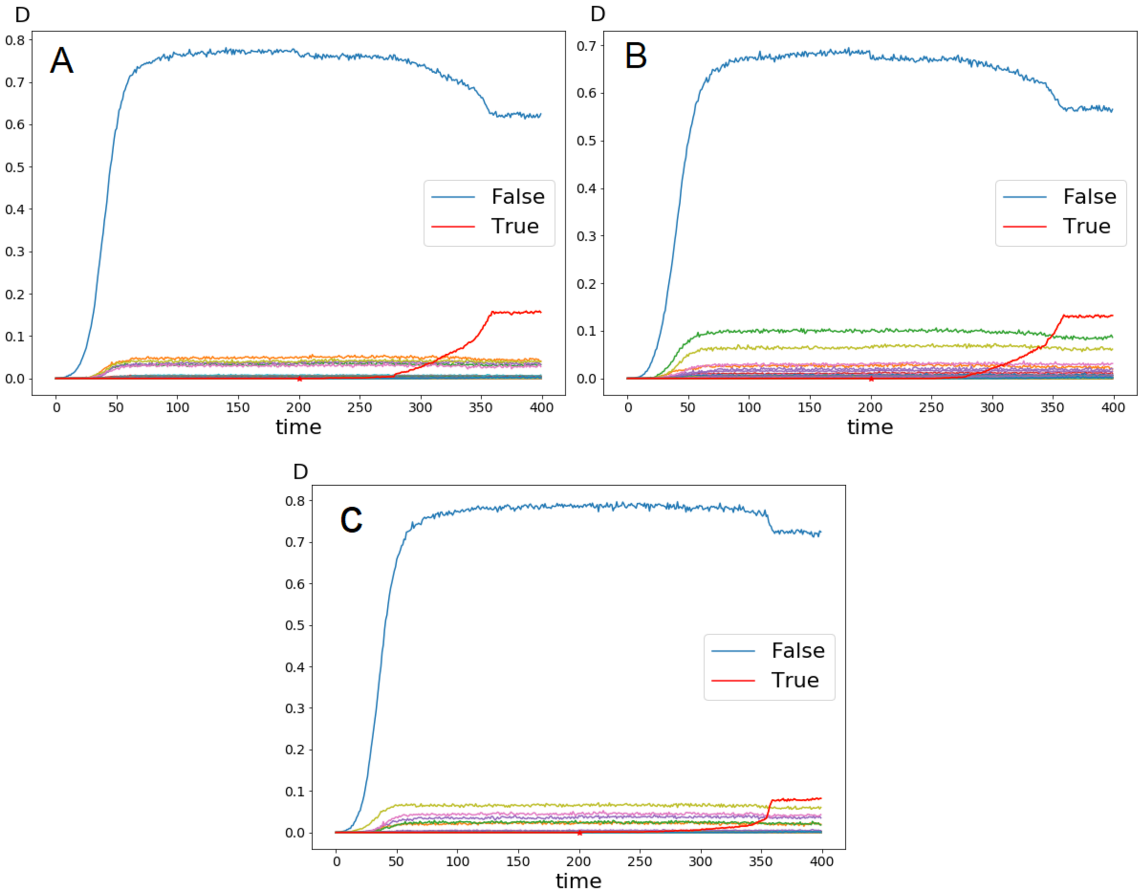

Figure 2: Evolution of opinions concentrations. At the time step 200 we have entered R=300 repairers inside the network. (N=3000). The figures A, B and C correspond respectively to entering rules with preferential attachment by betweenness centrality, preferential attachment by degree centrality and random attachment.

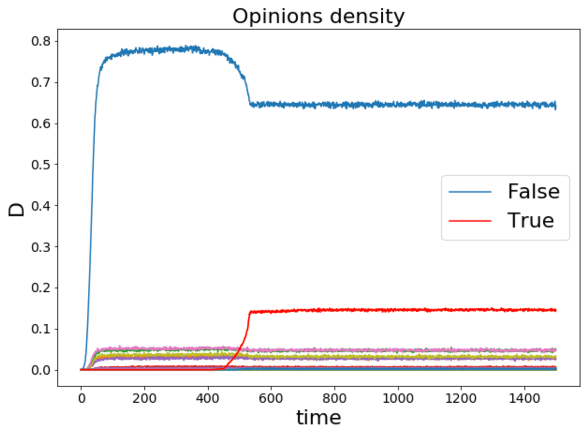

Figure 3: Evolution of opinions concentrations for a longer time range (Rule=Betweenness).

steps (fig.3) and we found that the state of the system appears stable. We cannot exlude that for longer time there is another jump (or more than one) for the value of the true opinion concentration (maybe due to the fact that the true opinion breaks in another community) but we didn't find it.

In fig.(4) the final concentrations of true and false opinions are displayed for

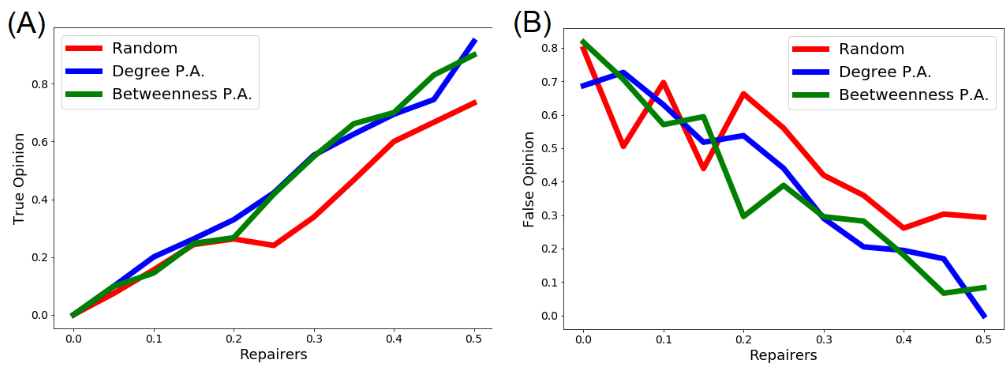

Figure 4: The evolution of the true opinion concentration (A) and false opinion concentration (B) for different values of the number of repairers entered the network.

different values of the number of repairers entered the network. It is important to observe that to have a suitable effect we need to insert in the network a number of repairers equal to a macroscopic fraction of network size. Clearly, this isn't a practicable strategy for a real network. Morever, one more time the three different strategies have similar effects: random strategy seems to involve a less efficent effect in order to diffuse the true opinion inside the network as compared to the other two strategies, but the difference is negligible. Unexpectedly, we don't found any transition point and the fraction of spreaders which have true opinion as major opinion increase linearly with the fraction of repairers entered the network.

The effect of strings length

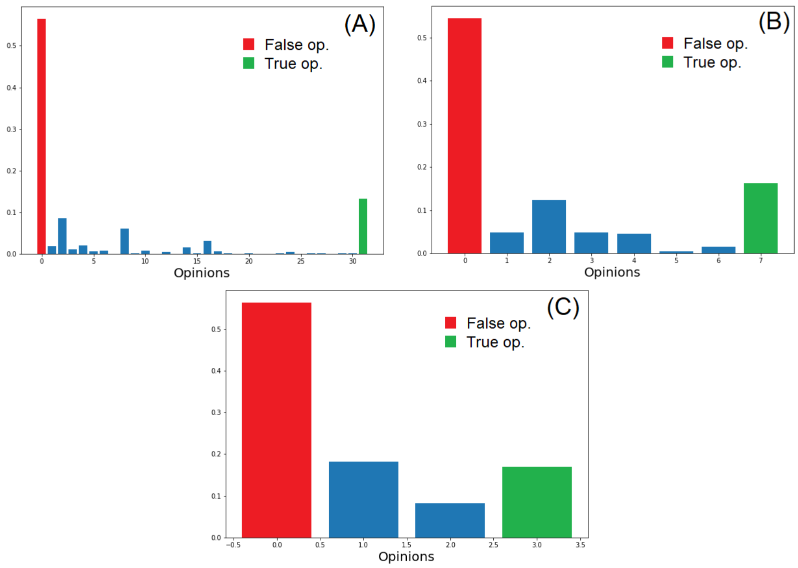

Figure 5: The distribution of the concentrations of opinion types at the end of the simulation for three differnt values of L: $L = 5$ (A), $L = 3$ (B) and $L = 2$ (C).

The strings length L is associated with the number of opinions types, which is 2^L. We can think that the effect of the repairers can depend on this parameter because it increases or reduces the types of opinions between the true opinion and the false opinion, and this can act as a kind of barrier that slows down the "conversion". However, three different simulations are made respectively with $L = 5$, $L = 3$ and $L = 2$ and they show how the lenght L does not involve any substantial difference. Fig. (5) shows the concentrations of the opinions at the end of the three simulations and the concentrations of the true opinion and false opinion are substantially the same. The reson can be find in the parameter K which in our simulations is fixed to the value $K = 3$ and which represent the resistance to the distortions. In fact, in the previous section we said that the phenomenon of distortion can induce a kind of metric among the opinion (i.e. the strings). If there is a strong resistance to the distortion the distance between all the pairs of opinions becomes larger and the opinions are basically independents.

The importance of the parameter β

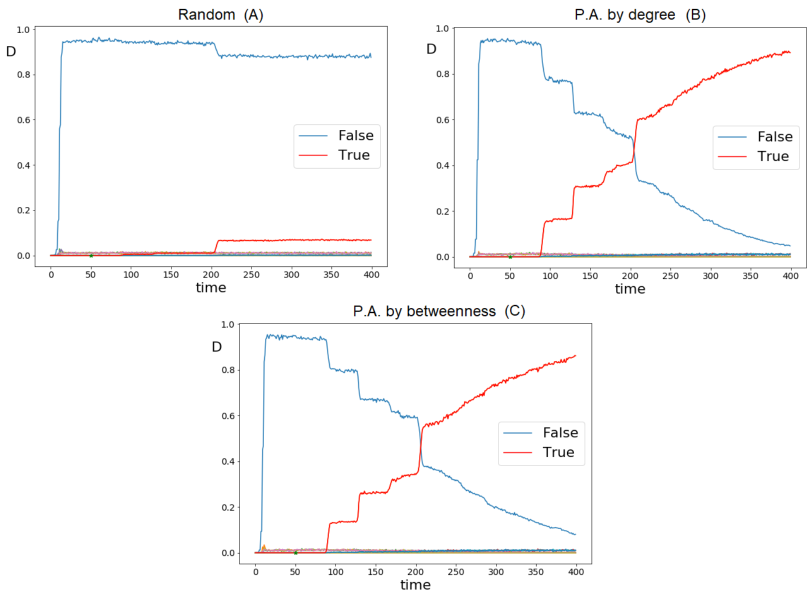

Figure 6: Evolution of the concentrations of the opinions for the three different entering rules with $\beta = 0$. The number of repairers entered in the network is $R = 300$, the 10% of the network size.

Until now, the three stategies to contrast the false opinion seem to not involve different effects. Such result is due to the value of the parameter β that we have chosen ($\beta = -3$). For such low value the agents have not confidence in the most connected nodes, which are yet the shortcuts to reach the different parts of the network. When the true opinion is inserted inside the network, it is not able to spread to the other network components and it remains confined in the part of the network where it was inserted.

In order to prove this idea, we have evaluated the effects of the three stategies in the case $\beta = 0$, keeping the value of parameter $K = 3$ (6). Such situation corresponds to a society that has no bias towards nodes, thus, the spreaders accept with the same probability the opinions that come from nodes with high degree and those that come from nodes with low degree. If we use the random attachment strategy (fig.6.A) there is a high probability to insert the repairers in nodes that are not well connected since there are few hubs in a scale free network. Thus, the hubs remain occupied by spreaders which are still infected by false opinion. In order to spread the true opinion across the network the repairers must first infect the closer hub. However, the hubs are not easily infected since they are connected to many other nodes that are continuing to spread the false opinion towards them. We may suppose that the repairers are able to infect only nodes with low degree which are inside the same network component where repairers were placed. At the contrary, the other two strategies are very efficient (fig.6.B/C). The true opinion rapidly diffuse among the network nodes and its concentration reach values close to one. We also may assert that there are not significant differences between the two stategies, this is probably due to the fact that in a scale-free network the hubs are at the same time the nodes with higest degree and the nodes with the highest betweenness centrality.

The fluctations: the parameter K

The parameter K is associated with the fluctations of the opinions concentrations. How it is showed in [1], for low values of K there is no opinion that prevails against the others. Thus, the distribution of opinions is very heterogeneous until the repairers are inserted (fig.7).

When the repairers enter the network, the concentration of true opinion

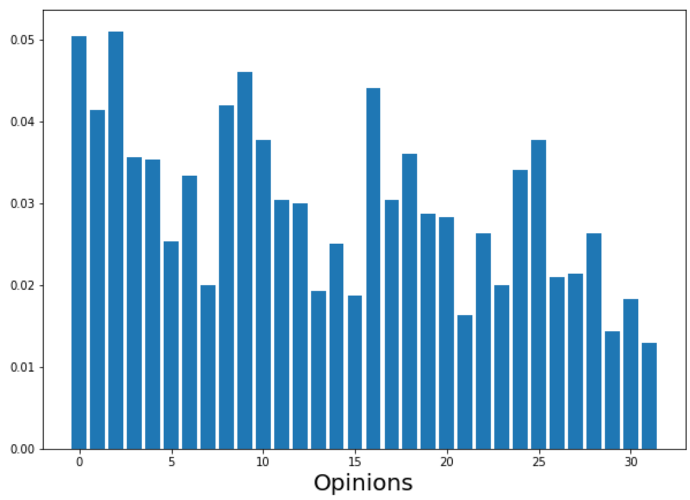

Figure 7: Dstribution of the opinions types in a network with $K = 0.1$ and $\beta = -3$.

grow but it stabilises at a low value and the three stategies don't involve different effects (fig.8).

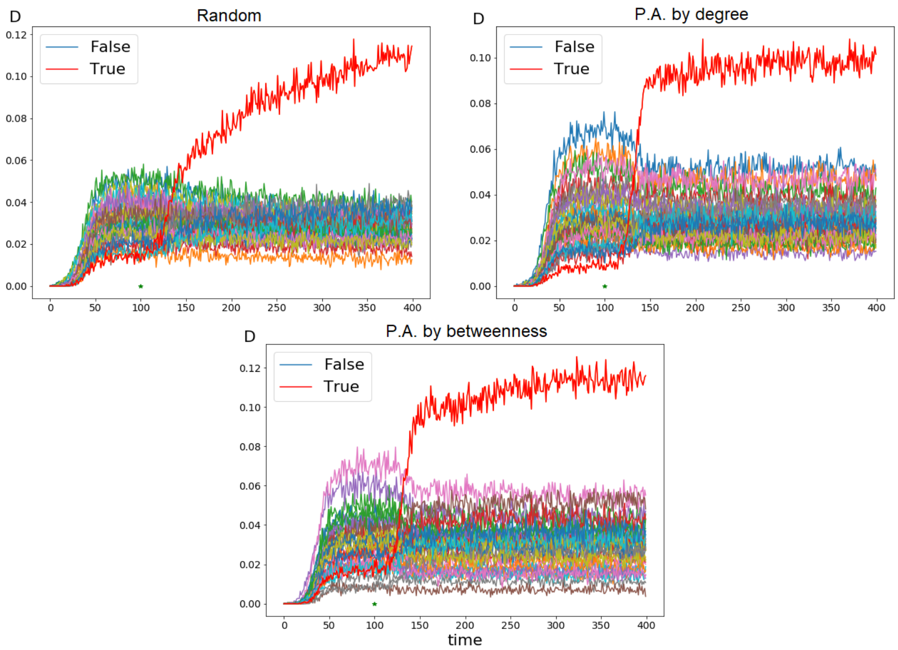

Figure 8: Evolution of the opinions concentrations for $K = 0.1$ and $\beta = -3$.

Conclusion

In conclusion, we may assert that the three different stategies may involve or not different effects depending on the values of the parameters β and K, which are characteristic parameters of the system. That is, we cannot change these two parameter. For example, a society composed of individuals with little confidence to the most connected individuals and with a high resistance to the distortions seems to be hardly infected by a new opinion. Indeed, the results we found with $\beta = -3$ and $K = 3$ and shown in fig.(2) are not very different to the result displayed in [1], in which the repaires are not inserted in the network. However, while our strategies are too dependent on the state of the system, new strategies can be tested in future to look for better results.

Links

https://mybinder.org/v2/gh/Lorenzochicchi/Rumors-Spreading-ABS/master?filepath=Run.ipynb
https://github.com/Lorenzochicchi/Rumors-Spreading-ABS

References

[1] Chao Wang, Zong Xuan Tan, Ye Ye, Lu Wang, Kang Hao Cheong, Neng-gang Xie1, *A rumor spreading model based on information entropy*, Scientific Reports (2017).

[2] A.L.Barabasi, *Network science*, Cambridge University Press, (2016).

[3] M.Newman, *Networks, an introduction*, OUP Oxford, (1984).

[4] R. Axelrod, *The complexity of cooperation*, Princeton University Press, (1997)