

## Article

# Generation of Scale-Free Assortative Networks via Newman Rewiring for Simulation of Diffusion Phenomena

Laura Di Lucchio and Giovanni Modanese \* 

Faculty of Engineering, Free University of Bozen-Bolzano, I-39100 Bolzano, Italy;  
laura.dilucchio@unibz.it

\* Correspondence: giovanni.modanese@unibz.it

**Abstract:** By collecting and expanding several numerical recipes developed in previous work, we implement an object-oriented Python code, based on the `networkX` library, for the realization of the configuration model and Newman rewiring. The software can be applied to any kind of network and “target” correlations, but it is tested with focus on scale-free networks and assortative correlations. In order to generate the degree sequence we use the method of “random hubs”, which gives networks with minimal fluctuations. For the assortative rewiring we use the simple Vazquez-Weigt matrix as a test in the case of random networks; since it does not appear to be effective in the case of scale-free networks, we subsequently turn to another recipe which generates matrices with decreasing off-diagonal elements. The rewiring procedure is also important at the theoretical level, in order to test which types of statistically acceptable correlations can actually be realized in concrete networks. From the point of view of applications, its main use is in the construction of correlated networks for the solution of dynamical or diffusion processes through an analysis of the evolution of single nodes, i.e., beyond the Heterogeneous Mean Field approximation. As an example, we report on an application to the Bass diffusion model, with calculations of the time  $t_{max}$  of the diffusion peak. The same networks can additionally be exported in environments for agent-based simulations like NetLogo.

**Keywords:** scale-free networks; assortative networks; network rewiring; agent-based models; Bass innovation diffusion model



**Citation:** Di Lucchio, L.; Modanese, G. Generation of Scale-Free Assortative Networks via Newman Rewiring for Simulation of Diffusion Phenomena. *Stats* **2024**, *7*, 220–234. <https://doi.org/10.3390/stats7010014>

Academic Editor: Wei Zhu

Received: 6 January 2024

Revised: 21 February 2024

Accepted: 22 February 2024

Published: 24 February 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Complex systems are composed of a large number of interacting elements, such that predictions of the “emergent” behavior of the system cannot be obtained from the simple knowledge of its elements, but only from suitable mathematical models which usually involve numerical solutions. This requires in turn considerable computational power, in order to handle large systems of non-linear differential equations, graphic visualization and input data organization. An approach of this kind allows one to tackle a wide class of problems, ranging from environmental to social issues, from the diffusion of innovations or diseases to the dynamics of networks and other complex structures.

Mathematical models of diffusion based on nonlinear differential equations have traditionally been very important and effective in biology for the study of epidemics and in finance and marketing for the description of technological or social innovations. In recent years, the crucial role of global networks of connections in these diffusion phenomena has become clear. Financial crises, epidemics and the spreading of disruptive innovations occur on a global scale due to the presence of efficient networks of transportation and communication. Such networks generally have the structure of complex networks, namely they are not regular but scale-free and contain some exceptionally large hubs.

As a consequence, the standard diffusion models involving few differential equations need to be re-written on scale-free networks and solved numerically. This requires an adequate computational power, for handling hundreds of coupled nonlinear differential

equations, and tools for the statistical analysis of the results (because we need to understand the emerging global behavior and cannot look at the outcome of the single equations). A good knowledge of the networks themselves is also necessary.

The configuration model is a well-known method for constructing uncorrelated networks having an assigned degree distribution. Such networks can be used, for example, as a connectivity environment for the simulation of dynamical processes or diffusion phenomena. In particular, one is often interested in scale-free networks, i.e., with degree distribution of the form  $P(k) = c_\gamma k^{-\gamma}$ , where  $P(k)$  is the probability that a randomly chosen node has degree  $k$ . There are standard commands in software packages like Python's `networkX` for generating samples from a scale-free distribution, and for performing on the corresponding "stubs" of nodes a wiring procedure uncorrelated from the degree. In Section 4.1 of this work we briefly recall this method and we compare its accuracy and efficiency with another method originally proposed in [1] ("random hubs" method) and here implemented using the object-based functions of `networkX` [2,3]. (All codes are available at <https://github.com/Ladilu/python-bass-accessible>. URL accessed on 21 February 2024).

Much less is known about the possibility of generating scale-free networks with assigned degree correlations, especially assortative networks, which are usually employed to represent social networks, financial and economic networks, and also some traffic networks and technological networks [4]. For this purpose it is possible in principle to apply to an uncorrelated scale-free network the Newman rewiring procedure with assortative target correlations [5]. It is not easy to tell in advance, however, if and when this procedure will be successful. The main purpose of this paper is to explore this issue and to find some useful common principles and numerical recipes.

At which level can degree correlations be assigned in a network? Given a statistical degree correlation matrix  $P(h|k)$  having the correct properties of positivity, normalization and pseudo-symmetry (and there are many ways of constructing such matrices), when can we say that scale-free networks with those correlations exist, or at least that a statistical ensemble of networks exists, which exhibit those correlations at an average level? Answering such questions is important for the sake of network theory itself, but also because in the so-called Heterogeneous Mean Field (HMF) approximation [6–10] several general results concerning diffusion processes have been obtained by postulating certain statistical correlations, or just by requiring that the function  $\bar{k}_{nn}(k)$  derived from the full  $P(h|k)$  has certain properties. We recall that  $\bar{k}_{nn}(k)$  represents the average degree of the nearest neighbours (whence the suffix "nn") of a node of degree  $k$ :

$$\bar{k}_{nn}(k) = \sum_{h=1}^n hP(h|k) \quad (1)$$

Here,  $n$  is the maximum degree present in the network and  $P(h|k)$  is defined as the conditional probability, for a node of degree  $k$ , of being connected to a node of degree  $h$ . In general, networks with different correlations  $P(h|k)$  may have the same  $\bar{k}_{nn}(k)$ . Rigorous results about the convergence of the  $\bar{k}_{nn}(k)$  function in random graphs with given joint degree distribution of neighbor nodes and in the configuration model have been given by [11].

A typical example is the assortative matrix by Vazquez-Weigt [12]. It has the form

$$P^{Vaz}(h|k) = (1-r) \frac{hP(h)}{\langle k \rangle} + r\delta_{hk} \quad (2)$$

and its  $\bar{k}_{nn}(k)$  function is

$$\bar{k}_{nn}^{Vaz}(k) = (1-r) \frac{\langle k^2 \rangle}{\langle k \rangle} + rk \quad (3)$$

Here, and in the following, the brackets  $\langle \dots \rangle$  denote an average weighted with the degree distribution  $P(k)$ , for ex.  $\langle k \rangle = \sum_{k=1}^n kP(k)$ .

The Newman assortativity coefficient  $r$  enters explicitly in the definition (in general  $r$  spans the range  $[-1, 1]$ , but here  $0 \leq r \leq 1$ ). Notice that this  $\bar{k}_{nn}(k)$  is linear but does not start from the origin. Using the correlation matrix above one can quickly write diffusion equations in HMF approximation, namely a system of  $n$  coupled differential equations, one for each degree class, which, e.g., for the Bass diffusion model have the form

$$\frac{dG_i(t)}{dt} = [1 - G_i(t)] \left[ p + iq \sum_{h=1}^n P(h|i) G_h(t) \right], \quad i = 1, \dots, n \quad (4)$$

With  $p = 0$  (no “publicity” or “broadcast” term) these equations reduce to those of the SI (“Susceptible-Infected”) epidemic model. However, it is not always possible to build real scale-free networks with these kinds of correlations.

The outline of the work is as follows. In Section 2, we introduce Newman’s assortative rewiring by applying it to the case of Erdős-Renyi random networks. In this case the technique works quite well already with the assortative matrix by Vazquez-Weigt. In Section 3, we show that the rewiring technique fails for Barabasi-Albert networks, partly due to their intrinsic correlations, but also to their scale-free degree distribution, as confirmed in the following, when other scale-free networks are considered. In Section 4, we look at the general scale-free case with exponents between 2 and 3. For this we first need to implement in an efficient way the configuration model in order to build the uncorrelated networks from which the rewiring process begins. We note that the Chung-Lu method has some limitations under this respect, at least when the size of the networks is not very large. For this reason we introduce the method of the random hubs (Section 4.2). In Section 5, we apply the rewiring procedure in the general scale-free case and we find that it works quite well with some target assortative matrices which were defined in previous work; they do not have an explicit expression like the Vazquez-Weigt, but their construction algorithm is included in the present code. In Section 6, we implement the network Bass model in first closure on the single nodes, i.e., with  $N$  differential equations coupled via the adjacency matrix ( $N$  is the number of nodes). We briefly discuss the outcome concerning the total diffusion curves and the peak time of the total diffusion rate, also comparing the uncorrelated and assortative case. In Section 7, we give a brief outline on future work in which we use NetLogo for agent-based simulations of the network Bass model (including extensions with modified network links and dynamics) implemented on the assortative networks built in this work.

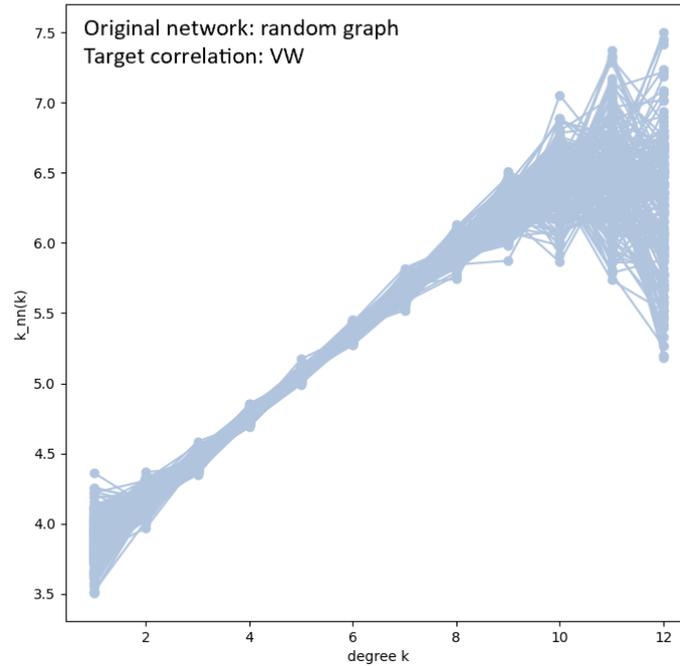
## 2. Newman Rewiring for Random Networks

Before focusing on the case of scale-free networks, we notice that the rewiring procedure does work quite well for random networks. As seen from the examples in Figures 1 and 2, one obtains, as an average in the ensemble of rewired networks, a  $\bar{k}_{nn}(k)$  function which is linear like the target, except for large  $k$ . This deviation at large  $k$  was actually expected, as a consequence of the known “structural disassortativity” effect, due to the small number of hubs, which makes impossible for hubs to connect mainly to other hubs as required in principle by assortative correlations.

Let us briefly explain how the plots in Figures 1 and 2 are obtained. This allows us to first illustrate the rewiring method in a simplified case where the degree distribution is obtained with simple networkX commands, without using the configuration model as described later for the scale-free case. First we generate an uncorrelated random network with `G = nx.gnp_random_graph(N, p, seed=...)` assigning a certain number of nodes  $N$  and a certain probability of connection  $p$ . Then we extract its degree sequence with `deg_sequence=sorted((d for n, d in G.degree()), reverse=True)`. The degree sequence is passed to a custom-built function called `random_reference` which performs a number of rewiring cycles fixed by the variable `num_cycles` (typically a few hundreds for better statistics, although already after the first few cycles correlations become close to the target as far as possible). The number of rewiring steps in each cycle is defined by the

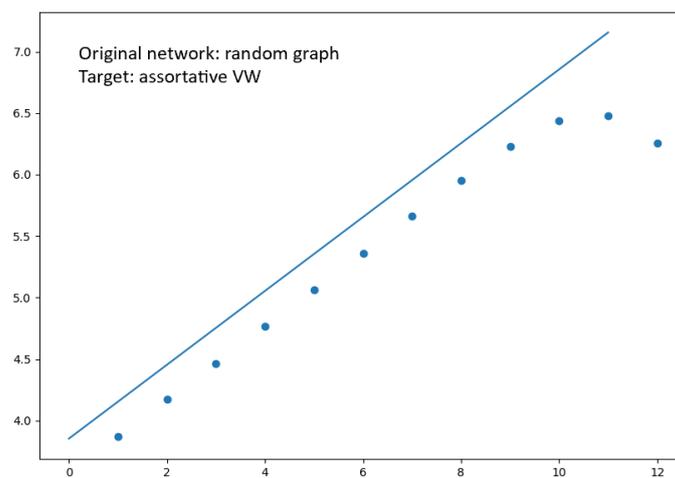
function parameter  $niter$ , which should be adjusted in such a way that each link of the network is cut and re-connected at least ten times in a cycle.

$k_{nn}$  functions for all rewiring cycles



**Figure 1.** The  $\bar{k}_{nn}$  functions of an ensemble of 200 networks obtained by Newman rewiring with target correlations of the Vazquez-Weigt type ( $r = 0.3$ ), starting from a random network of 4000 nodes, probability connection 0.001. The Newman coefficient of the ensemble is  $r = 0.20 \pm 0.01$ . The giant component after the last rewiring is 97.9%. In each of the 200 rewiring sub-cycles, the number of accepted rewiring steps was about  $2.18 \times 10^5$ .

Ensemble average of  $k_{nn}$  vs. assortative target



**Figure 2.** Average of the  $\bar{k}_{nn}$  functions of the ensemble of random networks shown in Figure 1 (dotted plot). The continuous line shows the target  $\bar{k}_{nn}(k)$  function (Equation (3), with  $r = 0.3$ ).

The general working principle of the Newman rewiring method is as follows. First, one chooses at random in the list of the links of the network two links  $(a, b)$  and  $(c, d)$ , between nodes  $a, b, c, d$ , with excess degrees  $A, B, C, D$ . Then one computes the probability

$E_1$  of these links according to a “target” correlation matrix  $e^0$ , namely  $E_1 = e_{AB}^0 e_{CD}^0$  and the analogous probability  $E_2$  for the exchanged links, i.e., for the couple  $(a, c)$  and  $(b, d)$ . After this, a sort of Metropolis-Monte Carlo criterion is applied: if  $E_2 > E_1$ , then the rewiring is performed with probability 1, otherwise it is performed with a probability proportional to the ratio  $E_1/E_2$ .

In our code, the target correlations are defined in terms of the nodes degrees by assigning  $P(h|k)$  before the rewiring cycles, and translated into the correlations  $e_{ij}^0$  of the excess degrees with the formula  $e_{ij}^0 = P(i+1|j+1)(j+1)P(j+1)/\langle k \rangle$ , written as `e0[i, j]=(Phk[i+1, j+1]*(j+1)*probability_seq[j+1])/aver_degree`. The array `probability_seq` gives the normalized degree distribution of the random graph  $G$  and is obtained via the function `degree_dist(G)`, which in turn uses the command `nx.degree_histogram(G)`. Note that the indices  $i$  and  $j$  are in the range  $0, \dots, n-1$ , while the indices  $h$  and  $k$  are in the range  $1, \dots, n$ .

The  $\bar{k}_{nn}(k)$  function of the target correlation matrix is also computed, according to the definition (1), for comparison with the average  $\bar{k}_{nn}(k)$  of the rewired ensemble. A counter variable returns for each sub-cycle the number of rewiring steps which have been accepted. (See typical values in Figures 1 and 2).

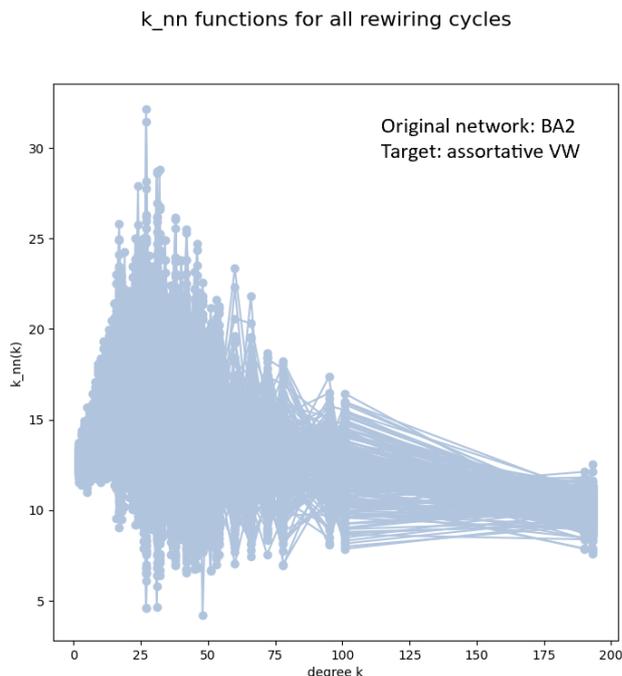
The choice of the edges for the rewiring is not performed by creating the edge list, but instead the module `discrete_sequence` is called. The latter returns a sample sequence of length  $n$  from the discrete cumulative distribution of the degrees. Then two numbers are picked from the sequence; if they are the same, the code skips. The choice among the neighbours for the rewiring is made by means of the built-in python module `random`, which allows to generate and choose random numbers in a list. The vertices are selected so that they are different from each other. Before finally performing the rewiring, after the condition is proven, another control is made, in order to check whether the edges that are going to be added already exist; in fact, they must not be already present in the graph, or the `networkX` package will not perform the rearrangement as prescribed.

At the end of each rewiring sub-cycle, the  $r$  coefficient of the network is computed and stored for a final evaluation of the average and standard deviation of  $r$  over all sub-cycles. As we shall see, this gives only a rough “integral” check of the convergence of the degree correlations to the target correlations.

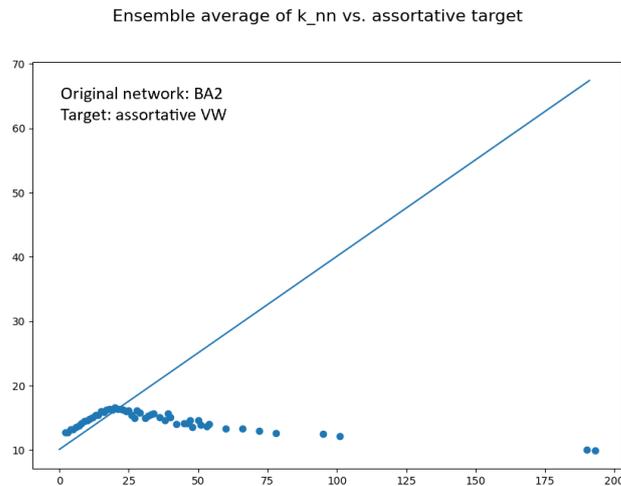
Moreover, at the end of each sub-cycle the  $\bar{k}_{nn}(k)$  function of the current rewired network  $G_2$  is computed with the command `knn=nx.average_degree_connectivity(G2, ...)` and accumulated into a dictionary `knn1=dict(sorted(knn.items()))` for subsequent evaluation of its final average. All the graphs of  $\bar{k}_{nn}(k)$  for each sub-cycle are plotted together in a “cloud” graph which gives a visual impression of their fluctuations. Finally, the ensemble average of  $\bar{k}_{nn}(k)$  and the target  $\bar{k}_{nn}(k)$  are plotted for comparison. The final size of the giant connected component is also computed.

### 3. Newman Rewiring for Barabasi-Albert Networks

A rewiring procedure similar to the one starting from random networks can be applied to Barabasi-Albert (BA) networks generated by preferential attachment to  $\alpha$  existing nodes with the method `G=nx.barabasi_albert_graph(N, alpha)`. In this case, however, the rewired network turns out to be not assortative, and so the procedure fails. See Figures 3 and 4. This is due to a limitation of the Newman rewiring when applied to scale-free networks (see Section 5) and, in addition, to the fact that BA networks have intrinsic degree correlations which cannot apparently adapt well to the Vazquez-Weigt correlation matrix. In fact, although the  $r$  coefficient of BA networks is close to zero, their  $\bar{k}_{nn}(k)$  function is decreasing at small degrees and increasing at large degrees.



**Figure 3.** The  $\bar{k}_{nn}$  functions of an ensemble of 200 networks obtained by Newman rewiring with target correlations of the Vazquez-Weigt type ( $r = 0.3$ ), starting from a BA-2 network of 5000 nodes. The Newman coefficient of the ensemble is  $r = -0.07 \pm 0.06$ . The giant component after the last rewiring is 100%. In each of the 200 rewiring sub-cycles, the number of accepted rewiring steps was about  $1.92 \times 10^5$ .



**Figure 4.** Average of the  $\bar{k}_{nn}$  functions of the ensemble of BA networks shown in Figure 3 (dotted plot). The continuous line shows the target  $\bar{k}_{nn}(k)$  function (Equation (3), with  $r = 0.3$ ).

We notice in this regard that the Newman rewiring works quite well, on the contrary, when one starts from an uncorrelated scale-free network with  $\gamma = 3$  and takes the BA correlation matrix as target [1].

The matching between target correlations and correlations of the rewired network is usually assessed by comparing the respective  $\bar{k}_{nn}$  functions, at least for degrees which are not too large (the structural disassortativity prevents any real  $\bar{k}_{nn}$  to increase indefinitely like a theoretical target assortative  $\bar{k}_{nn}$ , because there are simply not enough hubs in a scale-free network for connecting hubs preferably with hubs). A common phenomenon occurring in scale-free networks, however, is the strong fragmentation of the rewired network, especially if the smallest degree is  $k_{min} = 1$ . In that case, the nodes with  $k = 1$  represent a vast majority

of all nodes, and if  $P(1|1) \neq 0$  in the target correlations, then a large number of isolated couples are formed; if  $P(2|1) \neq 0$  a large number of isolated triples, and so on, leaving a small principal connected component. This phenomenon can be observed using the method `G2.subgraph(max(nx.connected_components(G2), key=len))`. The situation improves markedly if the minimum degree is equal to 2. In this case, the principal component is really a “giant” component, close to 100% of all nodes.

#### 4. Configuration Model for Scale-Free Networks

We are now going to illustrate a method for constructing uncorrelated scale-free networks with given exponent  $2 < \gamma \leq 3$  and for applying Newman rewiring to them in order to approach some target assortative correlations as well as possible. It is known that many relevant real networks are assortative, in particular social and economic networks; it is therefore very useful, in order to simulate dynamical processes on this kind of network, to generate them algorithmically. This issue is also interesting in itself, for “foundational” purposes, in order to understand if the scale-free property is fully compatible with assortative (or possibly disassortative) correlations.

The established paradigm of scale-free complex networks has been recently re-discussed [13,14] and it turns out that in many cases it is impossible, due to statistical errors and fit uncertainties, to state whether certain networks are scale-free or not. Since the preferential attachment method leads to an exponent  $\gamma = 3$ , for the generation of scale-free networks with a different exponent it is usually necessary to employ the configuration model. In Section 5.2 we shall present a novel alternative method [15] in which the excess degree correlations  $e_{jk}$  are assigned at the beginning and a good approximation of scale-free degree distributions is obtained as a consequence.

##### 4.1. Test of the Implementation of the Chung and Lu Model

Before describing our implementation of the configuration model, we will discuss the limitations of a method already available in `networkX` and based on the model by Chung and Lu [16,17]. This is a weaker version of the configuration model, treated also by Newman in his book [18] in comparison to the standard configuration model with “fixed degree distribution”, in which it is possible to analytically prove several general properties.

Suppose we want to build a network with  $N$  nodes, starting from a list of  $N$  values for the degrees of the nodes, randomly extracted from a power law distribution with given exponent  $\gamma$ . It is possible to generate such a list with various stochastic methods, in a similar way as for the generation of samples of the normal distribution or of other known distributions. A general procedure makes use of a probability transformation method. One defines first a vector  $F_k = \sum_{j=1}^k P(j)$  where  $k = 1, \dots, n$  and  $P(j)$  denotes the normalized degree distribution (the power law in our case, with  $n$  the maximum degree). The values of  $F_k$  define breakpoints of the unit interval  $(0, 1)$ . After generating a random number  $\xi$  in this interval, a new node is introduced into the list with degree  $k$  if  $F_{k-1} < \xi < F_k$ , and the procedure is repeated  $N$  times.

In `networkX` there is an auxiliary function which does just this, namely `S=nx.utils.powerlaw_sequence(N, gamma)`. The list of values obtained is real, in the interval  $(1, +\infty)$ . For transforming this into a list of degrees, a proper rounding to integers is necessary. It can be checked that for large  $N$  the degree succession obtained respects well the scale-free criterion of the ratio of probabilities, namely  $P(k_1)/P(k_2) = (k_2/k_1)^\gamma$ .

In the Chung and Lu method (`networkX` command: `G = nx.expected_degree_graph(S, selfloops=False)`) one actually starts from the not-rounded scale-free degree succession and connects nodes  $i$  and  $j$  having degrees  $k_i$  and  $k_j$  ( $i, j = 1, \dots, N$ ) with probability proportional to  $k_i k_j$ . It is possible to prove that the network obtained must be scale-free in the limit  $N \rightarrow \infty$ . The algorithm by Miller and Hagberg employed by `networkX` [17] runs in an expected time of order  $N$ . It is clear that the exact degrees of the nodes cannot be assigned a priori like in the standard configuration model, but become random variables themselves.

Our numerical tests of this method (code at <https://github.com/Ladilu/python-bass-accessible> (URL accessed on 21 February 2024)) give, at least for values  $N$  up to  $10^5$ , distributions of the final degrees which deviate significantly from the scale-free criterion (while the initial degree list, for the same values of  $N$ , is accurately scale-free).

#### 4.2. Random Hubs Method

Therefore, we chose to implement the standard configuration model in a custom function named `configuration_model`, in which a multigraph is constructed starting from the assigned degree sequence. In this work we only made use of the code for an undirected network. Once the degree sequence is given, a list containing all nodes is built, each node being repeated as many times as its degree determines. Then the list is reshuffled by means of the random module functions and split in half. The vertices for the edges are picked in an ordered way from the two halves of the list, so that the method cannot go back and accidentally select one node more times than allowed. This is possible with the built-in methods of the Python `itertools` package, which are programmed for operating on iterable objects such as arrays and lists. The edges are finally added to the graph until the list is finished, in such a way that the original degrees of the nodes are preserved.

This function works with any assigned degree sequence. For example, it is also possible to obtain from `networkX` a random graph or a BA network like in the rewiring examples previously discussed, then extract their degree sequence and apply the configuration model function to it, obtaining an uncorrelated network with the same degree sequence (of course, the random graph should already be uncorrelated). In order to build a scale-free network one could apply the function `configuration_model` to a degree sequence generated with the command `S=nx.utils.powerlaw_sequence(N, gamma)` as described above, but we chose instead to define a scale-free degree sequence with “minimal fluctuations” through a technique that we call the “random hubs method”. This allows to build networks of relatively small size which are as close as possible to an ideal scale-free distribution, and works as follows.

If the average number of nodes with degree  $k$  is smaller than 1, i.e.,  $NP(k) = X < 1$ , then a node with this degree will be created with probability  $X$ . Extending the procedure to all degrees, a random variable  $\zeta \in (0, 1)$  is generated for each value of  $k$ , and then denoting by  $Int(NP(k))$  the integer part of  $NP(k)$  and by  $Dec(NP(k))$  its decimal part, one sets the number  $N_k$  of nodes with degree  $k$  to  $N_k = Int(NP(k))$  if  $\zeta > Dec(NP(k))$  and  $N_k = Int(NP(k)) + 1$  if  $\zeta < Dec(NP(k))$ . The total number of nodes is therefore not fixed, with random variations of 1 for each degree.

With the random hubs method we thus obtain a list of values (called `s1` in the code) which shows how many nodes of each degree, starting from  $k_{min}$ , must be present in the network in order to satisfy the power law—except for the minimal fluctuations mentioned above. For the highest degrees, most values will be zero, because only a few hubs will actually be present. This is what one also observes in the degree sequence of real scale-free networks, e.g., BA networks: some hubs, with random degrees, are present, and the remaining high degrees are missing.

The maximum degree  $n$  considered for the list `s1` is connected to the number of nodes  $N$  through the Dorogotsev-Mendes relation. After the list has been built, it is necessary to check that the total number is even, otherwise the random wiring procedure cannot connect all nodes respecting their degrees. From the list `s1` another list is built with the command `list_of_nodes = _to_stublist_kmin(s1, kmin)`, giving the degree sequence of the nodes. For example, if we are generating a network with exponent  $\gamma = 2$  (this extreme value of  $\gamma$  is normally excluded, and taken here only for simplicity) and there are 100 nodes with degree 1, the nodes of degree 2 must be 25, or let us say 24, 25 or 26, depending on the integer rounding and on the fluctuations due to the term  $\zeta$ . The degree sequence `list_of_nodes` will start with 100 elements equal to 1, then 25 elements equal to 2, and so on. This degree sequence is fed to the wiring function `configuration_model`, which

creates the “stubs” of the network and adds links at random between them, until each node reaches the degree defined by the degree sequence.

### 5. Results of the Rewiring Procedure for Scale-Free Networks

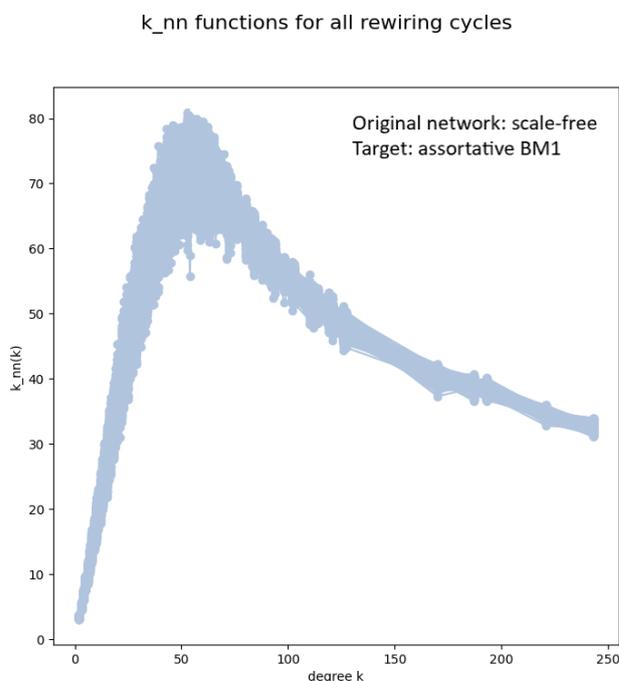
If we use as target the Vazquez-Weigt correlations, the average  $r$  coefficient of the resulting network ensemble is very close to zero, independently from the  $r$  parameter of the target. The ensemble average of  $\bar{k}_{nn}$  is linearly increasing only for small values of  $k$ , similarly to what happens for the rewiring of BA networks. We can thus conclude that the Vazquez-Weigt assortative correlations are incompatible with a scale-free degree distribution, not only due to structural disassortativity at large degrees, but for all degrees except in a small range close to  $k_{min}$ .

One might wonder whether it is possible at all to obtain positive ensemble values of  $r$  by rewiring a scale-free network. The answer is affirmative, as shown by Xulvi and Brunet [19,20] with their “empirical” rewiring method (links pairs are exchanged if the differences between their degrees increase) and also in [1,21] with a maximally assortative rewiring based on a formula for the variation  $\Delta r$ .

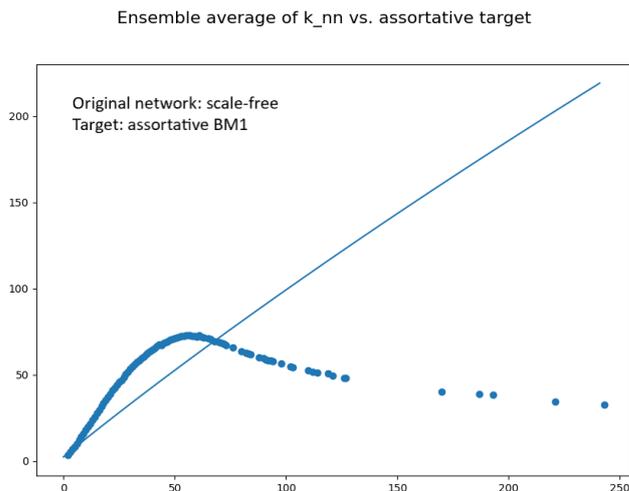
It is also possible to obtain assortative scale-free networks by assortative rewiring with correlations different from the Vazquez-Weigt recipe. This has the advantage, compared to empirical rewiring or maximally assortative rewiring, of maintaining some control about the functional form of the correlations. We shall illustrate two methods, denoted for brevity as BM1 and BM2.

#### 5.1. BM1 Assortative Matrices

This set of matrices is built through a procedure in which the matrix elements of  $P(h|k)$  are chosen to be largest on the main diagonal and decreasing elsewhere, then properly adjusted in order to satisfy the Network Closure Condition and normalized column-by-column. By performing the Newman rewiring with these kind of matrices as targets, one obtains ensembles with  $r \simeq 0.2 - 0.3$  and average  $\bar{k}_{nn}$  as shown in Figure 5 and 6.



**Figure 5.** The  $\bar{k}_{nn}$  functions of an ensemble of 400 networks obtained by Newman rewiring with target correlations of the BM1 type, starting from an uncorrelated scale-free network of 10,000 nodes,  $\gamma = 2.5$ . The Newman coefficient of the ensemble is  $r = 0.337 \pm 0.002$ . The giant component after the last rewiring is 100%. In each of the 400 rewiring sub-cycles, the number of accepted rewiring steps was about  $1.45 \times 10^5$ .



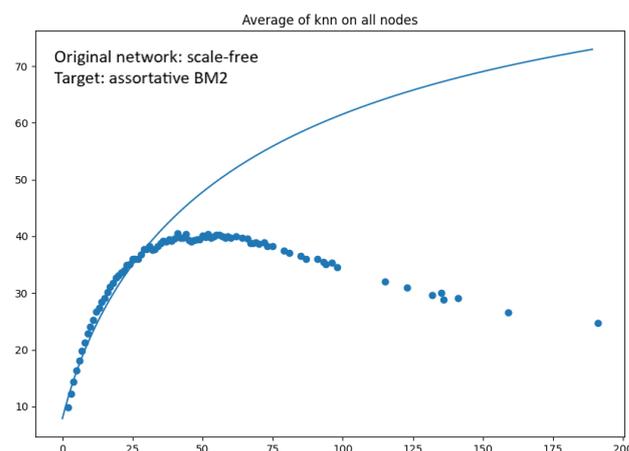
**Figure 6.** Average of the  $\bar{k}_{nn}$  functions of the ensemble of random networks shown in Figure 5 (dotted plot). The continuous line shows the target  $\bar{k}_{nn}(k)$  function.

### 5.2. BM2 Assortative Matrices

An alternative procedure for constructing assortative matrices [15] is to define them with an explicit formula in terms of the excess degrees, e.g.,

$$e_{jk} = \frac{c_\delta}{(1 + j + k)^\delta}, \quad j, k = 0, \dots, n - 1 \tag{5}$$

where  $c_\delta$  is a normalization constant. From this one can obtain the degree distribution  $P(k)$  in the form of a series and check that it behaves with good approximation as a scale-free distribution. The  $P(h|k)$  matrix is obtained from  $e_{jk}$  and  $P(k)$  as usual. The target  $\bar{k}_{nn}$  function is, unlike in the previous cases, increasing but definitely non-linear (see Figure 7). The disadvantage of this method is that the scale-free exponent  $\gamma$  cannot be fixed a priori but depends on the  $\delta$  exponent in the definition of  $e_{jk}$ .



**Figure 7.** Average of the  $\bar{k}_{nn}$  functions of an ensemble of 200 networks obtained by Newman rewiring with target correlations of the type  $e_{jk} = c_\delta / (1 + j + k)^\delta$ , with  $\delta = 2.2$ , starting from an almost scale-free uncorrelated network ( $\gamma \simeq 2.42$ ) with degree distribution derived from  $e_{jk}$ . The number of nodes is  $N = 5000$ . The Newman coefficient of the ensemble is  $r = 0.176 \pm 0.005$ . The giant component after the last rewiring is 99.9%. In each of the 200 rewiring sub-cycles, the number of accepted rewiring steps was about  $2.8 \times 10^5$ .

## 6. Numerical Solutions of the Bass Diffusion Equation on Assortative Rewired Networks

As mentioned in the Introduction, when we have a concrete realization (actually, a statistical ensemble of realizations) of a network having assigned degree distribution and correlations as close as possible to a certain theoretical target, we can solve systems of differential equations which describe a certain dynamical process unfolding on the network. The equations in the system are as many as the nodes. This means that the behavior of single agents is described much more accurately than in the HMF approximation, at the cost of course of a greater computational complexity which limits the network size to a few thousands of nodes.

The results of the numerical solutions still need to be aggregated in order to examine them. Nevertheless, one observes some interesting differences in comparison to the HMF results. To some extent these differences are due, in the scale-free case, to the real random occurrence of large hubs in the network, while in the HMF approximation all hubs contribute “virtually” to dynamics, each one with a small probability [22–25].

We have focused our attention on the network Bass model, which written on the single nodes in first closure [18,26] takes the form

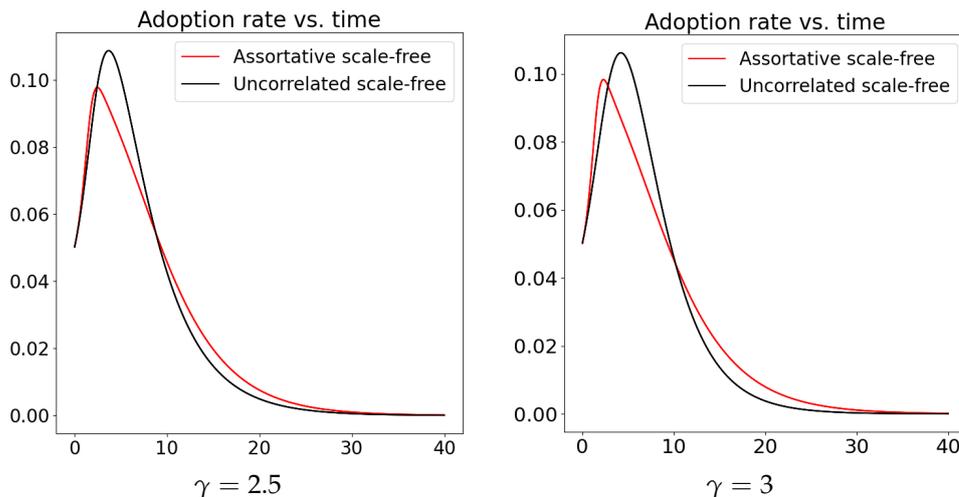
$$\frac{dX_l(t)}{dt} = (1 - X_l(t)) \left[ p + q \sum_{j=1}^N A_{lj} X_j(t) \right], \quad l = 1, \dots, N \quad (6)$$

where  $A$  is the adjacency matrix of the network and the variable  $X_l$  has to be understood as the expectation  $\langle \xi_l \rangle$  of the non-adoption ( $\xi_l = 0$ ) or adoption ( $\xi_l = 1$ ) state of node  $l$  over many stochastic evolutions of the system. The code for the solution is available at <https://github.com/Ladilu/python-bass-accessible> (URL accessed on 21 February 2024) and makes use of the Python method `odeint` which can be imported from the package `scipy.integrate`. The adjacency matrix is handled by `networkX` as a sparse matrix, with the commands `m = nx.adjacency_matrix(G)`, `A=m.todense()`. Like in the HMF equations, the imitation coefficient  $q$  is normalized to the average connectivity  $\langle k \rangle$ , in such a way to allow comparisons between networks of different kinds while re-scaling the dominant effect of the average degree on diffusion times.

The Bass model is reduced to a standard SI epidemic model when the publicity coefficient  $p$  is set to zero; however, the  $p$ -term allows one to produce a meaningful diffusion dynamics also starting from null initial conditions, i.e., without initial adopting nodes (the equivalent of infected nodes in the SI model). One can thus define a characteristic peak adoption time  $t_{max}$  as the time at which the adoption rate  $f(t)$  is maximum.  $f(t)$  is the derivative of  $F(t)$ , which represents the fraction of total adopters as a function of time and has a typical S-shaped plot ( $F(t) = \sum_{l=1}^N X_l(t)$ ). It is also possible to define other characteristic times, like the takeoff time, see [15].

In general, one observes that for scale-free networks which have the same critical exponent  $\gamma$  and have been rewired using the same target correlations,  $t_{max}$  depends strongly on the degree  $k_{max}$  of the largest hub present in the network (at least for the networks with 1000 nodes examined). The presence of a super-hub makes diffusion visibly faster, as could be intuitively expected, and it also appears to make the assortative rewiring process less efficient, leading to smaller values of the final  $r$  in correspondence of the same target correlations.

If we want to assess the effect of assortative correlations we thus need to compare network realizations which have similar  $k_{max}$ . Then it turns out that in assortative networks  $t_{max}$  is systematically smaller than in uncorrelated networks; in addition, the diffusion rate reaches its peak earlier, and decreases faster after the peak. See Figure 8 and Tables 1 and 2. This outcome differs from the predictions of HMF theory, according to which uncorrelated networks give in general a smaller  $t_{max}$  than assortative networks (see [1] and refs. therein).



**Figure 8.** Total adoption rates as a function of time for a network Bass model running on uncorrelated and assortative scale-free networks with 1000 nodes (system of coupled differential equations for the single nodes).

**Table 1.** Examples of peak times  $t_{max}$  observed in assortative networks with correlations of the BM1 type,  $\gamma = 2.5$ ,  $N = 1000$ ,  $k_{min} = 2$ . Note the strong dependence on the degree  $k_{max}$  of the largest hub, as discussed in the text.

Assortative			
$k_{max}$	$\langle k \rangle$	$t_{max}$	$r$
90	4.2	2.4	0.23
98	4.2	2.4	0.19
132	4.8	2.0	0.09
149	4.5	2.0	0.11

**Table 2.** Same as in Table 1, but with uncorrelated networks. For comparable  $k_{max}$ , the peak times are larger than with assortative networks. In the rewiring code, a Vazquez-Weigt matrix with  $r = 0$  is chosen as target. The resulting effective  $r$  is slightly negative due to structural disassortativity.

Uncorrelated			
$k_{max}$	$\langle k \rangle$	$t_{max}$	$r$
30	3.1	4.2	-0.03
33	3.1	4.1	0.02
122	4.3	3.2	-0.08
126	4.3	3.1	-0.08

The calculations were carried out on a quad-core processor computer with 11th Gen Intel(R) Core(TM) i5 processor, with 2.42 GHz frequency. The system type was a Windows 11 64-bit operating system, with an x64-based processor.

### 7. Outlook: NetLogo Simulations

In order to reconstruct dynamically the evolution of the Bass diffusion on an assortative network, preliminary agent-based simulations have been carried out with NetLogo. The code is also available at <https://github.com/Ladilu/python-bass-accessible> (URL accessed on 21 February 2024) and it consists of a few procedures. NetLogo allows the uploading of networks created by means of applying the Python rewiring algorithm with the

criterion of setting a target BM1 matrix to an initial graph  $G$ . The graph  $G$  is built constructing the degree sequence from a power law and then applying the `configuration_model` procedure. The Python code is able to produce a `graphml` output for the nodes and links of the network. There exists a corresponding method in the NetLogo code, `wire_graphml`, which loads the `graphml` file and then the procedure `create-network` uses the method to effectively reconstruct the corresponding network. Another method in the procedure `create-network` is `setup-nodes`, which places the network in the screen within the area of a circle so that the coordinates of the points corresponding to the agents are well readable. There exists the possibility of setting an arbitrary number of seed adopters. In the procedure `adopt` the Bass model is implemented. The  $p$  parameter for the broadcast influence is compared to a random variable, so that according to its value the agents which are subject to publicity adoptions are given the red color. The  $q$  parameter for a given node is set using the social influence parameter normalized to the number of neighbors divided by the mean degree of the network, which is computed separately. The `adopt` procedure is then used in the `go` procedure that can be set in the interface to make the diffusion process start in the simulation. The time-based approach is chosen, by means of resetting the ticks count before any new start of the Bass model simulation. The process can be monitored on the screen as represented in Figure 9. A detailed description of the simulations and their results will be presented in a forthcoming paper.

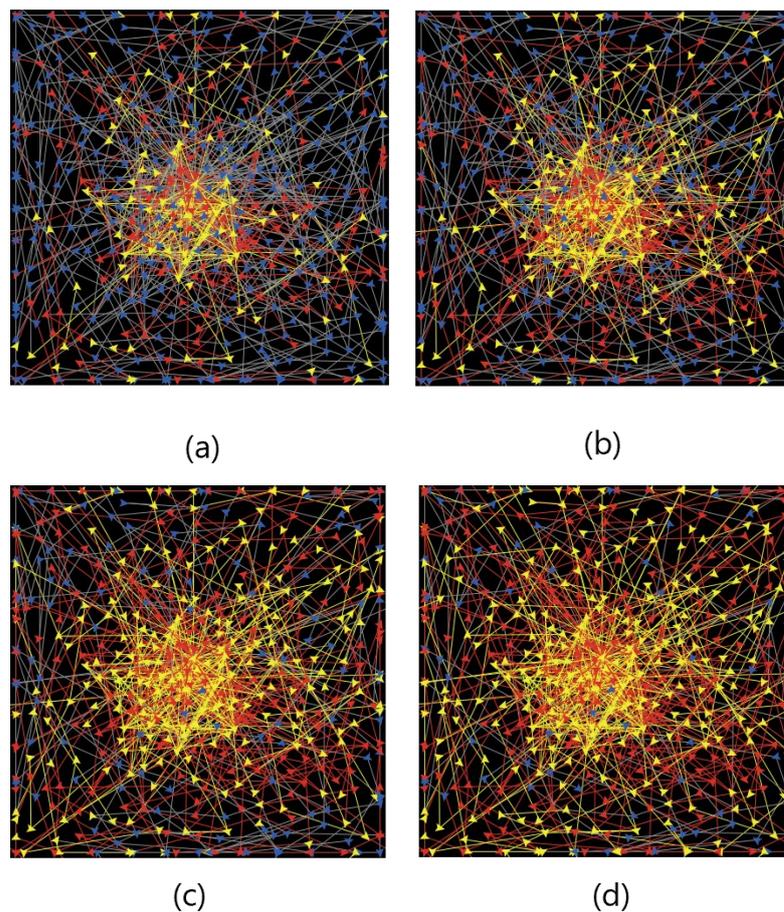
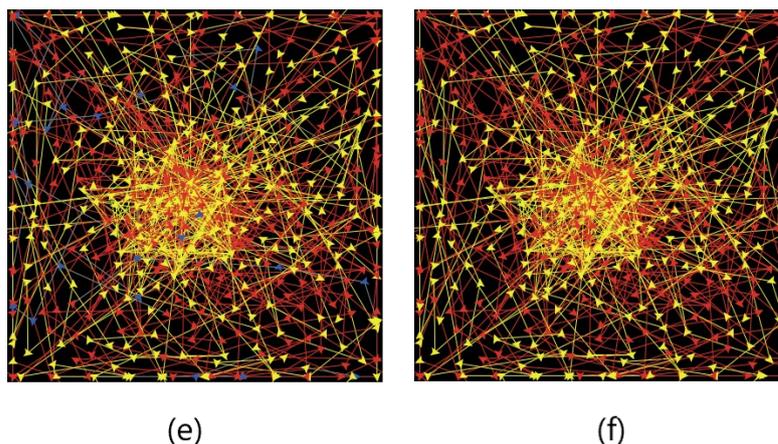


Figure 9. Cont.



**Figure 9.** Some NetLogo views of the agent dynamics for Bass adoption in an assortative scale-free network. The agents that have not adopted are initially colored in blue, then “broadcast” (publicity) adoptions are represented by red color and “network” (imitation) adoptions by yellow color. The network has 500 nodes,  $\gamma = 2.5$  and is obtained via Newman rewiring with a BM1 target matrix. The views are taken at approximate times (“ticks”) (a) 8, (b) 16, (c) 24, (d) 32, (e) 40, (f) 48.

## 8. Conclusions

The numerical code that has been developed, tested and described in this work allows one to generate realizations of networks in the form of statistical ensembles, using the configuration model, the random hubs method and Newman rewiring with various possible target correlations. It is a flexible code, which can be easily adapted by the user. The code handles the graphs as Python objects, so that it is possible to employ at any stage the advanced functions of the networkX extension, e.g., for evaluating clustering, centrality, path lengths and other measures of interest. The capability to generate full ensembles of networks with pre-defined degree distribution and correlations makes possible useful numerical experiments about the statistical properties of these ensembles.

In the case of scale-free networks with assortative target correlations (which are of special interest for diffusion studies in socio-economic systems), one observes the formation of a strongly connected core composed of a large number of “intermediate” hubs with a degree of the order of 30–40% of the maximum degree present in the network. The hubs in the core are preferably connected among themselves, but also of course with smaller hubs which reach out to the network periphery. This detailed structure is visible in the ensemble average of the  $\bar{k}_{nn}(k)$  function, but not in the global assortativity coefficient. At the same time, it is possible to examine the statistical properties of  $\bar{k}_{nn}(k)$  at large  $k$ , which turn out to be different from the target function used in the mean-field approximation. These structural differences show up in the dynamics of diffusion processes, when analysed in terms of the single nodes instead of degree classes. Similar effects are expected when the networks are used for agent-based simulations, as explained in the Outlook section and in more detail in forthcoming work.

**Author Contributions:** Conceptualization, G.M.; formal analysis, L.D.L., G.M.; software, L.D.L.; writing—original draft preparation, L.D.L., G.M.; writing—review and editing, L.D.L., G.M. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was financially supported by the Free University of Bozen-Bolzano with the research project NMCSYS-TN2815.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Acknowledgments:** G.M. is a member of INdAM (Istituto Nazionale di Alta Matematica), section Mathematical Physics.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Bertotti, M.L.; Modanese, G. The configuration model for Barabasi-Albert networks. *Appl. Netw. Sci.* **2019**, *4*, 32. [[CrossRef](#)]
2. Chult, D.S.; Hagberg, A.; Swart, P. *Exploring Network Structure, Dynamics, and Function Using NetworkX*; Report LA-UR-08-05495; Los Alamos National Lab.: Los Alamos, NM, USA, 2008.
3. Platt, E.L. *Network Science with Python and NetworkX Quick Start Guide: Explore and Visualize Network Data Effectively*; Packt Publishing Ltd.: Birmingham, UK, 2019.
4. Noldus, R.; Van Mieghem, P. Assortativity in complex networks. *J. Complex Netw.* **2015**, *3*, 507–542. [[CrossRef](#)]
5. Newman, M. Mixing patterns in networks. *Phys. Rev. E* **2003**, *67*, 026126. [[CrossRef](#)]
6. Boguñá, M.; Pastor-Satorras, R.; Vespignani, A. Epidemic spreading in complex networks with degree correlations. In *Statistical Mechanics of Complex Networks*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 127–147.
7. Moreno, Y.; Vazquez, A. Disease spreading in structured scale-free networks. *Eur. Phys. J. B* **2003**, *31*, 265–271. [[CrossRef](#)]
8. Goldenberg, J.; Han, S.; Lehmann, D.R.; Hong, J.W. The role of hubs in the adoption process. *J. Mark.* **2009**, *73*, 1–13. [[CrossRef](#)]
9. Vespignani, A. Modelling dynamical processes in complex socio-technical systems. *Nat. Phys.* **2012**, *8*, 32–39. [[CrossRef](#)]
10. Pastor-Satorras, R.; Castellano, C.; Van Mieghem, P.; Vespignani, A. Epidemic processes in complex networks. *Rev. Mod. Phys.* **2015**, *87*, 925. [[CrossRef](#)]
11. Yao, D.; van der Hoorn, P.; Litvak, N. Average nearest neighbor degrees in scale-free networks. *Internet Math.* **2018**, *2018*, 1–38.
12. Vázquez, A.; Weigt, M. Computational complexity arising from degree correlations in networks. *Phys. Rev. E* **2003**, *67*, 027101. [[CrossRef](#)]
13. Broido, A.D.; Clauset, A. Scale-free networks are rare. *Nat. Commun.* **2019**, *10*, 1017. [[CrossRef](#)]
14. Holme, P. Rare and everywhere: Perspectives on scale-free networks. *Nat. Commun.* **2019**, *10*, 1016. [[CrossRef](#)]
15. Bertotti, M.L.; Modanese, G. On the evaluation of the takeoff time and of the peak time for innovation diffusion on assortative networks. *Math. Comput. Model. Dyn. Syst.* **2019**, *25*, 482–498. [[CrossRef](#)]
16. Chung, F.; Lu, L. Connected components in random graphs with given expected degree sequences. *Ann. Comb.* **2002**, *6*, 125–145. [[CrossRef](#)]
17. Miller, J.C.; Hagberg, A. Efficient generation of networks with given expected degrees. In *Proceedings of the International Workshop on Algorithms and Models for the Web-Graph, Brisbane, QLD, Australia, 6–7 July 2019*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 115–126.
18. Newman, M. *Networks: An Introduction*; Oxford University Press: Oxford, UK, 2010.
19. Xulvi-Brunet, R.; Sokolov, I. Reshuffling scale-free networks: From random to assortative. *Phys. Rev. E* **2004**, *70*, 066102. [[CrossRef](#)]
20. Boccaletti, S.; Latora, V.; Moreno, Y.; Chavez, M.; Hwang, D.-U. Complex networks: Structure and dynamics. *Phys. Rep.* **2006**, *424*, 175–308. [[CrossRef](#)]
21. Bertotti, M.L.; Modanese, G. Network rewiring in the r-K plane. *Entropy* **2020**, *22*, 653. [[CrossRef](#)] [[PubMed](#)]
22. Fotouhi, B.; Rabbat, M.G. Degree correlation in scale-free graphs. *Eur. Phys. J. B* **2013**, *86*, 1–19. [[CrossRef](#)]
23. Cimini, G.; Squartini, T.; Saracco, F.; Garlaschelli, D.; Gabrielli, A.; Caldarelli, G. The statistical physics of real-world networks. *Nat. Rev. Phys.* **2019**, *1*, 58–71. [[CrossRef](#)]
24. Gray, C.; Mitchell, L.; Roughan, M. Generating connected random graphs. *J. Complex Netw.* **2019**, *7*, 896–912. [[CrossRef](#)]
25. Bertotti, M.L.; Brunner, J.; Modanese, G. The Bass diffusion model on networks with correlations and inhomogeneous advertising. *Chaos Solitons Fractals* **2016**, *90*, 55–63. [[CrossRef](#)]
26. Gleeson, J.P. High-accuracy approximation of binary-state dynamics on networks. *Phys. Rev. Lett.* **2011**, *107*, 068701. [[CrossRef](#)] [[PubMed](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.