

# Memory As an Organizer of Dynamic Modules In a Network of Potential Interactions

Mesut Yucel<sup>1</sup> and Uri Hershberg<sup>2\*</sup>

**Abstract**— Networks are amongst the most common tools to describe interactions in a spatial form. One failing of networks is that often there is little differentiation between the links of potential interactions and those that are actually active in a specific time or space. In this paper, we aim to show how spatial organization of a network of potential interactions effect the pattern formation and reproducibility of sub-patterns in the network. We further show that even a short memory of the path taken can radically effect the network characteristics and placement of the most common modules.

While memory helps the reproducibility of some dynamic patterns by creating a bias in the direction of learned patterns, the spatial organization does this by limiting the individual stochasticity. In both cases these forces increase the probability of a pattern to recur. We see that memory effects are highly dependent on the connectivity of the network. In high connectivity networks memory cannot effect the results much. This makes sense when the connectivity is seen as a measure of the stochastic potential of the network, e. g., if there are many connections from a node, then a memory remembering a specific path passing through that node will not create much bias in the directional preferences.

Many real networks are sparse. In light of this observation and our confirming simulation results, we may conclude that there should be a *connectivity constraint* for a network system to be able to use memory to create specific sub-modules in its network of potential interactions.

## I. INTRODUCTION

Networks are a convenient way to describe systems that consist of many interacting parts such as protein-protein interaction (PPI) or social interaction networks. By abstracting its components as nodes and the nature of interaction (directionality, weightiness, etc.) between them as edges, a system can be studied structurally and dynamically within the paradigm of network science at the whole-system level. A network model may represent a purely Euclidian spatial system such as the network of streets in a urban grid. For a city, nodes will mean intersection and links to routes between them if they exist. In such a system all the spatial relationship among the parts are abstracted to the network. On the other

hand, many real non-Euclidian systems are still studied as networks by using the language of spatiality (shortest/average distance, diameter, etc.). We can think those systems too having spatiality, though non-Euclidian, in the sense that interactions of their parts are conditioned by their relative positions on the network, that is to say, it is a spatiality of network not the real system described by the network. Though there are many promising studies [1], [2], [3] and [4] for identifying the structure of networks (connectivity distribution, community structure, motif patterns, etc.), most do not consider the dynamical aspect of networks [5]. One particular failing of network studies is that often there is little differentiation between the links of potential interactions and those that are actually active in a specific time or space. Networks are generally considered as static structures in which the disparate temporal and spatial contexts are collapsed into one set of network interactions. Instead, this static structure could be thought as the underlying network of potential interactions on which multiple changing sub networks evolve. For example if we look at the spatio-temporal dynamics of protein interactions, we can turn all potential interactions as a network whose nodes stand for the proteins and edges for the putative interaction [6], [7] and [8]. But again the challenge is to find a way to describe the protein interaction patterns in the specific spatio-temporal order exhibited physiologically at a given time by relating node level stochasticity to the network level behavior. Till now much attention has been focused to identify the structure of real networks and it has been shown that many real networks are better described by scale-free and community structures (including hierarchical) than random networks [4], [5] and [9]. But random networks (Erdős-Renyi networks) are still important as statistical benchmarks for determining other networks' "unusual" properties. Motivated by the challenges to the dynamical analysis of networks given above and by the importance of these structures, we developed an algorithm to simulate the dynamics of movement on random, scale-free and hierarchical networks. For this, we defined an abstract entity moving metaphorically on the networks with respect to some stochastic rules and analyzed the paths it traces. We considered the paths as the information flow regarding sequential nodal interaction. For example, for a protein interaction network, the path can be said to describe the protein interactions in temporal order. Interaction units of many real networks have a kind of memory capacity. Neurons that fire recurrently strengthen their signals. Immune co-interactions strengthen on secondary infection and even intracellular (protein) signaling networks, are produced more readily on secondary application of stimulus or stress,

---

21/02/2014 This work was supported in part by The Scientific and Technological Research Council of Turkey (TUBITAK) and NIH award P01 AI106697-01 "Tissue compartmentalization of human lymphocytes".

(1) M.Y. is with Department of Bioengineering, Faculty of Engineering, Ege University, 35100, Izmir, Turkey (e-mail: [yucel.mesut@hotmail.com](mailto:yucel.mesut@hotmail.com))

(2) U.H. is with The School of Biomedical Engineering & Health Systems, Drexel University, Philadelphia, PA, 19104, USA (e-mail: [uri\\_hershberg@drexel.edu](mailto:uri_hershberg@drexel.edu)).

\* To whom correspondence should be addressed.

implying a memory capacity inherent to signaling processes. Thus, we also assigned a memory capacity  $s$  to the entity that enables it to remember its last  $s$  steps (with  $s$  being  $\ll$  network size) and increases the probability of these steps being taken in the same temporal order if encountered.

We designed the simulations so as to give results that enable us to compare different network dynamics in terms of network topology, average connectivity and memory. Different memory cases are compared for the same network architecture and connectivity. We see that the addition of memory has a great effect on both the number of appearance of paths during simulation and continuity of some learned paths (consecutive reproducibility of them). However, these effects are highly dependent on the connectivity of the network. If the connectivity is greater than the memory size it highly degrades the effects of memory (although not completely). Lowering the connectivity (filtering more strictly) has a great effect on all results for all types of networks. The low connectivity is much more efficient for the reproducibility of the paths especially in the memory cases. This makes sense when the connectivity is seen as a measure of the stochastic potential of the network, e. g., if the entity has many directional choices, then its remembering a specific path will not create much bias in the directional preferences.

## II. METHODS

### Network Parameters.

In this study, we simulated the dynamics of artificial networks with 1000 nodes connected with either a random, scale-free or hierarchical architecture [10], [11] and [12]. For each network type we simulated a low ( $k \sim 3.5$ ) and high ( $k \sim 15$ ) connectivity example. We studied the movement of single entity on these networks. The entities has 3 levels of memory and could remember either their last 10, 5 or 0 steps. Each condition was run twice for 150,000 entity steps and then 20 times for 40,000 steps.

### Creating Networks

*Random Network* - In random networks each node is connected to other nodes randomly with an average connectivity of  $k$ . We create networks with two levels of connectivity – low,  $k=3.5$  and high  $k=15$ . The degree distribution for a random network is normal-like. i.e. the network has a characteristic degree and there are no highly connected nodes (or hubs) [3]. To satisfy these conditions, we developed an iterative algorithm that at each step creates links between nodes such that no node will have connectivity lower than  $(1 - amp)k$  or higher than  $(1 + amp)k$ . With *amp* (or amplitude) being the maximal range between the highest and lowest connectivity in the network. We took  $amp = 0.6$  for our simulations, which meant that the connectivity of any node in the network would be between  $0.4k - 1.6k$ .

*Scale free Network* In a scale-free network, the probability of any selected node to have  $k$  connections is given by  $P(k) = zk^{-\gamma}$  where  $P(k)$  is the probability for the node to have  $k$  connections,  $z$  is the Riemann's zeta function (a normalizing constant),  $k$  is the connectivity and  $\gamma$  is the degree exponent which is a characteristic value for a given network.  $\gamma$  is generally between (1.5 – 3.5) for many biological and social networks [1], [3], [13] and [14]. The minimum and maximum

connections a node may have in our algorithm were assumed to be 2 and  $N - 1$ , respectively, where  $N$  is the total node number in the network. After entering the degree exponent ( $\gamma$ ) our algorithm uses the following mathematical methodology to calculate first the Riemann's zeta function and then the probability density of connectivity for a given degree exponent.

$$\begin{aligned} \sum_{k=min}^{max} P(k) &= P(k_{min}) + P(k_{min+1}) + \dots + P(k_{max}) \\ &= 1 \\ 1 &= z(k_{min}^{-\gamma} + k_{min+1}^{-\gamma} + \dots + k_{max}^{-\gamma}) \\ z &= 1/(k_{min}^{-\gamma} + k_{min+1}^{-\gamma} + \dots + k_{max}^{-\gamma}) \end{aligned} \quad (1)$$

For a given degree exponent ( $\gamma$ ) and calculated zeta function ( $z$ ), the probability density vector (pdv)  $(\frac{z}{k_{min}^{\gamma}}, \frac{z}{k_{min+1}^{\gamma}}, \dots, \frac{z}{k_{max}^{\gamma}})$  for the connectivities was obtained. Then the connectivity of each node was determined iteratively according to the following probabilistic equation [15] and [16]:

$$k_i = \alpha \quad (i = 1, \dots, N) \text{ where}$$

$$\sum_{j=1}^{\alpha-1} pdv_j < rpdv_0 < \sum_{j=1}^{\alpha} pdv_j \quad (2)$$

Where  $k_i$  is the connectivity of the  $i^{th}$  node of network,  $pdv_j$  is the  $j^{th}$  element of the pdv,  $pdv_0$  is the sum of all the elements of pdv and  $r$  is a random variable drawn from uniformly distributed unit interval. We continue to create the network we reach a connectivity distribution corresponding to the calculated probability density vector. To verify the network structure we ranked the nodes by their connectivity and obtained a decaying power law graph ( $\gamma=1.85$  for high connectivity graph and  $\gamma=3.65$  for low connectivity graph) [1] and [17].

*Hierarchical Network* – We considered hierarchical networks to be those that had ‘communities’. We defined a community as a group of nodes whose connections in the community are much denser than the connections to different communities [1]. Every network had one central community and two peripheral ones. Centrality of a community meant it was connected to all of the other communities in the network [10] and [11] while any two non-central communities were connected to each other only through the central one. We assumed that intra- and inter-community connections were random with the approximate average connectivity of  $k_1$  and  $k_2$ , respectively. To maintain hierarchical structure  $k_1$  was  $\gg k_2$ . We set their values at  $k_1=21$ ,  $k_2=6$  for high connectivity and  $k_1=4$ ,  $k_2=2$  for low connectivity. For a given community number,  $k_1$ ,  $k_2$  and *amp* values, our algorithm first allocates the nodes to the communities randomly and then connects the nodes of the same community as in the random networks. After this step, non-central communities are connected to the central one so as to give an average connection number of  $k_2$  between all three communities. Since the hierarchical network is, in a way, a random network, its degree distribution is again

normal-like with no hubs [11]. In all three network types no node has a connectivity of less than 2.

*Using a hybrid Gillespie algorithm to simulate movement.*

To simulate the movement of the entity in the network at certain probabilities determined by its memory and the network connectivity we used a stochastic simulation algorithm originally developed to simulate time evolution of a chemical reaction system [15] and [16]. We consider the directions in our system as corresponding to the reactions in Gillespie's original system. There is an important difference between the original algorithm and one modified and used here that whereas the number and types of possible reactions in a chemical system remain constant during the process, they vary from one step to another in our network system since at each step the set of potential directions changes. According to our modified Gillespie algorithm, at each step a probability distribution function for the directions must be constructed. After this, the next direction for the entity is chosen randomly from the constructed distribution function as follows: take  $\mu$  as an integer that satisfies the following equation [15] and [16]

$$\sum_{v=1}^{\mu-1} a_v < r a_0 \leq \sum_{v=1}^{\mu} a_v \quad (3)$$

where  $r$  is a random number drawn from uniformly distributed unit interval and  $a_0$  and  $a_v$  are the sum of all the elements of frequency (tendency) vector and  $v^{th}$  element of it, respectively. The number of  $\mu$  will be the direction (reaction) number from the overall frequency vector. All the nodes traced by the entity throughout the simulation were recorded in temporal order and analysed in terms of several connectivity measures as explained in the results section.

*Rules for movement and memory in the network*

At each simulation step, the entity changes its position to a neighbor node (a node linked to the original node). To prevent the entity from being trapped in a very short circuit, the entity will never go back to its previous node. Without memory the entity is likely to go in any direction, as all the potential directions will have an equal probability of being taken. With a memory of size  $s$ , the entity remembers the last  $s$  nodes in temporal order and if it steps back into a part of the path that is in memory it tends to retrace this path in the same order. Therefore, the memory creates a bias in the directional preference of the entity. To quantify the effect of memory we defined a parameter  $alpha$  that increases the probability of a direction that is already memorized. In our simulation,  $alpha$  was taken as  $1/1000$  meaning that a memorized direction will be 1000 times more probable than an un-memorized one. In our study, we simulated both  $s = 0$  (no memory),  $s = 5$  and  $s = 10$  cases.

For a given network type and connectivity, we used the same network for simulating different memory cases. At each simulation, we have a network and an entity moving stochastically on that network.

### III. RESULTS

We analyzed the structure of the paths visited by the entity and attempted to identify to what extent the overall structure of the network on which they moved, and the extent of their memory influenced these structures.

*Memory causes a greater regularity of path structure*

To characterize the regularity of paths at different path lengths we characterized each network by all the possible sub-paths of length or window size (in our case *window size = 5 or 10*). We then counted how often each such pattern occurred. In Table I and II, the results for average connectivity  $k = 3.5$  and  $k = 15$  are given, respectively, subdivided by network topology and memory. Here, the max value gives the maximum recurring number of a specific sub-path throughout the simulation whereas the mean value gives the mean recurring number of each sub-path type. Maximum consecutive retracing number gives the max retracing number of a sequence of size 5 traced *consecutively*. This number measures the capability of the memory within a network to enable the entity to keep its learned path without fail. Movement in even the less connected network is highly stochastic and even in the low connected networks only a relatively low number of paths are retraced (Max  $\sim 2800$  out

TABLE I  
SIMULATION RESULTS FOR  $k = 3.5$

Win size	Random		No memory Scale-free		Hierarchical	
	max	mean	max	mean	max	mean
5	13	2.353	22	2.1346	18	2.0339
10	3	1.0099	3	1.0096	2	1.0079
Max cons. retracing	1		2		3	
Win size	Random		Memory size = 5 Scale-free		Hierarchical	
	max	mean	max	mean	max	mean
5	2785	3.6439	2302	2.8642	750	21.719
10	2776	2.2105	2291	1.7457	748	17.104
Max cons. retracing	248		452		593	
Win size	Random		Memory size = 10 Scale-free		Hierarchical	
	max	mean	max	mean	max	mean
5	1010	3.6785	492	4.1711	824	32.839
10	1003	2.6405	491	2.8876	821	23.132
Max cons. retracing	168		162		233	

TABLE II  
SIMULATION RESULTS FOR  $k = 15$

Win size	Random		No memory Scale-free		Hierarchical	
	max	mean	max	mean	max	mean
5	2	1.0015	2	1.0009	2	1.0015
10	1	1	1	1	1	1
Max cons. retracing	1		1		1	
Win size	Random		Memory size = 5 Scale-free		Hierarchical	
	max	mean	max	mean	max	mean
5	156	1.1847	41	1.1338	95	3.6975
10	151	1.1691	40	1.0929	93	3.0677
Max cons. retracing	59		39		107	
Win size	Random		Memory size = 10 Scale-free		Hierarchical	
	max	mean	max	mean	max	mean
5	152	1.5045	40	1.319	137	6.8795
10	149	1.454	39	1.1882	135	4.8548
Max cons. retracing	39		21		47	

of 150,000). However, it is clear that even so memory has a significant effect on reproducibility of paths as the mean retraced path number grows by up to 2 to 17 fold (Table I window 10) and the Max. retraced path number by between ~ 80 and ~ 250 times. Interestingly, while the longer memory of 10 helps to raise the mean path number, the longest Max path number is at memory 5 in all network types.

Raising connectivity raises the stochasticity of traveling in the networks and degrades the reproducibility of movement both with and without memory, dropping 5 to 60 fold in numbers. Despite this it is still quite clear in Table II that entities with memory will show more recurrence by 40-150 orders of magnitude than those without (Table II).

*Memory and structure of the complete network effect the relative connectivity of the visited network*

To understand the interactive effects of the memory and topology on the network dynamics, we derived a plot that shows the relations of the most visited nodes and positions of them in the network. For this, we constructed the sets of the 5, 10,..., 100 most visited nodes (in total, we had 20 sets of nodes). Then, we calculated the average connectivity and relative connectedness of each set and plotted these values with respect to most visited node sets. The relative connectedness (in %) of a set is a measure of how coherent and interconnected a set of nodes is. We defined it by the following equation:

$$Rel_{con}(in\%) = \frac{l_{obs}}{n \times \frac{n-1}{2}} \times 100 \quad (4)$$

where  $l_{obs}$  is the number of observed direct links between the nodes in the set and  $n$  is the number of nodes in that set. To make the calculation clear, see Figure 1.

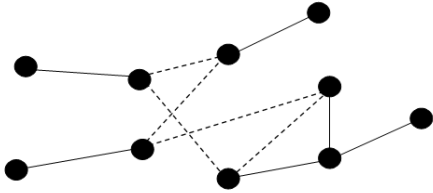


Fig. 1. For the given network of  $N = 10$ , the relative connectedness of the chosen set of 5 nodes (those connected to one another by dashed links) is  $5/(5 \times 4/2) \times 100 = 50\%$ .  $N$  is the total number of nodes in the network.

In the figures (Fig. 2, 3 and 4), we see the similar patterns for all the networks. When there is no memory, the entity prefers to wander within the high average connectivity groups (notice the average connectivity as a monotonically decreasing function of mostly visited nodes). In no memory cases there is always a good correlation between the mostly visited nodes' average connectivity and relative connectedness. But the memory disturbs this correlation possibly by driving the entity into a coherent group of nodes (compare blue with red and black with green curves in Fig. 2, 3 and 4). The memory works best in a group of nodes that is relatively well

connected but does not have a high average connectivity (blue to red curves each case).

By limiting the movement potential of the entity, the average connectivity of the network acts as a filter for the noise (randomness). It is very clear from the tables that lowering the connectivity (filtering more strictly) has a substantial effect on all the results for all the networks (compare the results for no memory cases of the networks with different connectivity). The low connectivity is much more efficient for the reproducibility of the paths even for no memory cases. However, with memory the effect is even more dramatic and we see that most highly visited nodes are highly coherent (i.e. have high relative connectedness).

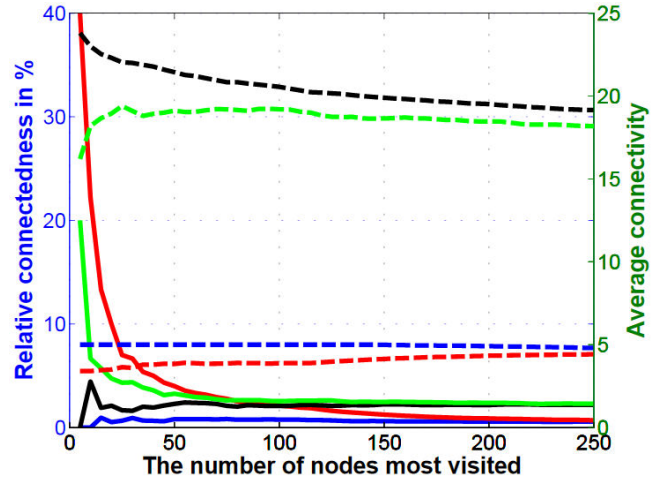


Fig. 2. Relative connectedness in % (full lines) and average connectivity (brocken line) of nodes ranked by times visited over entire simulation in random networks. Notice that the memory always decreases the average connectivity of the most visited sets and increases their relative connectedness. Blues:  $k=3.5, s=0$ , Reds:  $k=3.5, s=5$ , Blacks:  $k=15, s=0$  and Greens:  $k=15, s=5$ .

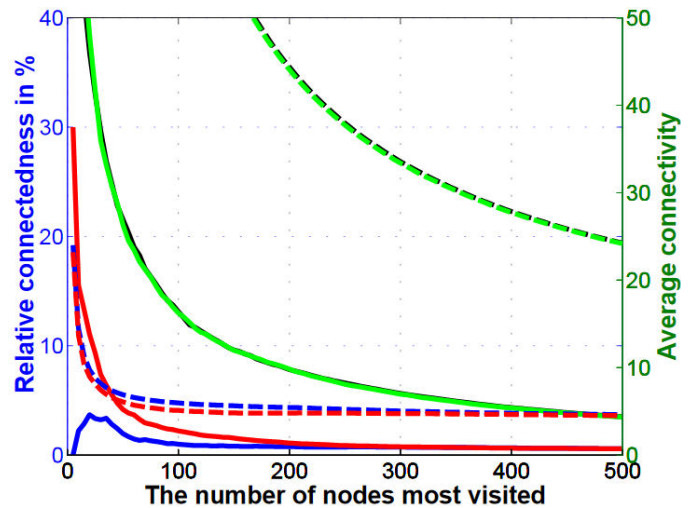


Fig. 3. Relative connectedness in % (full lines) and average connectivity (brocken line) of nodes ranked by times visited over entire simulation in scale free networks. Notice that the effects of memory are lost at high connectivity. Blues:  $k=3.5, s=0$ , Reds:  $k=3.5, s=5$ , Blacks:  $k=15, s=0$  and Greens:  $k=15, s=5$ .



Moreover, this high relative connectedness is not directly connected to average connectivity as it is in the no memory case. In high connectivity networks the memory cannot effect the results as much as it does in low connectivity networks with and without memory cases (compare black with green curves in Fig. 2, 3 and 4). This is especially true for scale-free networks possibly due to the fact that the memory cannot keep the entity in sub-networks when there are very huge hubs (the size of the biggest hubs in low and high connectivity scale-free networks are  $\sim 100$  and  $\sim 900$ , respectively). From Fig. 3, we can additionally say that in high connectivity networks the average connectivity of the sub-regions is the main determinant for the entity's tendency to re-visit a site during the simulation, namely, the entity is attracted to the regions with high average connectivity and memory does not have much influence.

As stated above, memory is more efficient in a group of nodes that is relatively well connected but has a low average connectivity (see Fig. 2, 3 and 4). In high connectivity networks it is less probable to have this kind of groups, especially given the small network size of our simulation. If the network average connectivity is greater than max group average connectivity ( $k > 2l_{max}/n$ ) then adding a link to the network will less probably increase relative connectedness though it certainly does the average connectivity. To prove this numerically, we simulated a network evolution model and calculated a normalized ratio of the relative connectedness of randomly chosen sets of 5 nodes to the average connectivity of the entire network. In this way we show how the relative connectedness changes with respect to the average connectivity. We ran the simulations 1000 times and used the mean values of the ratios for the plot. We also added  $\mp 2$  standard error curves to the plot. Fig. 5 shows that the ratio quickly reaches a steady-state value of 1 especially after the condition  $k > 2l_{max}/n = 4$  is satisfied. To us, this is the reason why in high connectivity network it is very hard to find a group of nodes with high relative connectedness and low average connectivity in which memory would drive the entity to stable patterns.

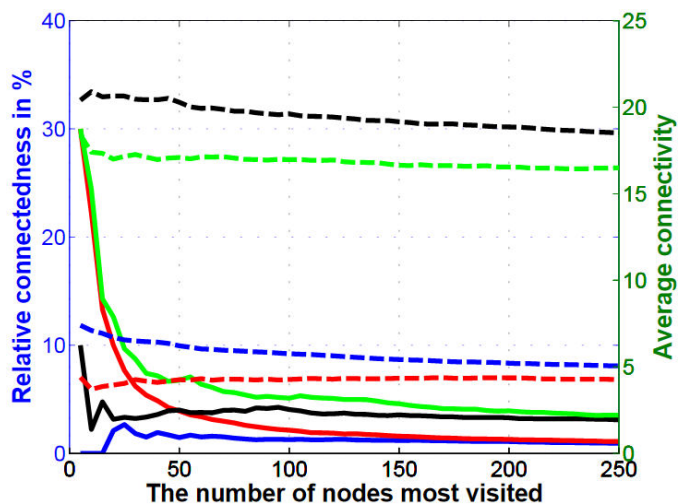


Fig. 4. Relative connectedness in % (full lines) and average connectivity (brocken line) of nodes ranked by times visited over entire simulation in hierarchical networks. Blues:  $k=3.5, s=0$ , Reds:  $k=3.5, s=5$ , Blacks:  $k=15, s=0$  and Greens:  $k=15, s=5$ .

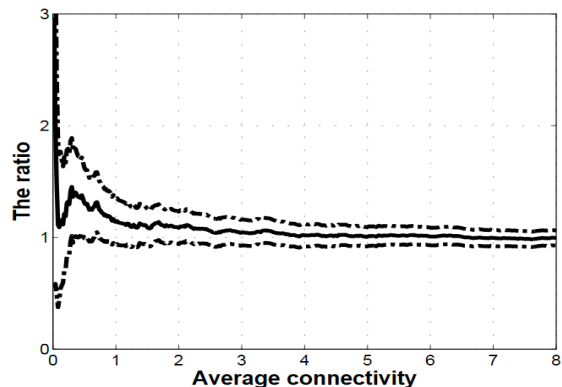


Fig. 5. The ratio of relative connectedness of randomly chosen 5 node-group to its average connectivity as a function of its connectivity (the dashed curves represent  $\pm 2$  standard errors). Notice that the band between the error bars quickly reaches a normalized steady-state value of 1 especially till the average connectivity reaches the max connectivity of the group that is 4.

We also analyzed the visiting frequencies of the most visited nodes (for how long the most visited nodes are visited) to understand the corresponding biases in the temporality. To do this, we plotted the total fraction of nodes visited (as % of simulation time) with respect to the number of most visited nodes. When there is no memory, the bias in the temporality can completely be attributed to the network topology. All the time, we saw the most and least biased temporality in the scale-free and random networks. And interestingly, with increased connectivity the bias in the random network was lowered whereas it was dramatically raised in the scale-free network. In each case, the memory discernibly increased the bias in the temporality except in the high connectivity case of scale-free network implying that the larger hubs prevent the formation of sub-networks in which the memory would drive the entity. What was peculiar to the hierarchical networks was that only in those networks under memory cases there were some nodes left unvisited at all (see Fig. 6). We can interpret this as stemming from the community structure of the hierarchical networks (as sub-networks, communities can keep the entity longer by being relatively less connected to the other parts of the network).

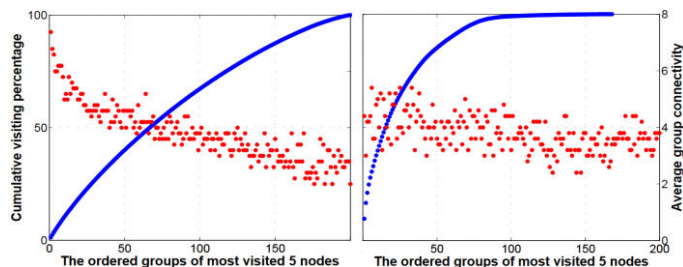


Fig. 6. Cumulative visiting percentage (blue) and average group connectivity (red) with respect to the ordered groups of most visited 5-nodes for hierarchical networks with  $k=3.5, s=0$  (left) and  $k=3.5, s=5$  (right). Notice that there is already a bias, though very slight, in the visiting numbers without memory (left plot). This bias comes from the spatiality of the networks implying that the spatiality is an independent factor effecting the dynamics. Memory creates a huge bias as seen in the rightside plot. Blue scatters reach the steady state at around 170. 5-node group meaning that there are almost 30 groups (150 nodes) left unvisited. We analyzed these nodes and saw that only 1 of them belonged to the central community. Two communities were never visited at all, whereas 76% of another community was left unvisited.

#### IV. DISCUSSION AND CONCLUSIONS

Many real networks (biological, social, etc.) are sparse [18]. For instance the average connectivity of the protein interaction network of *S. cerevisiae* is 2.12-2.9 whereas the total protein number of the network is 2115 [19]. In light of this observation and our confirming simulation results we may conclude that there should be a *connectivity constraint* for a network system to be able limit the agent level stochasticity. In such an instance it appears that even a very rudimentary memory, on its own without goals or more specific context, can give rise to a network dynamics that is less stochastic than the network structure alone would predict. Real networks seem to have evolved so as to have low connectivity to act as spatial filter on the highly stochastic input data. As a result, we would suggest that memory and spatial organization act on the network dynamics in their own manners and cannot be reduced to each other.

Also, there is seemingly always some sort of memory inherent to self-organizing systems including PPI and road networks. Though the memory aspect of road networks is self-explaining (if a city dweller knows a road better than we can expect him/her tending to prefer that road), it is more ambiguous and therefore interpretative to show the memory in the PPI interaction networks. As discussed above, the memory drives the entity into some sub-networks. Though the sub-networks are the parts of the network structure (they are not dynamically formed but they are already there before entity dynamics), it is the memory that enables the entity to interact the network structure giving rise for the sub-networks to have some emergent functionality in the network dynamics. As an example, the functional sub-networks emerged from the random network dynamics are given in Fig. 7.

In essence it appears that memory causes the entity to trace and thus identify specific communities. While community identification is by no means new [10], we build here a first version which is based on an internal ‘bottom up’ sweep of the network. More generally we believe that our modeling paradigm presented here could provide one of the early steps for dynamical approaches to describe many real networks and how individual entities or information flow along them.

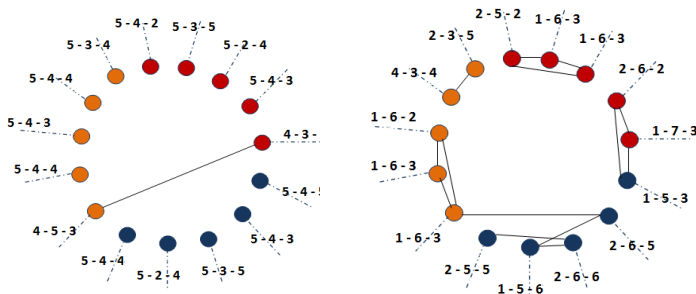


Fig. 7. The sub-networks of the most visited 15 nodes with and without memory cases for random networks for with low connectivity ( $k=3.5$ , left:  $s=0$ , right:  $s=5$ ). The first, second and third numerals show the number of other connections the node has to other nodes (outside the top 15), the shortest distance back to top 15 and to the least visited 15 nodes. The red, blue and orange circles stand for the first, second and third most visited 5-node groups. Notice how memory ‘leads’ the entity to the islands of nodes with high intra-connections and low outer connections, i.e. to specific communities.

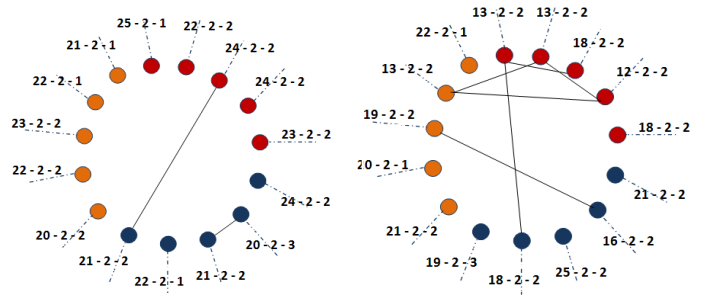


Fig. 8. The sub-networks of the most visited 15 nodes with and without memory cases for random networks with high connectivity ( $k=15$ , left:  $s=0$ , right:  $s=5$ ). For numeral and color codings see Fig. 8. Though the memory is still effective on the identification of some specific communities (sub-networks), these sub-networks however are less tightly connected than the ones found in low connectivity networks (compare *in-community* link numbers of  $k=15$  and  $k=3.5$  cases, that is, 6 and 14).

#### REFERENCES

- [1] A.-L. Barabasi and R. Albert, "Emergence of scaling in random networks." *Science*, 286:509-512, 1999
- [2] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii and U. Alon, "Network Motifs: Simple Building Blocks of Complex Networks", *Science*, vol. 298, 2002.
- [3] O. Hein, M. Schwind and W. König, "Scale-Free Network: The Impact of Fat Tailed Degree Distribution on Diffusion and Communication Processes", *Wirtschaftsinformatik*, vol. 48, pp. 267-275, 2006.
- [4] E. Alm and A. P. Arkin, "Biological Networks", *Current Opinion in Structural Biology*, vol. 13, pp. 193-202, 2003.
- [5] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez and D.-U. Hwang, "Complex networks: Structure and dynamics", *Physics Reports*, vol. 424, pp. 175-308, 2006.
- [6] H. Kitano, "Systems biology: A brief overview", *Science*, vol. 295, pp. 1162-1164, 2002.
- [7] A. Levchenko, "Dynamical and integrative cell signalling: Challenges for the new biology", *Biotechnol. Bioeng.*, vol. 84, pp. 773-782, 2003.
- [8] R. J. Prill, P. A. Iglesias, A. Levchenko, "Dynamic properties of network motifs contribute to biological network organization", *PLoS Biol.*, vol. 3, pp. 343, 2005.
- [9] A.-L. Barabasi and R. Albert, "Emergence of scaling in random networks", *Science*, vol. 286, pp. 509-512, 1999.
- [10] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks", *PNAS*, vol. 99, pp. 7821-7826, 2002.
- [11] A.L. Barabasi and Z.N. Oltvai, "Network Biology: Understanding The Cell's Functional Organization", *Nature Reviews, Genetics*, vol. 5, pp. 101, 2004.
- [12] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and A.-L. Barabasi, "The large-scale organization of metabolic networks," *Nature*, vol. 407, pp. 651, 2000.
- [13] R. Albert, "Scale-free networks in cell biology", *J. Cell Sci.* vol. 118, pp. 4947-4957, 2005.
- [14] S. Wuchty, "Scale-free behavior in protein domain networks", *Mol. Biol. Evol.*, vol. 18(9), pp. 1694-1702, 2001.
- [15] D. T. Gillespie, "Approximate accelerated stochastic simulation of chemically reacting systems", *J. Chem. Phys.*, vol. 115, pp. 1716, 2001.
- [16] D. T. Gillespie, "Exact stochastic simulation of coupled chemical reactions", *The Journal of Physical Chemistry*, vol. 81, No. 25, 1977.
- [17] G. D'Angelo and S. Ferretti, "Simulation of Scale-Free Networks", *Digital Object Identifier*, vol. 10.4108, 2009.
- [18] A.-L. Barabasi, "Network Science", chapter 3, pdf version, 2012.
- [19] H. Jeong, S. P. Mason, A.-L. Barabasi and Z. N. Oltvai, "Lethality and centrality in protein networks", *Nature*, vol. 411, 2001.
- [20] J. J. Kupiec, "On the lack of specificity of proteins and its consequences for a theory of biological organization", *Progress in Biophysics and Molecular Biology* 102, 45-52, 2010.