A Cognitive-inspired Model for Self-organizing Networks

Daniel Borkmann*, Andrea Guazzini†‡, Emanuele Massaro§, Stefan Rudolph‖
*Communication Systems Group, ETH Zurich, Switzerland
† Institute for Informatics and Telematics, National Research Council, Pisa, Italy
‡ Department of Psychology, University of Florence, Italy
§Department of Informatics and Systems, University of Florence, Italy
‖ Organic Computing Group, University of Augsburg, Germany

***Abstract*—In this work we propose a computational scheme inspired by the workings of human cognition. We embed some fundamental aspects of the human cognitive system into this scheme in order to obtain a minimization of computational resources and the evolution of a dynamic knowledge network over time, and apply it to computer networks. Such algorithm is capable of generating suitable strategies to explore huge graphs like the Internet that are too large and too dynamic to be ever perfectly known. The developed algorithm equips each node with a local information about possible hubs which are present in its environment. Such information can be used by a node to change its connections whenever its fitness is not satisfying some given requirements. Eventually, we compare our algorithm with a randomized approach within an ecological scenario for the ICT domain, where a network of nodes carries a certain set of objects, and each node retrieves a subset at a certain time, constrained with limited resources in terms of energy and bandwidth. We show that a cognitive-inspired approach improves the overall networks topology better than a randomized algorithm.**

***Keywords*-complex networks; cognitive modelling; self-awareness systems;**

I. Introduction

Among the capabilities of the human cognitive system that is attracting most computer scientists, there is the ability of humans to develop local algorithms able to exploit what might be called "collective human computation". As collective human computation, we refer to the natural synchronization between the cognitive elaborations made by a person which is immersed into group dynamics. In such a condition, human beings analyze only some relevant information coming from the group, giving to the group only some relevant contributions for the general problem which is faced. In this way, the group can be described as more than just the sum of its single components [9].

When being faced with insufficient data or insufficient time for rational processing, humans have developed strategies that allow to take decisions in these situations. In general, such an effect has been well described in the cognitive heuristics program proposed by Goldstein and Gigerenzer, which suggest starting from fundamental psychological mechanisms in order to design models of heuristics [8].

Only some relevant information are extracted from the environment while the rest is interpolated for building our knowledge. Among others, Milgram et al. have shown experimentally how humans are able to adopt effective strategies to solve very complex problems, exploiting optimally their partial knowledge of their environment [4, 13].

This kind of human distributed computing has been studied deeply only from the perspective of disciplines such as social cognition and social psychology, while it is not yet well known in other domains.

The social cognition domain studies human cognition as characterized by the use of "fast and frugal" solutions, that are specialized for a social context in which we live using a bounded rationality and limited computational resources [17]. Therefore, the aim of our work is to assemble a working computational scheme inspired by the operating principles of human cognition, based on general assumptions about cognitive high-level functions.

This approach promises to embed some fundamental aspects of the human cognitive system in a computational model in order to obtain a minimization of computational resources needed for the task and the evolution of a dynamic knowledge network capable of generating strategies suitable for networks like the Internet, which are too large and too dynamic to ever be fully/perfectly known [12].

The fundamental aspects on which we focused our modeling, involves the spread of information through a human network, and the knowledge representation arising from the dynamics of short-term memory (STM) and long-term memory (LTM). The passage of information between STM and LTM occurs through a simple cognitive heuristic approach, which compatibly with their computational capacity reduces the dimensionality of the information required to represent the environment in a dynamic manner.

In our previous work [12], we applied such an approach to the community detection problem, which can be considered as a task of great importance in many disciplines [1, 3, 16, 18, 20], where systems can be represented as graphs. The first version of the algorithm was characterized by a two step procedure (e.g. discovering and elaboration phases), in which the effect of the nodes' connectivity on the information spreading was exploited by nodes to assess a first approximation of the topology of the network. In this work, we present a second version of the algorithm in which the third phase was added in order to refine the topology detection by a cognitive inspired strategy which embeds the cognitive dissonance theory [5].

In general, as the Internet nowadays, human social networks have to be considered as a continuum of nested communities whose boundaries are somewhat arbitrary [10].

Here, we propose such a tool for detecting communities in complex networks using a local algorithm, applied as a cellular automaton. In this approximation, a node is just modeled as a memory and a set of links to other nodes. The information about neighbouring nodes is propagated using a standard diffusion process, and elaborated locally using a non-linear competition process among the information. This process can be considered an implementation of the "take the best" heuristic [7], which relies on the assumption that the most relevant or easily detectable information gives an accurate estimate of the frequency of the related event/contents in the population. The result of the algorithm equips each node with information about possible hubs or super-nodes present in its environment, and such information can be used by the node to rewire its connections whenever its fitness does not satisfy some given requirements.

In real-world applications, such a process can be engineered within the ICT domain. Consider for instance resilience and scalability effects in service ecosystems. There, one important factor is to decentralize services. This can be done with the help of creating overlay networks on top of large-scale ones such as the Internet. An adaptive, intelligent or even resource-optimizing algorithm plays a crucial role for the (self-)maintenance of such systems.

In that way, we could tackle the first steps to create an intelligent, semi-structured peer-to-peer overlay network from an unstructured one, e.g. like a *self-optimizing* FastTrack [11] network. FastTrack itself uses a semi-structured overlay network with a mix of *designated* super-nodes and normal nodes. The latter have to connect to one of the super-nodes in order to minimize redundant communication overhead. There, participating nodes could retrieve content at a certain time with given resource constraints (e.g. bandwidth, energy, latency), detect super-nodes automatically during an operation, and thus change their connections (and therefore the overlay topology) for better conditions.

The rest of this paper is structured as follows: section 2 describes the scenario and section 3 the cognitive-inspired algorithm. In section 4, we evaluate our cognitive-inspired algorithm with a randomized algorithm. Eventually, in section 5 we conclude our findings.

II. Scenario

As a first step towards such a real-world self-optimizing peer-to-peer network, we consider the following *simplified* approach: given N individuals (nodes), labeled from 1 to N, where each individual hosts exactly one item. There are I items distributed over individuals. Also, we label items from 1 to I, where $I \leq N$, thus two or more individuals can host same items. Each individual has a pre-defined maximum number of links, where it can connect to other nodes. As a simplification, we can denote a link between two individuals as "wired". During the initial state, not all links are wired. Hence, some individuals still have free capacities in our network topology. For each node from 1 to N, the free link capacities are uniformly distributed within a given interval $[a, b]$. Now, each individual acts greedy and wants to collect up to a maximum number of unique items $I_{curr} \leq I_{max}$ from other individuals, where $I_{max} \leq I$ and I_{curr} defines the actual number of items that have been collected. However, in collecting, an individual is constrained by a given budget/energy $C_{curr} = C_{max}$ it can spend. While exploring its ith-degree neighbors ($i > 0$), it has to pay for the number of hops if it has enough budget left, so that $C_{curr} \leftarrow C_{curr} - i$. Note that an individual does not have a global knowledge of the topology. After this process has been completed by each node, a given fitness function f can be calculated. f is then used in order to find "weak nodes". Candidates must give up one of their links and create a new one to a more "promising" node, e.g. to a super-node/hub. Hence, in each round of this process, the topology will be partially changed by a set of weak nodes and a minor randomly selected component of the system.

In this paper, we are evaluating our network's behaviour from two perspectives: (i) f_1 : maximizing a node's I_{curr}, that is, each node shall collect as many unique items as possible, so that $I_{max} - I_{curr} \to 0$ while complying to its energy constraint, (ii) f_2 : minimizing a node's $C_{max} - C_{curr} \to 0$ while having $I_{curr} = I_{max}$ items collected. In this case, the pre-defined energy C_{max} is sufficiently large (no energy constraint) to collect I_{max} items, so that the system's focus is to minimize its overall energy. In each round in (i) and (ii), I_{curr} and C_{curr} are reset and f reevaluated. We claim that by carefully choosing weak nodes and promising nodes for rewiring links, we can optimize f over time. Hence, instead of just randomly selecting individuals, we give each of them a bounded memory that provides knowledge about its surrounding for a better decision making as provided in the next section. Thus, we make nodes self-aware of their own "world".

III. Algorithm

We create a local algorithm where an individual is simply modeled as a memory and a set of connections to other individuals. The "learning" (nonlinear) phase is modeled after competitions found in the chemical/ecological world, where resources compete against each other in order to not fall into oblivion.

We consider an unweighted undirected network with the adjacency matrix A: the adjacency matrix of a finite graph G on n vertices is the $n \times n$ matrix where the non-diagonal entry A_{ij} represents the presence ($A_{ij} = 1$) or the absence ($A_{ij} = 0$) of a link between the vertices i and j.

Then each vertex i in the graph is characterized by a state vector S_i that represents its knowledge of the others. In our model, we consider S as a probability distribution, in particular $S_i^{(k)}$ is the probability that individual i belongs to the community k.

Then $S_i^{(k)}$ is normalized using the index k. Considering $S = S(t)$ the state matrix of the network at time t, with $S_{ik} = S_i^{(k)}$. At time $t = 0$ each node only knows about itself so $S_{ij}(0) = 1$ if $i = j$ and 0 otherwise. As mentioned before, the competition phase is modeled analogous to a chemical/ecological concept. Our algorithm is inspired by the concept of *diffusion and competitive interaction* in network structure introduced by Nicosia et al. [15].

If two populations x and y are in competition for a given resource, their total abundance is limited [14]. After normalization, we can assume that $x + y = 1$, i.e., where x and y are the frequency of the two species, and $y = 1 - x$. The reproductive step is given by $x' = f(x)$, which we assume to be represented by a power $x' = x^\alpha$. For instance, $\alpha = 2$ models the birth of individuals of a new generation after binary encounters of individuals belonging to the old generation, with non-overlapping generations (eggs laying) [2].

After normalization we obtain:

$$x' = \frac{x^\alpha}{x^\alpha + y^\alpha} = \frac{x^\alpha}{x^\alpha + (1-x)^\alpha}. \tag{1}$$

Introducing $z = (1/x) - 1$ ($0 \leq z < \infty$), we get the map

$$z(t+1) = z^\alpha(t), \tag{2}$$

whose fixed points (for $\alpha > 1$) are 0 and ∞ (stable attractors) and 1 (unstable), which separates the basins of the two attractors. Thus, the initial value of x, x_0, determines the asymptotic value, for $0 \leq x < 1/2$, $x(t \to \infty) = 0$, and for $1/2 < x < 1$, $x(t \to \infty) = 1$.

The dynamics of the network are given by an alternation of communication and elaboration phases. In the communication phase, there is a diffusion of information in which each node has a memory factor m; in this way, in each time step nodes update the previous information with new information. Due to this parameter, we can introduce some limitations into the algorithm as in the human cognitive system such as the mechanism of oblivion and the timing effects: the most recent information has more relevance than previous information [6, 19].

We assume that nodes talk with each other and we suppose that nodes with high connectivity degree have greater influence in the process of information's diffusion. This is due to the fact that during a conversation it is more likely to know a vertex with high degree instead of one that has few links. For this reason, the information dynamics is a function of the adjacency matrix A.

Then, in the communication phase, the state of the system evolves as

$$S\left(t + \frac{1}{2}\right) = mS(t) + (1-m)AS(t). \tag{3}$$

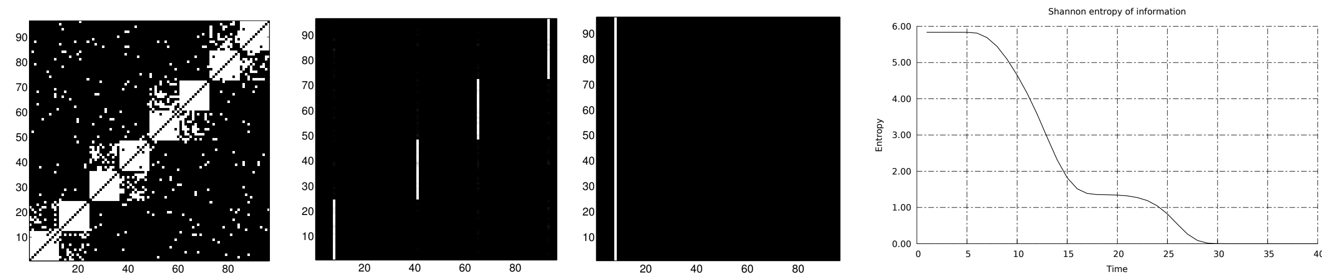
The competition phase is modeled analogously to a competitive interaction between the nodes in the network [15]. In this way the dynamic of the model is given by the sequence $S(t) \to S(t + \frac{1}{2}) \to S(t+1)$:

$$\begin{aligned} S_{ik}\left(t + \frac{1}{2}\right) &= mS_{ik}(t) + (1-m)\sum_j A_{ij}S_{jk}(t), \\ S_{ik}(t+1) &= \frac{S_{ik}^\alpha(t + \frac{1}{2})}{\sum_j S_{ij}^\alpha(t + \frac{1}{2})}. \end{aligned} \tag{4}$$

The node's memory is assumed to be large enough to contain all information about other nodes, and the model is characterized by two free parameters: the memory m and the coefficient α. As Figure 1 shows, this model is correlated to the values of parameters, and it is able to discover different final structures and results. In Figure 1 (a), an example of a hierarchical network in form of an adjacency matrix A is represented, where a three-levelled matrix is composed by 4 blocks of 2 sub-communities of 8 nodes each, with a link probability that is respectively of 0.98 inside sub-community, 0.3 in the first level of nested blocks, and 0.03 among blocks. The white points indicate the presence of a link between the node i and the node j, $A_{ij} = 1$. In Figure 1 (b), the asymptotic configuration of the matrix S using $m = 0.7$ and $\alpha = 1.4$ is shown, while in Figure 1 (c) with $m = 0.27$ and $\alpha = 1.25$. Finally, in Figure 1 (d), the dynamic evolution of the entropy of information, E corresponding to the case (c), defined as $E^{(S)} = -\sum_i P_i^{(S)} \log(P_i^{(S)})$, is represented, where $P_i^{(S)} = \sum_i S_{ij}$. The entropy E reaches the maximum only for the flat distribution, where each node knows only itself, and reaches a minimum (zero) when all nodes know the same label (i.e. all state vectors are the same and contain just one element different from zero). If the population is evenly distributed among n clusters, the entropy is $E = \log(n)$. The value of Entropy $E(t)$ allows us to discover the structure of the network, where different levels of the hierarchical structure are identified by plateaus as shown in Figure 1 (d).

Now the question is: is it possible to design the algorithm independently from the parameters?

In order to solve this task, we explore a "cognitive algorithm". We first define the concept of cognitive dissonance between adjacent agents. Cognitive dissonance has been defined within the field of social psychology from Leon Festinger [5], in order to explain the natural tendency of people to reduce conflicting cognitions creating a consistent belief system, or alternatively by reducing the importance of any source of dissonant elements (e.g. sometimes friends or neighbors). The theory shows a good predictive power,



(a) (b) (c) (d)

Figure 1. (a) Hierarchical three-level network with 4 principal communities. (b) Final configuration of state matrix S with $m = 0.7$ and $\alpha = 1.4$. (c) Final configuration of state matrix S with $m = 0.27$ and $\alpha = 1.25$: the final mono-cluster is identified by the major hub in the network. (d) Entropy of information for the whole network during time regarding the case (c).

shedding light on otherwise apparently irrational or destructive behavior, and can be reduced in our work as described in the following equation:

$$D_{ij} = \frac{|S_i - S_j|}{2}, \tag{5}$$

that is the difference between the absolute values of the state vectors between i and j. Then, we define the local entropy for each node at time t, considering the state matrix S:

$$E_i^t = -\sum S_i^t \log S_i^t. \tag{6}$$

In Figure 1 (d), we show the global entropy of information of the network during the time. The three plateaus correspond to three different levels: if we evaluate the first derivative of the entropy we can identify three peaks, while in the second derivative, we observe three changes of sign. For this reason, we evaluate the first and the second derivative of the local entropy for each node. Analogously for the entropy defined above, it is possible to introduce the concept of local entropy for each node in order to study the local view of agents. Similar as we can observe in Figure 1 (d), it is possible to detect different plateaus corresponding to the different network sub-clusters that the single node discovers during time. We observed that we can use a fixed value of the parameter m, while we have to change the value of α in order to find the community and in particular the hubs that labels each community. For this reason, we simulate an exploration phase of the network several times in which the nodes save their state vector S_i^t, in a *temporary memory box*, when they observe a change in sign of the second derivative. If the following condition is satisfied

$$sign\left(\frac{\delta^2 E_i^{t-1}}{\delta t^2}\right) \neq sign\left(\frac{\delta^2 E_i^t}{\delta t^2}\right), \tag{7}$$

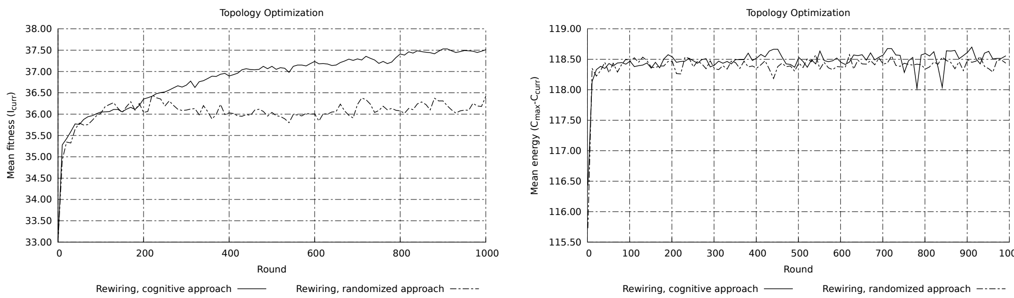
the state vector S_i^t is stored into the temporary long term memory together with the value of the first derivative of the local entropy and the entropy. When a node meets an impasse (e.g. its state vector entropy and its cognitive dissonance do not evolve anymore) its α is changed by the following mechanism, if

$$\left|\frac{E_i^{t-1} + D_i^{t-1}}{K_i}\right| - \left|\frac{E_i^t + D_i^t}{K_i}\right| < \epsilon, \tag{8}$$

where K_i is the connectivity degree of the node i. Then, a counter τ is increased by 1 ($\tau_i \leftarrow \tau_i + 1$), and if τ_i becomes greater than a given threshold (say τ^*), the parameter α is updated in the following way:

$$\alpha_i = 1.5|\eta\sigma| + 1, \tag{9}$$

where η is a random Gaussian variable with mean 1 and standard deviation σ. After a typical period of a fixed length (ΔT), the process is stopped for all nodes and a node's long term memory is updated with a new sample respectively experience. The long term memory is characterized by a bound threshold B^1 (here $B^1 = 5$) in order to mimic the ecological limits of such cognitive functions (i.e. bounded rationality). After the node has saved its state vector when the sign of its second entropy derivative changed (eq. 7), it proceeds in structuring its long term memory. First, its first derivatives are decreasingly sorted, and then the first B^1 time positions are recorded. Later, such B^1 element vectors are descendingly sorted with respect to the entropy. Finally, using the time positions, the correspondent state vectors is analyzed and larger elements for each state vector are assumed as potential hubs and therefore stored into the long term memory. At this stage, the long term memory of each node is composed by a list of B^1 sets of potential hubs, ordered following the procedure from the more local to the more global one. Moreover, the long term memory is bounded by another threshold (B^2), which represents the long term memory buffer, i.e. the maximum number of the B^1 sets it can consider/contain, so that the long term memory is represented by a (B^1, B^2) matrix. Finally, each node summarizes its knowledge of the network building a *hub list* obtained by analyzing the frequency in which each hub appears within the long term memory, which is subsequently ordered from the most represented (i.e. the



(a) (b)

Figure 2. (a) Fitness function that shows the mean number of retrieved items per node. The fitness of the randomized algorithm is represented as the dashed line, our approach as the solid line. (b) Mean energy usage per node of the randomized algorithm (dashed) and our approach (solid).

hub with a larger frequency) to the least represented one. The knowledge of the network (i.e. the hub list) is used by weak nodes in order to increase the fitness.

The nodes' fitness is computed in a general and conservative way following the ratio presented in section II. In the first scenario, the nodes are sorted with respect to the number of objects they collected trough their neighbors, while in the second scenario, they are sorted with respect to the amount of energy they spent to collect the maximum number of items. After this phase, the last 9% of the nodes (e.g. the weakest nodes) are chosen for the cognitive rewiring, and in addition 3% of the nodes are chosen for a random rewiring.

Whenever a node does not have a "sufficient" fitness, it eliminates a portion of unnecessary links (i.e. those links which point to nodes detected as non-hubs, in this work just 1 link) and proceeds to try to establish new connections using the hub list it has. Starting from the most relevant hub (i.e. the first from the list) and continuing towards the last one, the rewiring node tries to build new links. Finally, if no hubs have available links, because they have reached the maximum number of connections, the rewiring node adopts a random strategy and establishes a link towards the first available node it finds.

IV. Evaluation

We compared our model from section III with a randomized algorithm. For comparability reasons between the algorithms, they are kept similar, apart that the randomized algorithm is memoryless and therefore nodes have no knowledge about its surrounding and potential hubs they might connect to. Consequently, the randomized algorithm selects the nodes that have to rewire using the same method as the cognitive algorithm does; but where the cognitive algorithm prefers to connect to a hub, the randomized one chooses a random node.

For the evaluation, we used the two scenarios described in section II in order to test our algorithms. The initial network topology consists of $N = 200$ with a mean connectivity per node of 4. A total of $I = 50$ unique items is distributed among the nodes, where each node needs to retrieve $I_{max} = 45$ objects from its neighbors. We used this setting in order to analyze a network on a larger scale. Further, we also tested the algorithm for smaller networks, and the results imply a similar behaviour as presented here. We run the simulation 50 times on our Matlab cluster with different random seeds. Figure 2 shows an initial result for the first scenario and Figure 3 for the second. Both figures show values of the median run regarding final results of fitness and energy.

In the first scenario, the number of retrievable items shall be maximized. Therefore, the "weakest" nodes are determined by the sum of collected items. In Figure 2, it is shown that both approaches improve the initial topology significantly at the beginning. After having reached a plateau of 36 items, the randomized approach begins to oscillate, whereas the cognitive approach can exploit its knowledge of potential hubs and steadily micro-optimizes the topology up to more than a *mean* of 1.1 items by not having significant differences in their energy usage. We can also observe that the cognitive approach is less prone to oscillations.

The second scenario shown in Figure 3 shows the energy dynamics of both approaches. Each node has unlimited energy available, so that it is able to retrieve all necessary 45 items. The weakest nodes are now defined as nodes who consume the most energy of all. Hence, those are candidates

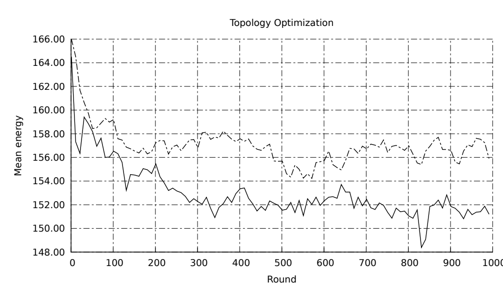


Figure 3. Energy minimization approach: mean energy usage per node of the randomized algorithm (dashed) and our approach (solid).

for rewiring in order to minimize the system's energy. The behaviours of both approaches are quite similar as in the fitness optimization from Figure 2. The initial topology improvement significantly reduces the energy consumption of the system. However, oscillation effects occur more often than in the first scenario. Our cognitive approach reduces the *mean* energy consumption of the nodes of more than 4.1 hops per node compared to the randomized algorithm.

V. Conclusion

In this work, we described how we optimize a topology by the means of a cognitive-inspired algorithm. The resulting online optimization problem was tackled with a cognitive model that enables a node to be self-aware about its surrounding community and eventually to detect and distinguish between hubs and non-hubs. This knowledge was exploited by a node to gain a more effective rewiring to other nodes than by random selection. We showed the effectiveness of our approach in two scenarios, in each comparing the achieved results to a randomized algorithm using the same network conditions. In the first scenario, the goal was to find a topology in which a maximum number of unique items can be retrieved for the system under a given energy constraint that was spent for "hopping". In the second one, we removed the energy constraint, so that nodes had enough energy for retrieving all items in each round, with the focus on decreasing the system's overall energy. In both scenarios, the cognitive-inspired algorithm performed significantly better than the random one.

Despite the fact that the algorithm uses global information for the selection of rewiring nodes, the approach shows first steps towards a pure self-organizing network since only local information is used for the hub detection. Overall, we showed first steps that information generated by a cognitive-inspired algorithm can be exploited in order to optimize network topologies. As future work, we plan to (i) deploy the algorithm on a wide range of *large scale* network topologies, (ii) localize the decision making of a node when to rewire or not, and (iii) further elaborate the used scenario by introducing more dynamics into items and nodes. We think that our algorithm is generic enough that it could also be used as a foundation in a wide area of applications beyond the scenario proposed here.

Acknowledgment

We would like to thank the organizers of the Awareness Summer School 2012 where we started our work on this topic. This research has received funding from the European Union 7th Framework Programme under grant agreement n° 257906 and n° 257756.

References

[1] R. Albert and A. L. Barabasi. *Rev. Mod. Phys.*, (74): 47, 2002.

[2] J. E. de Freitas, L.S. Lucena, L.R. da Silva, and H.J. Hilhorst. Critical behavior of a two-species reaction-diffusion problem. *Phys. Rev. E.*, (61):6330–6336, 2000.

[3] S. N. Dorogovtesev and J. F. F. Mendes. *Evolution of Networks*. Oxford University Press, Oxford, 2003.

[4] R. Dunbar. Neocortex size as a constraint on group size in primates. *Journal of Human Evolution*, 22(6): 469 – 493, 1992.

[5] Leon Festinger. *A Theory of Cognitive Dissonance*. Stanford University Press, 1957. ISBN 0804709114.

[6] K. I. Forster and C. Davis. Repetition priming and frequency attenuation. *Journ. Exp. Psyc.: Learning Memory and Cognition*, 10(4), 1984.

[7] G. Gigerenzer and W. Gaissmaier. Heuristic decision making. *Ann. Rev. of Psyc.*, (62):451–482, 2011.

[8] G. Gigerenzer and G. Goldstein. Models of ecological rationality: The recognition heuristic. *Psyc. Rev.*, 109 (1):75–90, 2002.

[9] L. Richard Hoffman. Group problem solving. volume 2 of *Advances in Experimental Social Psychology*, pages 99 – 132. Academic Press, 1966.

[10] A. Lancichinetti, S. Fortunato, and J. Kertész. Detecting the overlapping and hierarchical community structure of complex networks. *New Journal of Physics*, 033015(11), 2009.

[11] J Liang, R. Kumar, and K. W. Ross. The fasttrack overlay: a measurement study. *Comput. Netw.*, 50(6): 842–858, April 2006. ISSN 1389-1286.

[12] E. Massaro, F. Bagnoli, A. Guazzini, and P. Lió. Information dynamics algorithm for detecting communities in networks. *CNSNS Elsevier*, 17(11):4294 – 4303, 2012. ISSN 1007-5704.

[13] S. Milgram. The small world problem. *Psychology Today*, 61:60–67, 1967.

[14] J.D. Murray. *Mathematical biology*. Number v. 1 in Interdisciplinary applied mathematics. Springer, 2002.

[15] V. Nicosia, F. Bagnoli, and V. Latora. Impact of network structure on a model of diffusion and competitive interaction. *EPL*, 94(68009), 2011.

[16] J. Scott. *Social Networks Analysis: A Handbook*. Sage, London, 2nd, edition, 2000.

[17] H.A. Simon. A behavioral model of rational choice. *The Quarterly Journal of Economics*, 69(1):99–118, 1955.

[18] S.H. Strogatz. Exploring complex networks. *Nature(London)*, (410):268, 2001.

[19] E. Tulving, D. L. Schacter, and H. A. Stark. Priming effects in word fragment completion are independent of recognition memory. *Journ. Exp. Psyc.: Learning Memory and Cognition*, 8(4), 1982.

[20] S. Wasserman and K. Faust. *Social Networks Analysis*. University Press, Cambridge, England, 1994.