Optimal Message Interchange in a Self-organizing Multi-agent System

Antonio González-Pardo, Pablo Varona, David Camacho, and
Francisco de Borja Rodriguez Ortiz

Abstract. Over the last decade there has been a growing interest on Intelligent Agents and Multi-Agent Systems (MAS) in several fields such as Artificial Intelligence (AI), Software Engineering, Psychology, etc... Different problems can be solved in these fields by creating societies of agents that communicate with each other. Nevertheless, when the number of agents is large and the connectivity is extensive, the system suffers from overhead in the communication among agents due to the large number messages exchanged. This work addresses the search for an optimal communication topology to avoid these situations. This optimal topology is characterized by the use of a redirecting probability in the communication. The redirection of a communication is performed before the execution of the MAS. Once agents start the execution, the topology is fixed and remains unchanged. This characteristic is useful in those systems where a given topology can not be changed as, for example, in wired networks. On the other hand, in the proposed solution agents contain a local message discrimination process as a function of the sender of the message. Experiments show an important improvement in terms of a reduction in the number of iterations needed to solve the problem and also in the number of messages exchanged.

1 Introduction

A problem using Multi-Agent Systems appears when the system contains a large number of agents connected extensively. Agents need to stablish connections to interchange messages and realize a set of goals, or tasks, for which the system is

Antonio González-Pardo · Pablo Varona · David Camacho
· Francisco de Borja Rodriguez Ortiz
Departamento de Ingeniería Informática, Escuela Politécnica Superior,
Universidad Autónoma de Madrid
C/Francisco Tomás y Valiente 11, 28049 Madrid, Spain
e-mail: {antonio.gonzalez,pablo.varona,david.camacho,
f.rodriguez}@uam.es

M. Essaaidi et al. (Eds.): Intelligent Distributed Computing IV, SCI 315, pp. 131–141.
springerlink.com

designed. Depending on the topology of the network created and the number of agents in the system, overhead could appear due to the large number of messages interchanged, for example in broadcast topologies. To deal with the communication overhead, this paper studies the influence of the topology of the communication among agents. The main characteristic of the agents in the proposed solution is the discrimination of messages as a function of the sender in combination with an optimal topology that provides a faster propagation of the information without overhead the system.

The overhead problem in Multi-Agent Systems has been studied before and it is independent on the architecture used, [Jurasovic et al., 2006]. For example, in mobile agents it has been proposed the use of a dynamic agent distribution that reduces the communication between servers, [Jang and Agha, 2006]. This distribution is based on the idea of allocating agents with high communication rate between them, in the same server. This is useful because inter-node communication does not affect the overhead. But with this approach, a new problem appears which is how to select the agents with heavy traffic between them to migrate all of them into the same server. To solve this problem a new algorithm is proposed in [Miyata and Ishida, 2008].

Other works have addressed the discrimination of messages, [Sugawara and Lesser, 1993] and [Sugawara and Kurihara, 1998]. In these works agents learn what messages have high probability of being important based on rules created by each agent. Analysing the history of past inferences, agents create local rules that allow them to behave in a different way depending on the received message.

The discrimination technique described in this paper is a bio-inspired method based on the bursting activity of living neurons, [Szücs et al., 2003]. The bio-inspiration comes from the role of identity codes in neural systems which has been extensively with realistic models in [Latorre et al., 2006].

The main contribution of this paper is the search of an initial communication topology for a self-organizing Multi-Agent System. Static agents discriminate messages depending on the sender but their behaviour is not influenced by inference rules or learning procedures. This discrimination procedure has been applied previously in a new self-organizing neural network paradigm [Latorre et al., 2010] where the communication topology is dynamic and changes during the execution of the system. Although there are other approaches based on mobile agents ([Jang and Agha, 2006]), this work uses the optimization of the network topology to reduce the communication impact in the network. The communication topology suggested in this work deals with the communication overhead problem. This topology is based on a regular topology where each edge is redirected randomly according to a specific probability. Once the topology is created, it does not change during the execution of the system. Finally, our approach is illustrated by solving a jigsaw puzzle problem.

2 Description of the Model

This section describes the agent model, and the topology used to allow the optimal communication in the system.

2.1 Description of the Agent Model

Agents modelled in the system are static lightweight agents with a very low autonomy. However, most of the approaches of swarm and collective intelligence agents have the same behaviour and agents are indistinguishable. This work does not use this idea and each agent contains an unique identification that enables the differentiation between them. This characteristic is a key concept of the system because it will allow the discrimination of information as a function of the sender of the message.

In Multi-Agent Systems, agents have an incomplete information for solving the problem at hand. The piece of information contained in each agent is called, in this work, *Agent Information*, *[AIn]*, and the identification of each agent is called *Agent Identification*, *[AId]*. Both concepts compose the message that will be sent to the agents neighbourhood. Therefore, messages have the following structure $[AId|AIn]$. The first part of the message contains information that allows the identification of the sender, and the rest are data needed to solve the problem. Note that information stored in each part depends on the problem, *[AIn]* could contain much more information than *[AId]*. In these situations where Agent Information is large, it is very important the discrimination based on the recognition of the sender to avoid analyzing non relevant information.

The tasks performed by the agents will depend on the content of the messages received. That is, the behaviour of the agents will depend on who sends the information because depending on the sender, the message will be ignored or not. Nevertheless, the set of actions performed by the agents is always the same, there are not self-adaptation nor learning in its behaviour. Figure 1 shows the behaviour of agents.

Agents contain a temporal memory, called *local informational context*. Using this memory, agents can retain, during a certain period of time, a set of messages received by their neighbourhood. All received messages are stored in this memory, which means that it is not important whether the sender is recognized or not, because the message will be added to that memory in both cases. The fact that an agent does not recognize the sender of a message does not mean that any agent belonging to its neighbour will not recognize the sender. In the case where the received messages exceed the memory size, the oldest messages are replaced by the most recently ones. Finally, the information of the agent is added to this memory and all the content of the memory is sent.

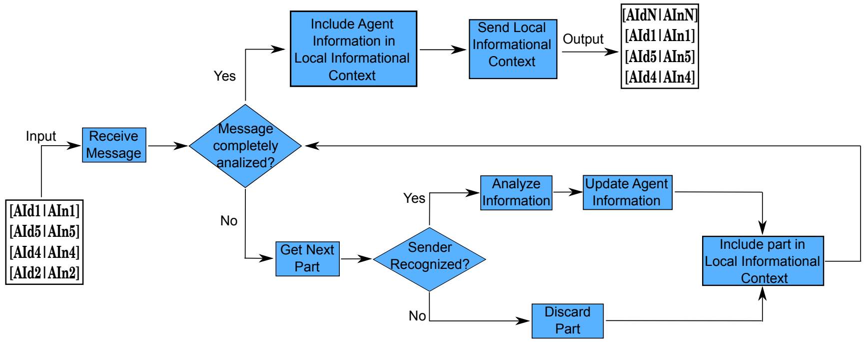

Fig. 1 Flow diagram that describes the behaviour of agent N. Each agent receives a set of messages, and starts analysing the sender of each message ($[AId]$). If the sender is not recognized, the message is discarded without analyzing the information contained ($[AIn]$). This discrimination procedure is really useful in environments where $[AIn]$ is bigger than $[AId]$ because agents does not analyze uninteresting $[AId]$ saving processing time.

2.2 *Network Topology*

The organization of any MAS can be analyzed from different perspectives which goes from hierarchical structures to completely random structures. However, it is necessary to provide some kind of structure to allow the interactions between agents. This organization can be studied from the point of view of a network topology.

In this work the topology selected to connect different agents is the Ring topology. This topology depends on a parameter named *connectivity degree (k)* that defines the number of connections that each node will have. Figure 2.a shows an illustrative example of a Ring Topology with $k = 1$, and 9 agents (nodes).

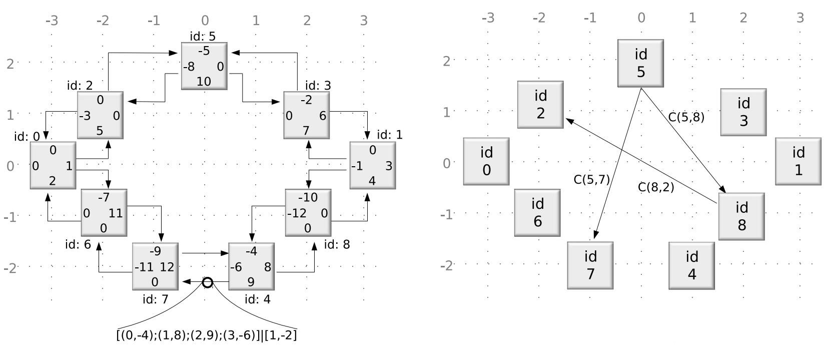

Fig. 2 Details of the communication topology. Figure on the left shows an example of a ring topology with 9 agents, and connectivity degree (k) 1. The figure on the right is a graphical representation of communication costs in the topology.

As it can be seen in Figure 2.a, given a particular connectivity degree, k, each node will be connected to $2*k$ agents. Note also that the connections are unidirectional. This means that if node A is the origin of a connection to node B, node B is not able to use that connection to send a message to node A, and thus node B need another connection that goes from B to node A.

One of the problems of regular topologies is that the propagation of the messages in the network is very slow. This is produced due to the characteristic path length is very high. This metrics represent the mean length of the paths in the network.

For a specific value of k, each node will be connected to its $2*k$ nearest nodes. Increasing the k of the topology, nodes will be highly clustered and the characteristic path will be decreased because the network will contain more edges. The minimum value for k is 1, in this situation each node will be connected to its 2 nearest nodes. On the other hand, the maximum connectivity corresponds with the broadcast mode, and this situation is produced when the connectivity degree is $n/2$, where n is the number of nodes in the network.

It is important to take into account that by increasing the connectivity degree of the topology, the system will converge in less iterations but the number of messages exchanged will also increase. Therefore, the value chosen for k is a trade off between the number of messages exchanged and the number of iterations to converge.

2.3 Searching an Optimal Communication Topology

As it was described in Section 2.2, the main problem in regular topologies is the value of the characteristic path length. From a communicational point of view, the lower the value of the characteristic path length is, the faster the messages are propagated.

In order to reduce the characteristic path length, a new parameter p is introduced in the system. This parameter is the probability of redirecting a connection. Given a specific value of p, each connection is analyzed and connections will be redirected with a probability p.

Using this new parameter, the connections are redirected to a randomly chosen node in the network, so the characteristic path length is reduced. Moreover, considering only the connectivity degree of the node (k, see Section 2.2) there is only one topology for each connectivity degree. Considering the parameter p, for each k and p fixed there are a family of topologies.

The probability of redirecting a connection was introduced by [Watts and Strogatz, 1998] in the definition of a Small World Network. It is important to note that for values of p close to 0 the network will have a regular topology, because few connections will be redirected. On the other hand, values close to 1 will generate Random Networks.

Small World Networks are characterized by having short path lengths between their nodes, which provides fast propagation of information, and high clustering rate that makes the network robust to attacks. This type of topology has been used to optimize the mutual information exchanged between nodes [Dominguez et al., 2009].

Finally when all connections have been treated by a fixed probability, agents start the execution and the topology created does not change.

3 Solving Puzzles Using an Optimal Communication Topology

There are many practical applications in which some agents need to know certain information about their environment in order to solve the problem. These applications can take advantage of the system described in this paper because the aim is to determine the optimal communication topology to solve the system by reducing the number of iterations and avoiding the overhead problem. Some examples of those applications, where our approach could be used, are scheduling problems, job shop problems, routing problems, etc.

The application selected to apply our approach tries to solve a puzzle where each piece needs to know which pieces match with its sides. Note that the neighbour of a piece does not necessarily contain pieces that matches with it. Using the technique described in this paper, each piece sends its information to the minimum number of agents to solve the puzzle in a reasonable amount of time.

Using the model to solve a puzzle, each agent will represent a single piece of the puzzle. Puzzles compose an image, and each piece contains a part of that global image. For that reason the agent information will be that part of the whole image contained in the piece.

3.1 Piece-Agent Codification

In order to model the shape of a side, each shape is represented by an integer value. For example, the border of the puzzle is represented as 0. With this approach, two pieces are compatible if the addition of the corresponding side values are equals 0 and none of these values are 0. This means that a piece with a side value X will match with the piece with value $-X$.

Using this representation, a message is relevant for an agent when the message contains, at least, one agent information compatible with any receiver agent sides. When a relevant message is received, the agent extracts the information about the compatible agent and memorizes that it must be connected to the sender through the corresponding side.

Each agent will be referred by its coordinates in a plane. That identification is unique because the agents are static and two agents cannot be located in the same place.

When an agent processes a relevant message, it updates its agent information, as shown in Figure 1. This means that agents need to delete from its agent information the side just matched. Each piece has four sides: upper side, right side, bottom side and left side. Those sides are identified by 0, 1, 2 and 3 respectively. In Figure 2.a, the piece with *AId* 4, which is located in $(1,-2)$, sends the message *[(0,-4);(1,8);(2,9);(3,-6)]*$|$*[1,-2]* to its neighbour.

3.2 Piece-Agent Communication

In real world, the communication between two partners has a cost which can be expressed in terms of monetary, time or performance cost. In this work, the cost of sending a message is proportional to the distance between the sender and the receiver. The cost of a communication is defined by equation 1.

$$C(i,j) = \lfloor \frac{EuclideanDistance(i,j)}{\min(EuclideanDistance(l,m))} \rfloor \forall l,m \mid l \neq m \tag{1}$$

The cost of a communication between two agents is the first positive number that exceeds the euclidean distance between both agents, normalized by the minimum distance in the system. Costs resulting from Equation 1, describe the number of iterations that a message will take to go from the sender to the receiver. The minimum cost is 1 and means, that receiver will have the message available in the next iteration. Figure 2.b shows a representation of the cost in the network.

4 Experimental Results

This section describes the different experiments carried out in this work. As it was described in section 3, the model has been adapted to solve a puzzle as an illustrative example.

In order to analyse the impact of the probability on the performance of the system, the original topology will be tested with with 21 different values of this probability. The values go from 0 to 1 with increases of 0.05.

All the experiments carried out in this work are based in a puzzle with 100 pieces, and all the shapes of the pieces are unique. This means that each piece will match, exactly, with one piece for each side. The algorithm initializes the agents and creates the topology defined by a fixed probability. Finally, agents are executed to solve the puzzle.

Figure 3 shows the iterations taken by the system to solve the puzzle. Those charts represents the number of iterations taken by the system to solve the problem for a specific probability value. The value of k is 3 for Figure 4.A, while the rest of figures corresponds to executions with $k = 8$.

In order to analyze how the memory size affects the performance of the system in terms of iterations taken to solve the puzzle, four experiments have been carried out. All experiments try to solve a puzzle with 100 pieces, and all pieces have $k = 3$. The memory size changes in each experiment, and it takes values 3, 8, 15 and 30. Figure 4 shows the performance of these experiments. From this figure, it is deduced that a larger memory provides an important improvement in the system with lower values of probability.

Furthermore, there is a limit probability from which there is no important improvement in the system. This limit is located between 0.3 and 0.4 and means that with this probability the characteristic path is very low.

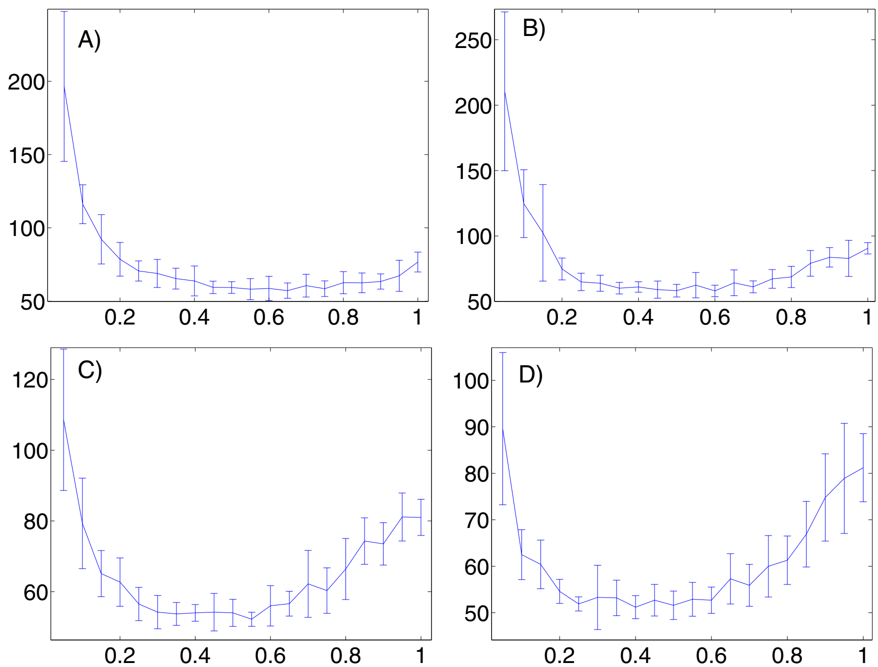

Fig. 3 Iterations per probability resulting from the execution of the system with different values for the memory and the connectivity degree of the topology. The system is composed by 100 agents that represent a puzzle with 100 pieces. For a specific probability, the system has been executed 10 times. Y-axis represents the number of iterations taken by the program, while X-axis shows the value of probability p. Figure A shows the performance of the system with memory 3 and $k = 6$. Figure B represents the system with memory 8 and $k = 3$. Finally, performance of the system with $k = 3$ and memory 15 and 30 is shown in Figures C and D.

Finally, as it was stated previously, the memory size affects the performance of the system. Nevertheless, the performance of a system with size memory 15 is very similar to the performance with memory 30. This fact suggest the existence of a limit in the memory size. This limit must be studied in future works because it is important to build a system that optimizes resources.

In order to measure the overhead problem in the system, the number of messages sent by the agents is compared with the a broadcast situation. The broadcast mode guarantees a solution, because each agent will received a message from the rest but the cost of sending a message affects the number of iterations taken to achieve the solution. Apart from this, designing a broadcast mode in a real system, for example, communication between hosts, it is not useful because to build a system with these characteristic could be very expensive.

Figure 5 shows the performance of the system taken into account the total number of messages sent. Although these messages could have different length, (because if

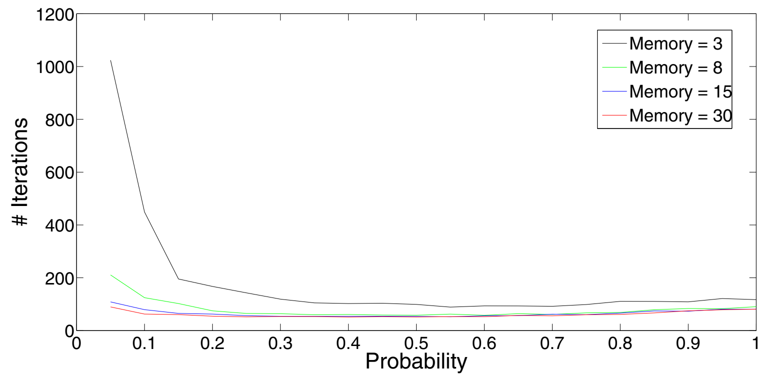

Fig. 4 Performance of the system, in terms of number of iterations taken to solve the puzzle, with memory 3, 8, 15 and 30. All executions try to solve a puzzle with 100 pieces, and each agent has $k = 3$.

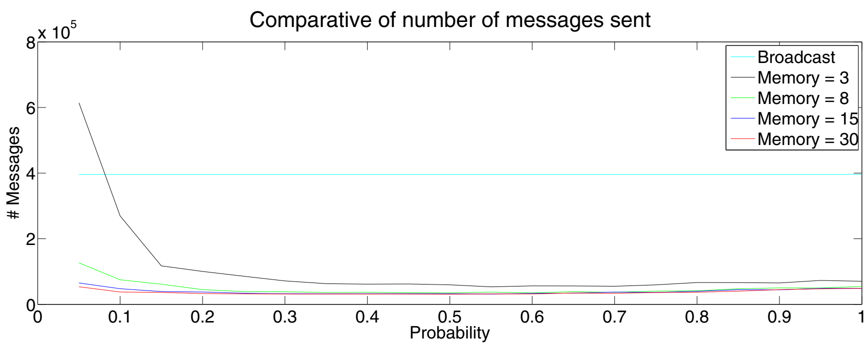

Fig. 5 Difference between the system and the broadcast. The connectivity degree of the topology in the broadcast mode is 50 ($n/2$), in the rest of executions the value of k is 3.

a piece has a side matched, the information belonging to that side is not sent) in these experiments, the suggestion that all messages have the same length is taken into account. This is not strange because, in this problem, the difference between the largest message and the smaller message is not relevant. Nevertheless, a more specific study on the total number of messages is required.

From the four experiments described in this section, the only execution that exceeds the number of messages sent by the broadcast mode is with memory 3 and lower probability (0.05). This result is expected because in this situation, the topology is very similar to the regular ring (with high characteristic paths) and the message size is very small. Nevertheless, with a little change in the probability (from 0.05 to 0.1) the number of sent messages is reduced around 56%. The rest of executions, independently of the probability, improve the number of messages sent in the broadcast situation.

5 Conclusions

This paper studies the problem of communication overhead in a network of agents analyzing the topology of the communication. The characteristics of this topology try to meet the following goals. On one hand, to reduce the geodesic path of the network in order to propagate messages in fewer iterations. On the other hand, to reduce the number of messages exchanged among agents and, in that way, to avoid the overhead problem.

This work modifies the regular topology with a specific probability, as a Small World Network. Nevertheless, it is important to notice that this redirection of communication channels is only performed once and when agents start the execution the topology is fixed. This concept is similar to a computer network where, once there is a connection between two devices or sub-networks, the connection is not redirected because the wires are physically inaccessible or the cost of redirecting is expensive.

The results show that modifying the probability of redirection, there are important improvements in the number of iterations needed to solve the problem (as compared to the regular topology). The value of the optimal probability depends on the connectivity degree, Figure 3 shows that for $k = 3$ the optimal probability is around 0.25 and 0.45, and for $k = 6$ the optimal probability is located between 0.55 and 0.65. As there is no direct relation between the connectivity degree and the probability, a deeper study is needed.

The memory of the agents plays an important role in the performance. Figure 4 shows that with larger values for the memory, the system solves the problem in fewer iterations. Taking into account the number of messages sent in the system, Figure 5 shows that with minor modifications in the communications (low probability), the number of messages sent is less than the number of messages sent in the broadcast situation.

Finally, the experimental results show how our approach allows to reduce significantly the communication overhead in a MAS by modifying the regular topology of the agent communication network.

Acknowledgements. This work has been partially supported by the Spanish Ministry of Science and Innovation under TIN 2007-65989, TIN 2007-64718 and MICINN BFU2009-08473.

References

[Dominguez et al., 2009] Dominguez, D., Gonzalez, M., Serrano, E., Rodriguez, F.B.: Structured information in small-world neural networks. Physical Review E (Statistical, Nonlinear, and Soft Matter Physics) 79(2), 021909 (2009)

[Jang and Agha, 2006] Jang, M.-W., Agha, G.: Agent framework services to reduce agent communication overhead in large-scale agent-based simulations. Simulation Modelling Practice and Theory 14(6), 679–694 (2006)

[Jurasovic et al., 2006] Jurasovic, K., Jezic, G., Kusek, M.: A performance analysis of multi-agent systems. ITSSA 1(4), 335–342 (2006)

[Latorre et al., 2006] Latorre, R., Rodríguez, F.B., Varona, P.: Neural signatures: multiple coding in spiking-bursting cells. Biological Cybernetics 95, 169–183 (2006)

[Latorre et al., 2010] Latorre, R., Rodríguez, F.B., Varona, P.: Signature neural networks: Definition and application to jigsaw puzzle solving. IEEE Transaction on Neural Networks (2010) (page to appear)

[Miyata and Ishida, 2008] Miyata, N., Ishida, T.: Community-based load balancing for massively multi-agent systems. In: Jamali, N., Scerri, P., Sugawara, T. (eds.) MMAS 2006, LSMAS 2006, and CCMMS 2007. LNCS (LNAI), vol. 5043, pp. 28–42. Springer, Heidelberg (2008)

[Sugawara and Kurihara, 1998] Sugawara, T., Kurihara, S.: Learning message-related coordination control in multiagent systems. In: Selected Papers from the 4th Australian Workshop on Distributed Artificial Intelligence, Multi-Agent Systems, pp. 29–44. Springer, Heidelberg (1998)

[Sugawara and Lesser, 1993] Sugawara, T., Lesser, V.: Learning coordination plans in distributed problem-solving environments. Technical report. In: Twelfth International Workshop on Distributed Artificial Intelligence (1993)

[Szücs et al., 2003] Szücs, A., Pinto, R., Rabinovich, M., Abarbanel, H., Selverston, A.: Synaptic modulation of the interspike interval signatures of bursting pyloric neurons. Journal of neurophysiology 89(3), 1363–1377 (2003)

[Watts and Strogatz, 1998] Watts, D.J., Strogatz, S.H.: Collective dynamics of ’small-world’ networks. Nature 393(6684), 440–442 (1998)