Resilient Self-Organizing Networks in Multi-Agent Systems via Approximate Random k-Regular Graphs

Zohreh Al Zahra Sanai Dashti, Diego Deplano, Carla Seatzu and Mauro Franceschelli

***Abstract*— This paper addresses the problem of making a network of cooperative agents more resilient against disconnections due to link or node failure, or DoS cyber-attacks. We propose a distributed protocol to let the network self-organize and maintain an approximate random k-regular graph topology, which has interesting robustness properties. The proposed method can be applied in a scenario where the agents communicate over an internet protocol, limited to two-hop interactions, and can log-in and log-out according to the framework of open multi-agent systems. We provide a preliminary characterization of the self-organization protocol, and a numerical validation with a comparison with the state-of-art.**

I. Introduction

In networks of collaborating agents, the pattern of interaction among the agents highly impacts the performance of the network. A compelling model of the network is drawn by graph theory: agents are modeled as nodes and their interactions are modeled as edges between nodes, thus constituting a graph. The properties of the graph modeling the pattern of interactions among the agents are crucial for characterizing several properties of the network, such as resilience to perturbations [1], [2], controllability [3]–[5], and feasibility of distributed algorithms [6], [7].

In several applications, multi-agent networks must deal with perturbations such as sudden disconnections of agents due to failures [8]–[10], or attacks carried out by malicious agents [11]. Intuitively, one of the worst events that should be avoided is the disconnection of the network into two or more components, which impedes the flow of information through the whole network. Several measures have been proposed in the current literature to quantify how well connected a graph is, which are mostly related to the number of nodes and edges that should be removed to make the graph disconnected. Some of the most established connectivity measures and node ranking are the algebraic connectivity and the Fiedler eigenvector [12], [13], the Kirchoff index [14], [15], the edge/node expansion ratio [16], [17]. The magnitude of any of these measures reflects the quality of connectivity of a graph, and it has been employed for characterizing its robustness and synchronizability [18].

Consequently, the design of algorithms to improve the connectivity of a graph according to different connectivity measures has recently attracted much attention [19]–[22]. A naive attempt is that of adding more edges to the graph, which increases the connectivity of the graph but it is not practically appropriate in several applications where each edge represents some virtual or physical link between the corresponding agents. Indeed, a high number of edges is usually not the desired fact in different applications due to higher costs, e.g., when the edges represent physical communication channels [23], higher risk of ripple effect, e.g., the propagation of fake news in a social network [24].

An interesting class of graphs that score high values for several connectivity measures while maintaining a low number of edges is the one of random regular graphs [17], [25]. A graph is said to be *k-regular* if each node has a number of incident edges (the degree) equal to k. A k-regular graph is said to be random if it is selected uniformly at random from the set of all k-regular graphs with the same number of nodes. This motivated the seminal series of works by [26]–[28] that led to a distributed protocol to transform any connected graph into a connected random regular graph with a similar number of edges as the initial graph.

The main contribution of this paper is a novel distributed protocol to reshape a graph into a random k-regular graph in open multi-agent networks, wherein agents may leave or join the network at any time. The main novelty of the proposed protocol is that it allows to arbitrarily choose the degree k of the regularity of the graph independently from the initial average degree m, in contrast to the approach of [28] where k is restricted to the range $[m, m+2]$.

Structure of the paper. The notation used in this paper along with some useful preliminaries are given in Section II. Section III first formalize the problem under study and then describes the proposed algorithm to solve the problem. In Section IV numerical simulations are provided which corroborate the validity of the proposed algorithm and its better performances with respect to the state-of-art. Concluding remarks and future perspectives are given in Section V.

II. Preliminaries

We consider networks of multiple agents whose pattern of interaction/communication is modeled by a *graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \ldots, n\}$ is the set of *nodes*, representing the agents, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of *edges* connecting the nodes, representing the point-to-point communication channels between the agents. We assume networks to be undirected, i.e., if $(i,j) \in \mathcal{E}$ and $(j,i) \in \mathcal{E}$, and therefore adjacency matrices are symmetric.

A path between two nodes $i, j \in \mathcal{V}$ is a sequence of consecutive edges $\pi_{ij} = (i,p), (p,q), \ldots, (r,s), (s,j)$ where

This work was partially supported by the Fondazione Banco di Sardegna with the grant "Formal Methods and Technologies for the Future of Energy Systems", cup F72F20000350007.

All the authors are with the DIEE, University of Cagliari, 09123 Cagliari, Italy. Emails: {zohreh.sanai, diego.deplano, carla.seatzu, mauro.franceschelli}@unica.it

M. Franceschelli is the corresponding author.

each successive edge shares a node with its predecessor. An undirected graph $\mathcal{G}$ is said to be *connected* if there exists a path π_{ij} between any pair of nodes $i, j \in V$. Nodes i and j are said to be *neighbors* if there exists an edge between them, i.e., $(i, j) \in \mathcal{E}$. The set of neighbors of the i-th node is denoted by $\mathcal{N}_i = \{j \in V : (i, j) \in E\}$. We consider graphs without self-loops, i.e., $i \notin \mathcal{N}_i$. Similarly, the set of 2*-hops neighbors* of agent i is denoted by $\mathcal{N}_i^2$, which includes only agents j such that there exists a path π_{ij} between i and j of exactly 2 edges. The *degree* of a node i is the number of neighbors and it is denoted by $d_i = |\mathcal{N}_i|$, where $|\cdot|$ denotes the cardinality of a set. Consequently, the minimum, maximum and average degree of the graph $\mathcal{G}$ are denoted by $d_{min}(\mathcal{G}) = \min_{i\in\mathcal{V}} d_i$, $d_{max}(\mathcal{G}) = \max_{i\in\mathcal{V}} d_i$ and $\bar{d}(\mathcal{G}) = \sum_{i\in\mathcal{V}} d_i/n$, respectively. The *degree matrix* is $D = \{d_{i,j}\} \in \mathbb{R}^{n\times n}$, which is diagonal and such that $d_{i,i} = d_i$. The *Laplacian matrix* of graph $\mathcal{G}$ is defined as $L = D - A \in \mathbb{R}^{n\times n}$, and its smallest nonzero eigenvalue is called the *algebraic connectivity* and is denoted by $\lambda_2(\mathcal{G})$.

III. Distributed self-organization of random k-regular graphs

We consider networks of agents that are allowed to establish or close connections between themselves, their neighbors, and their 2-hop neighbors. The knowledge of the 2-hop neighborhoods is a standard assumption in many distributed algorithm and protocols such as constructing structures [24,6], improved routing [20], broadcasting [9], and channel assignment [3]; indeed, it can be directly retrieved from the neighbors without causing any delay in the execution of the algorithm. The time is considered to be discretized such that each time step is indexed by $t = 1, 2, 3, \ldots$, and the number of the agents, as well as their interconnections, can change at each time step t.

These networks can be made more resilient to perturbations by a proper design of the graph describing the interconnection of the agents. In particular, the class of random regular graphs has been pivotal in the analysis of robust or resilient multi-agent systems, due to the intrinsic uniform connectivity they ensure throughout the network.

Definition 1 *A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is said to be k-*regular*, with $k \in \mathbb{N}$, if each node has exactly k neighbors, i.e.,*

$$\exists k \in \mathbb{N} : \qquad |\mathcal{N}_i| = k.$$

Definition 2 *A k-regular graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is said to be* random *if it is selected uniformly at random from all k-regular graphs with the same number of nodes.*

Due to the time-varying nature of the problem under study, a network with a regular graph is likely to have a node with degree different from all others, thus arousing the need of defining the subclass of *approximate* regular graphs.

Definition 3 *A k-regular graph is said to be* approximate *if there is only one node with degree different from k.*

The problem of interest in this work is that of increase the resiliency of a network against perturbations by providing a local and distributed procedure to iteratively reshape any connected graph toward a random (approximate) k-regular graph, despite the time-varying number of agents. Our only working assumption is that the initial network is connected, while we do not make any assumption on the final degree k of the network, which is a completely arbitrary parameter.

Assumption 1 *The initial network's graph is connected.*

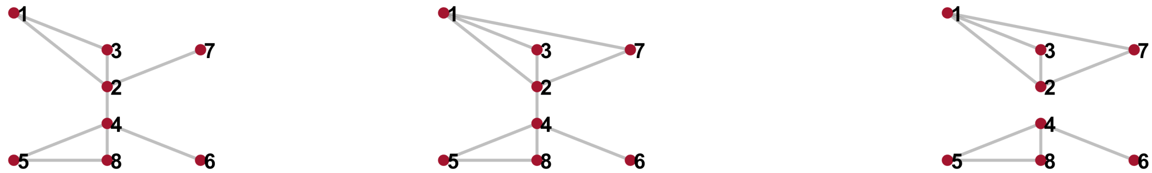
A. Proposed distributed protocol

The proposed estimation methodology is detailed in Algorithm 1, which envisages three potential operations/rules described next. At each time step, each agent $i \in \mathcal{V}$ is randomly activated and performs one of these rules according to the value of its own degree d_i and the desired degree k of the final approximate random k-regular graph. It is not necessary that all agents are sequentially activated at each time step, which would require some additional coordination among the agents, but their operations must not be concurrent.

- **Remove edges if $d_i > k$ (Rule 1).** Agent i selects at random an agent j from its neighborhood $\mathcal{N}_i$, such that the degree degree d_j of agent j is greater than k. If there is such agent j, then agent i closes the connection with agent j, i.e., edge (i, j) is removed from $\mathcal{E}$; otherwise, no action is taken.
- **Add edges if $d_i < k$ (Rule 2).** Agent i selects at random an agent j from its 2-hop neighborhood $\mathcal{N}_i^2$, such that

Algorithm 1: Distributed Formation and Maintenance of Random k-Regular Graphs

```
Input: Degree k ∈ ℕ.
at each step t = 1, 2, 3, ... do
    G = (V, E) is the current graph
    V_a ⊆ V is the set of randomly activated nodes
    each node i ∈ V_a does
        Compute N^>_{i,k} = {j ∈ N_i : d_j > k}
        Compute N^<_{i,k} = {j ∈ N_i^2 \ N_i : d_j < k}
        if d_i > k and N^>_{i,k} ≠ ∅ then            // Rule 1
            Select j ∈ N^>_{i,k} at random
            Remove edge (i, j) from E
        else if d_i < k and N^<_{i,k} ≠ ∅ then       // Rule 2
            Select j ∈ N^-_{i,k} at random
            Add edge (i, j) to E
        Compute N^≥_{i,k} = {j ∈ N_i : d_j ≥ k}
        if d_i ≥ k and N^≥_{i,k} ≠ ∅ then            // Rule 3
            Select j ∈ N^≥_{i,k} at random
            Select p ∈ N_i \ {j} at random
            Select q ∈ N_j \ {i} at random
            if (i, q) ∉ E and d_q < k then         // Rule 3a
                Add edge (i, q) to E
                Remove edge (i, j) from E
            else if (i, q), (j, p) ∉ E then        // Rule 3b
                Add edges (i, q), (j, p) to E
                Remove edges (i, p), (j, q) from E
```



(a) Initial graph. (b) Agent 1 executes Rule 2 selecting agent 7. (c) Agent 2 executes Rule 1 selecting agent 4.

Fig. 1: Example of Algorithm 1 execution that leads to a disconnected network.

agent j is not in its neighborhood $\mathcal{N}_i$ and its degree d_j is smaller than k. If there is such agent j, then it establishes the connection with agent j, i.e., edge (i, j) is added to $\mathcal{E}$; otherwise, no action is taken.

- **Move edges if** $d_i \geq k$ **(Rule 3).** Agent i selects at random an agent j from its neighborhood $\mathcal{N}_i$, such that the degree d_j of agent j is greater or equal to k. If there is such agent j, then both agents i and j select at random two agents p and q from their respective neighborhoods, excluding themselves; otherwise, no action is taken. Two cases are of interest:

  - (Rule 3a). If agents i and q are not linked and the degree d_q of agent q is less than k, then:
  1) the connection between agents i and j is closed, i.e., edge (i, j) is removed from $\mathcal{E}$;
  2) the connection between agents i and q is established, i.e., edge (i, j) is added to $\mathcal{E}$.

  - (Rule 3b). Otherwise, if agents i and q as well as agents j and p are not linked, then:
  1) the connections between agents i, p and j, q are closed, i.e., edges $(i, p), (j, q)$ are removed from $\mathcal{E}$;
  2) the connections between agents i, q and j, p are established, i.e., edges $(i, q), (i, p)$ are added to $\mathcal{E}$.

B. Characterization of Algorithm 1

In this preliminary work, we do not provide a rigorous theoretical characterization of Algorithm 1, but we do state some preliminary results regarding its convergence to a random regular graph while providing some proof sketches. For the sake of readability, we first introduce these results, along with their statements, and postpone the proof sketches to the next subsections.

The first result characterizes the behavior of a metric that measures the distance of a given network to a random regular graph. Such metric, whose inputs are the network $\mathcal{G}$ and the degree k, is given by

$$f(\mathcal{G}, k) = \|d_{max}(\mathcal{G}) - k\| + \|k - d_{min}(\mathcal{G})\|, \qquad (1)$$

where, we recall, d_{max} and d_{min} denote the maximum and the minimum degree among all agents in the network. We drop the input k and use the notation $f(\mathcal{G})$ whenever the degree k is clear from the context.

Theorem 1 *Consider a network with a fixed number of agents that implements Algorithm 1. If the graph is initially connected and remains connected thereafter, then:*

i) The metric $f(\mathcal{G})$ is non-increasing.

ii) At each time step there exists a set of agents that executing Algorithm 1 makes the metric $f(\mathcal{G})$ decrease.

The second result concerns the convergence of Algorithm 1 toward a random k-regular graph, possibly approximate.

Theorem 2 *Consider a network with a time-varying number of agents that implements Algorithm 1. At every step t_0 such that the number of agents remains constant for a sufficiently large window of time T, if the graph is connected for all $t \in [t_0, t_0 + T]$, then the graph converges for $t \to t_0 + T$ to an approximate k-regular graph.*

We emphasize that Algorithm 1 is suitable for open networks, i.e., it ensures convergence to a time-varying approximate k-regular graph even when the number of agents within the network changes. We claim that once the graph has been reorganized into an approximate k-regular graph, the algorithm uniformly randomizes this graph during its execution thus leading to an approximate random k-regular graph. In this preliminary work we provide no formal proof for this fact.

Conjecture 1 *The probability of existence of an edge between any pair of nodes, after a sufficiently long time in which no node enters or leaves the network during the execution of Algorithm 1 is uniform.*

As a final remark, we explicate that the execution of Algorithm 1 does not ensure that the network remains connected. This is likely the case when the initial graph has few nodes and the degree k is chosen too small. Next, we discuss a simple example that shows this behavior.

Consider a network of 8 agents as in Fig. 1(a) and choose $k = 3$. Assume that during the execution of the algorithm is first selected agent 1 such that $d_1 = 2 < 3$. Therefore, it executes Rule 2. If it selects agent 7 and adds the edge $(1, 7)$, the resulting graph is the one depicted in Fig. 1(b). Assume now that agent 2 is activated with $d_2 = 4 > 3$. Therefore, it executes Rule 1. If it selects agent 4 and removes the edge $(1, 4)$, the resulting graph is depicted in Fig. 1(c), which is a disconnected graph. This example shows that Algorithm 1 does not ensure to keep the graph connected. However, we conjecture that the probability of disconnection is inversely proportional to the algebraic connectivity and the chosen degree. Thus, if the initial algebraic connectivity is large enough, greater than 1, the algorithm won't disconnect the graph. We next formulate this conjecture, whose proof is postponed to future works.

Conjecture 2 *Higher values of the algebraic connectivity and of the chosen degree k implies lower disconnection probability while running Algorithm 1. Moreover, the probability of getting the graph disconnected goes to zero as the number of nodes and the degree k goes to infinity.*

Remark 1 *The connectivity of the network can be preserved by modifying Rule 1 as follows: agent i can close the communication with agent j only if $j \in \mathcal{N}_i^2$. However, this could slows down the convergence rate of the algorithm.*

C. Sketch of the proof of Theorem 1

We proceed by performing an exhaustive analysis of all cases that may occur during the execution of Algorithm 1.

1) It is activated an agent i with degree $d_i > k$

- Rule 1 may be executed. If this happens, the degree of agent i and the selected agent j are both strictly greater than k. Thus, the removal of the edge between these two agents implies that their degree reduces by 1 but remains greater or equal to k. This ensures that the metric $f(\mathcal{G})$ is non-increasing. Moreover, if one between these agents is the unique agent with maximum degree, then $d_{max}(\mathcal{G})$ decreases and $d_{min}(\mathcal{G})$ remains unaltered, thus $f(\mathcal{G})$ decreases.
- Rule 2 is not executed.
- Rule 3a may be executed. If this happens, the degree of agent i is strictly greater than k, the degree of agent j is greater or equal to k and the degree of agent q is smaller or equal than $k-1$. Therefore, the addition of the edge between i and q and the removal of the edge between nodes i and j imply that the degree of agent i remains unaltered, the degree of agent j decreases but remains greater or equal than $k-1$, while the degree of agent q increases but it remains smaller or equal to k. This ensures that the metric $f(\mathcal{G})$ is non-increasing. Moreover, if agent j is the unique agent with maximum degree or agent q is the unique agent with minimum degree, then either $d_{max}(\mathcal{G})$ decreases or $d_{min}(\mathcal{G})$ increases, while keeping unaltered the other one, thus $f(\mathcal{G})$ decreases.
- Rule 3b may be executed. If this happens, the degree of agent i is strictly greater than k, the degree of agent j is greater or equal to k, and the degree of agents p and q are smaller or equal than $k-1$. Therefore, the addition of edges between nodes i, q and j, p and the removal of edges between i, p and j, p imply that the degree of all these nodes remains unaltered. This ensures that the metric $f(\mathcal{G})$ is non-increasing.

2) It is activated an agent i with degree $d_i < k$

- Only Rule 2 may be executed. If this happens, the degree of agent i and the selected agent j are smaller or equal to $k-1$. Therefore, the addition of the edge between these two agents implies that their degree increases by 1 but remains smaller or equal to k. This ensures that the metric $f(\mathcal{G})$ is non-increasing. Moreover, if one between these agents is the unique agent with minimum degree, then $d_{min}(\mathcal{G})$ will be increased and $d_{max}(\mathcal{G})$ remains unaltered, thus $f(\mathcal{G})$ decreases.

3) It is activated an agent i with degree $d_i = k$

- Only Rules 3a and 3b may be executed. If this happens, the same arguments for the case $d_i > k$ hold.

Conclusion: It has been proved that the execution of any rule by any agent in the network does not increase the metric $f(\mathcal{G})$ in eq. (1), thus proving statement $i)$. It has been shown that if all the agents with the maximum degree execute one between Rule 1 or Rule 3a then the metric $f(\mathcal{G})$ decreases. The same occurs also when all agents with minimum degree execute Rule 2. This proves that $f(\mathcal{G})$ is eventually decreasing, i.e., there always exists a set of agents that executing Algorithm 1 led to a decrease of the metric $f(\mathcal{G})$, thus proving statement $ii)$.

D. Sketch of the proof of Theorem 2

Each graph at time $t+1$ generated by Algorithm 1 depends only on the graph at time t and not on the graphs before t. Let $\mathcal{G}_0$ be the graph at the generic instant of time t_0 and let $\mathcal{G}_t$ be the graph at time $t \geq t_0$. Assume that the initial graph $\mathcal{G}_0$ is connected but it is not an approximate k-regular graph, and assume also that there exists T such that all graphs $\mathcal{G}_t$ with $t \in [t_0, t_0+T]$ are connected and the number of agents does not change.

By Theorem 1 the metric $f(\mathcal{G}_t)$ eventually decreases until one of the following cases occurs:

1) $d_{min}(\mathcal{G}) = k$ **and** $d_{max}(\mathcal{G}) > k$**.**

In this case, the degree of each agent $i \in V$ is such that $d_i \geq k$, which may only execute one of the following rules:

- Rule 1 if $d_i > k$: i-th agent's degree decreases.
- Rule 3b if $d_i = k$: i-th agent's degree remains the same.

While the degree of any agent i with degree equals to k remains constant at each step due to Rule 3b, the degree of any agent i with degree greater than k will eventually reduce until there are at least two of them due to Rule 1. If only one agent is left with degree greater than k, then node i cannot execute any rule and its degree remains stuck at $d_i > k$. If instead, the execution of Rule 1 reduces the degree of the last two nodes with degree $k+1$, then all nodes achieve degree equal to k. Therefore, the graph converges to a k-regular graph, possibly approximate.

2) $d_{min}(\mathcal{G}) < k$ **and** $d_{max}(\mathcal{G}) = k$**.**

In this case, the degree of each agent $i \in V$ is such that $d_i \leq k$ and it may only execute one of the following rules:

- Rule 2 if $d_i < k$: i-th agent's degree increases.
- Rule 3 if $d_i = k$: i-th agent's degree remains the same.

By similar reasoning as to the previous case, we conclude that the graph converges to a k-regular graph, possibly approximate.

3) $d_{min}(\mathcal{G}) = d_{max}(\mathcal{G}) = k$**.**

In this case, the graph is already a k-regular graph, and only Rule 3b can be executed, which does not modify the degree of any agent.

(a) Initial graph. (b) final graph via Algorithm 1 (c) final graph via algorithm in [28].

Fig. 2: Simulation results of Section IV-A.

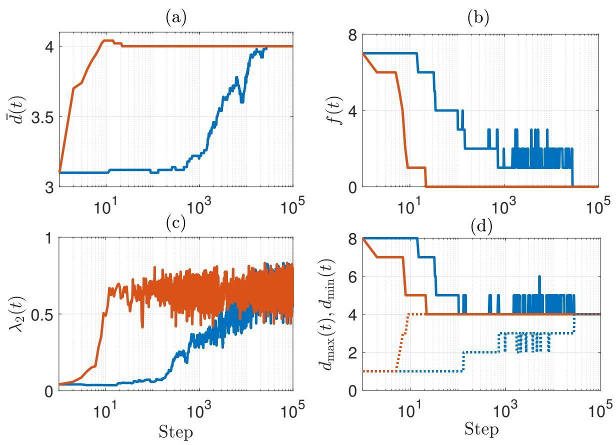


Fig. 3: This figure shows the time-evolution of (a) the average degree, (b) the metric in eq. (1), (c) the algebraic connectivity, (d) maximum/minimum degree, during the execution of (red) Algorithm 1 and (blue) algorithm in [28].

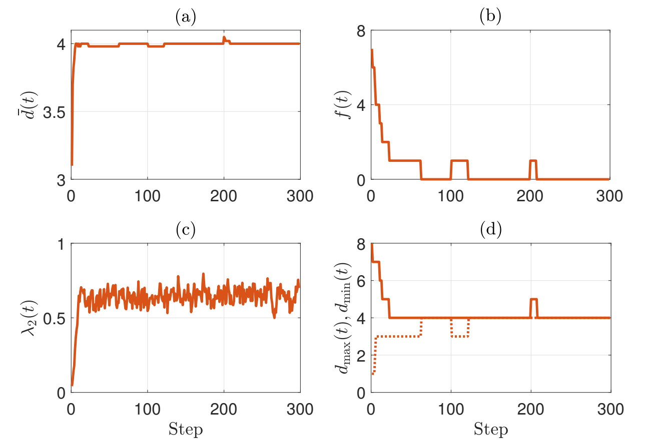


Fig. 4: This figure shows the time-evolution of (a) the average degree, (b) the metric in eq. (1), (c) the algebraic connectivity, (d) maximum/minimum degree, during the execution of Algorithm 1 in the case of open networks.

Conclusion: We have shown that if the number of nodes remains constant for a sufficiently large time window T, then Algorithm 1 eventually produces a k-regular graph for $t \to t_0 + T$, possibly approximate.

IV. Numerical Simulations

In this section, we provide two different simulations that validate the functioning of the proposed algorithm. First, we consider the case when the agents are not allowed to join or leave the network, and we provide a comparison with the protocol proposed by [28]. Secondly, we consider the scenario of an open multi-agent system.

A. Closed networks and comparison with the state-of-art

We assume that the initial graph $\mathcal{G}_0$ is connected and consists of $n = 100$ agents, with average degree equal to $\bar{d}(\mathcal{G}_0) = 3.1$ which is depicted in Fig 2(a). In order to provide a fair comparison, the goal degree is selected to be $k = 4$, such that the protocol proposed in [28] is feasible and results in a random m-regular graph with $m \in \{4, 5\}$. Indeed, by looking at Figs. 2(b)-(c) it can be verified that the final networks achieved by Algorithm 1 and algorithm in [28] are both regular with degree $k = m = 4$.

In Fig. 3(a) we compare the variation of the average degree of the network while running both algorithms. Fig. 3(a) reveals that the rate of convergence of Algorithm 1 is way higher since a steady state is reached in less than 30 steps, instead of the 3000 steps required by the algorithm in [28]. Theoretical characterization of the convergence rate of Algorithm 1 is postponed to future investigations.

Fig. 3(b) illustrates the time evolution of the metric in eq. (1) evaluated at each step while running both algorithms. According to Theorem 1, the metric shows a non-increasing behavior, furthermore, it eventually decreases until the optimum goal value is reached. On the contrary, the metric evolution is not monotonic in the case of algorithm in [28].

Moreover, in Figs. 3(c)-(d) we also plot the algebraic connectivity and the maximum/minimum degree among the nodes for both algorithms. Our final remark concerns the algebraic connectivity of the graph, whose value in the case of random k regular graphs is well-approximated by $\lambda_2 \approx k - 2\sqrt{k-1}$, and the result is tighter as the number of nodes increases. In our case it holds that $k - 2\sqrt{k-1} \approx 0.536$. Indeed, both algorithms construct graphs with algebraic connectivity close to this value, but Algorithm 1 achieves it in a smaller time.

B. Open networks

We consider the same set-up of the previous example: the initial graph $\mathcal{G}_0$ is connected and depicted in Fig. 2(a), it consists of $n = 100$ agents and its average degree is $\bar{d}(\mathcal{G}_0) = 3.1$. The join/leave events occur as described next:

- At $t = 100$ an agent leaves the network. Since the degree of the neighbors of the departing agent is $k - 1 = 3$, such event implies that the minimum degree of the network reduces, and thus the metric in eq. (1) increases when it occurs.
- At $t = 200$ an agent joins the network. Since the agent joins with degree equal to $k = 4$, and since the degree of its neighbors increases to $k = 5$, such event implies that the maximum degree of the network increases, and thus the metric in eq. (1) increases when it occurs.

The results of the simulation are shown in Fig. 4. From Fig. 4(a), one can appreciate that the average degree of the network eventually converges to the desired k whenever an agent joins or leave the network. However, these events produce a temporary deviation from the k-regular graph. This can be appreciated by looking at Fig. 4(b) which shows that the metric in eq. (1) increases in correspondence of these events, due to the changes of the maximum/minimum degree shown in Fig. 4(d). The time evolution of the algebraic connectivity depicted in Fig. 4(c) does not seem to be affected by these events while remaining close to the expected value.

V. Conclusions

In this paper we have presented a novel algorithm to reorganize in a distributed way a random approximate k-regular graph which can be executed in open multi-agent networks. The algorithm provides a set of local interaction rules to be performed by the agents to cut, add and move connection links with their 1-hop and 2-hop neighbors, thus leading to a randomized time-varying approximated k-regular graph. This kind of networks possess a number of properties that make them robust and resilient to sudden disconnections of links or nodes, possibly due to cyberattacks.

References

[1] B. Kaviarasan, R. Sakthivel, C. Wang, and F. Alzahrani, “Resilient control design for consensus of nonlinear multi-agent systems with switching topology and randomly varying communication delays,” *Neurocomputing*, vol. 311, 2018.

[2] Y. Mo, S. Dasgupta, and J. Beal, “Stability and resilience of distributed information spreading in aggregate computing,” *IEEE Transactions on Automatic Control*, 2022.

[3] M. Franceschelli, S. Martini, M. Egerstedt, A. Bicchi, and A. Giua, “Observability and controllability verification in multi-agent systems through decentralized laplacian spectrum estimation,” in *49th IEEE Conference on Decision and Control*, 2010.

[4] B. Liu, H. Su, L. Wu, and X. Shen, “Controllability for multi-agent systems with matrix-weight-based signed network,” *Applied Mathematics and Computation*, vol. 411, 2021.

[5] Z. Ji, H. Lin, S. Cao, Q. Qi, and H. Ma, “The complexity in complete graphic characterizations of multiagent controllability,” *IEEE Transactions on Cybernetics*, vol. 51, no. 1, 2021.

[6] D. Deplano, M. Franceschelli, and A. Giua, “Dynamic min and max consensus and size estimation of anonymous multi-agent networks,” *IEEE Transaction on Automatic Control*, 2021.

[7] K.-T. Foerster, J. Hirvonen, Y.-A. Pignolet, S. Schmid, and G. Tredan, “On the feasibility of perfect resilience with local fast failover,” in *Symposium on Algorithmic Principles of Computer Systems*, SIAM, 2021.

[8] J.-N. Li, X. Liu, X.-F. Ru, and X. Xu, “Disturbance rejection adaptive fault-tolerant constrained consensus for multi-agent systems with failures,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 12, 2020.

[9] T. Murayama and L. Sabattini, “Robustness of multi-robot systems controlling the size of the connected component after robot failure,” *IFAC-PapersOnLine*, vol. 53, no. 2, 2020.

[10] A. Jamakovic and S. Uhlig, “On the relationship between the algebraic connectivity and graph’s robustness to node and link failures,” in *2007 Next Generation Internet Networks*, 2007.

[11] Y. Wang, H. Ishii, F. Bonnet, and X. Défago, “Resilient consensus against mobile malicious agents,” *IFAC-PapersOnLine*, vol. 53, no. 2, 2020.

[12] M. Fiedler, “Algebraic connectivity of graphs,” *Czechoslovak mathematical journal*, vol. 23, no. 2, 1973.

[13] D. Deplano, M. Franceschelli, A. Giua, and L. Scardovi, “Distributed fiedler vector estimation with application to desynchronization of harmonic oscillator networks,” *IEEE Control Systems Letters*, vol. 5, no. 2, 2021.

[14] D. J. Klein and M. Randić, “Resistance distance,” *Journal of mathematical chemistry*, vol. 12, no. 1, 1993.

[15] B. Zhou and N. Trinajstić, “On resistance-distance and kirchhoff index.,” *Journal of mathematical chemistry*, vol. 46, no. 1, 2009.

[16] M. S. Pinsker, “On the complexity of a concentrator,” in *7th International Telegraffic Conference*, Citeseer, vol. 4, 1973.

[17] N. Alon, “Eigenvalues and expanders,” *Combinatorica*, vol. 6, no. 2, 1986.

[18] A. H. Dekker and B. D. Colbert, “Network robustness and graph topology,” in *Proceedings of the 27th Australasian conference on Computer science-Volume 26*, 2004.

[19] A. Beygelzimer, G. Grinstein, R. Linsker, and I. Rish, “Improving network robustness by edge modification,” *Physica A: Statistical Mechanics and its Applications*, vol. 357, no. 3-4, 2005.

[20] C. M. Schneider, A. A. Moreira, J. S. Andrade, S. Havlin, and H. J. Herrmann, “Mitigation of malicious attacks on networks,” *Proceedings of the National Academy of Sciences*, vol. 108, no. 10, 2011.

[21] M. Capalbo, O. Reingold, S. Vadhan, and A. Wigderson, “Randomness conductors and constant-degree lossless expanders,” in *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, 2002.

[22] M. Morgenstern, “Existence and explicit constructions of q+ 1 regular ramanujan graphs for every prime power q,” *Journal of Combinatorial Theory, Series B*, vol. 62, no. 1, 1994.

[23] L. Blume, D. Easley, J. Kleinberg, R. Kleinberg, and É. Tardos, “Which networks are least susceptible to cascading failures?” In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, 2011.

[24] A. Ganesh, L. Massoulié, and D. Towsley, “The effect of network topology on the spread of epidemics,” in *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, vol. 2, 2005.

[25] J. Friedman, “A proof of alon’s second eigenvalue conjecture,” in *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, 2003.

[26] A. Y. Yazıcıoğlu, M. Egerstedt, and J. S. Shamma, “Decentralized degree regularization for multi-agent networks,” in *52nd IEEE Conference on Decision and Control*, 2013.

[27] A. Y. Yazıcıoğlu, M. Egerstedt, and J. S. Shamma, “Decentralized formation of random regular graphs for robust multi-agent networks,” in *53rd IEEE Conference on Decision and Control*, 2014.

[28] A. Y. Yazıcıoğlu, M. Egerstedt, and J. S. Shamma, “Formation of robust multi-agent networks through self-organizing random regular graphs,” *IEEE Transactions on Network Science and Engineering*, vol. 2, no. 4, 2015.