Algebraic Connectivity Control and Maintenance with Application to Open Multi-Agent Networks ⋆,⋆⋆

W. Zhao [a,b], D. Deplano [b], Z. Li [a,c], A. Giua [b], M. Franceschelli [b]

[a] *SEME, Xidian University, Xi'an, China*

[b] *DIEE, University of Cagliari, Cagliari, Italy*

[c] *ISE, Macau University of Science and Technology, Taipa, Macau*

Abstract

This paper tackles the challenge of improving connectivity in multi-agent networks with a distributed protocol. The adopted strategy involves the design of local interaction rules for the agents to locally modify the graph topology by adding and removing links with neighbors while keeping the number of connections limited and exploiting only locally available information. Two distributed protocols are presented to boost the algebraic connectivity of the network graph beyond $k - 2\sqrt{k-1}$ where $k \in \mathbb{N}$ is a free design parameter; these two protocols are achieved through the distributed construction of (approximate) random k-regular graphs. One protocol leverages coordinated actions between pairs of neighboring agents and is mathematically proven to converge almost surely to the desired graph topology. The other protocol relies solely on the uncoordinated actions of individual agents and it is validated by a spectral analysis through Monte Carlo simulations. Numerical simulations offer a comparative analysis with other state-of-the-art protocols, showing the ability of both proposed protocols to maintain high levels of connectivity despite node failures or planned disconnections.

1 Introduction

Ad-hoc multi-agent networks refer to the groups of agents, either static or mobile, that communicate directly with each other through communication links, wirelessly or over the internet. These networks are inherently characterized by a time-varying pattern of interactions among agents, often accompanied by a dynamic set of participants, where new agents may join while others leave, forming what are known as *open networks* [1, 2] or *open multi-agent systems* (OMAS) [3–7]. An unstructured peer-to-peer network [8] for file-sharing over the internet also falls under this scenario. An important application is that of multi-robot systems or swarms, where the networks of robots may change in composition and size while executing tasks. However, most theoretical results do not account for this scenario, instead focusing on static cases or changes that occur only due to failures or faults [9]. Few studies address connectivity maintenance [10–12] in such open networks. Another important application consists in the networks of IoT devices involved in distributed control, optimization and monitoring tasks, for instance for energy management in residential areas [13, 14]. These networks of devices coordinate themselves in distributed fashion to preserve user data privacy and need to maintain unstructured peer-to-peer networks for exchanging information required to achieve global objectives, such as demand response or demand-side management programs. Other important applications include maintenance of sensor networks [15, 16], optimization and learning over networks [17, 18], majority and practical consensus [19, 20], and so on.

A critical task in OMAS is preventing network disconnection into disjoint components as the topology changes due to the arbitrary decisions of participating agents. Such disruptions hinder information flow and, in turn,

⋆ The work of Diego Deplano was supported by the project e.INS-Ecosystem of Innovation for Next Generation Sardinia (cod. ECS 00000038) funded by the Italian Ministry for Research and Education (MUR) under the National Recovery and Resilience Plan (NRRP) - MISSION 4 COMPONENT 2, "From research to business" INVESTMENT 1.5, "Creation and strengthening of Ecosystems of innovation" and construction of "Territorial R&D Leaders".

⋆⋆The work of Alessandro Giua was supported by the MUR National Recovery and Resilience Plan funded by the European Union - NextGenerationEU under project SERICS (PE00000014).

Email addresses: `wnjzhao@stu.xidian.edu.cn` (W. Zhao), `diego.deplano@unica.it` (D. Deplano), `zhwli@xidian.edu.cn` (Z. Li), `giua@unica.it` (A. Giua), `mauro.franceschelli@unica.it` (M. Franceschelli).

[1] M. Franceschelli is the corresponding author.

compromise the functioning of algorithms that rely on network connectivity. Agents and their interconnections are typically modeled using graph theory, where nodes represent agents and edges denote their point-to-point interactions. Graph connectivity is quantified using various metrics, primarily based on the minimum number of nodes and edges required to disconnect the graph. Key measures include algebraic connectivity and the Fiedler eigenvector [21, 22], the Kirchhoff index [23–25], and the edge/node expansion ratio [26]. Algorithms designed to improve graph connectivity based on various measures have recently gained significant attention [27–31]. A simple way to improve connectivity is by adding edges, but this is often impractical when edges represent virtual or physical links between agents [32]. Random regular graphs represent an interesting class of graphs that score high on several connectivity measures while maintaining a low number of edges [33, 34]. A graph is *k-regular* if every node has exactly k-connections, and it is classified as *random* if it is selected uniformly at random from the set of all k-regular graphs with the same number of nodes. Yazıcıoğlu *et al.* developed a distributed protocol enabling the agents to open and close connections between them in a way that the graph topology is iteratively reshaped into a random regular graph [35, 36]. More precisely, their algorithm works for any connected graph with an initial average degree $d_{\text{AVG}} > 2$ and allows to construct a random k-regular graph with a regularity degree such that $k \in [d_{\text{AVG}}, d_{\text{AVG}} + 2]$. Dashti *et al.* introduced another algorithm to construct an *inexact* random k-regular graph—where all but one node have degree equal to k—which allows for an arbitrary choice of the degree k and with a faster convergence rate [37], at the cost of not ensuring connectivity maintenance.

The **main contribution of this paper** is the design and characterization of two distributed protocols that continuously self-organize a network, by local modification of the graph topology, to boost its connectivity:

- Protocol 1 enables the construction of random k-regular graphs with arbitrary degree $k \in \mathbb{N}$, while maintaining connectivity;
- Protocol 2 generates *approximate* k-regular graphs, in which each node's degree lies within $[k, k + 1]$. This protocol offers improved performance—with respect to Protocol 1—in terms of the number of edge additions and removals needed to reach the desired topology, but it does not guarantee the preservation of connectivity [2].

By achieving the desired topology, both protocols increase the algebraic connectivity of the Laplacian matrix to at least $k-2\sqrt{k-1}$, where $k \in \mathbb{N}$ is a freely chosen design parameter. Numerical simulations corroborate the theoretical findings, showing how both protocols are able to reduce algebraic connectivity degradation in OMAS while handling high-frequency node join or leave events.

[2] A preliminary version of this manuscript appeared in [38].

Structure of the paper. Preliminaries are introduced in Section 2. In Section 3, two distributed protocols are proposed to control the algebraic connectivity of a graph by rewiring its topology. Section 4 presents numerical simulations that validate the proposed protocols and compare them with the state-of-the-art. Concluding remarks are provided in Section 5.

2 Preliminaries on graph theory and networks

A multi-agent system (MAS) with $n \in \mathbb{N}$ agents is modeled by an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \ldots, n\}$ represents the agents in the network, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ represents bidirectional communication channels among the agents. A *path* between two nodes i and j is a sequence of adjacent edges, and the graph is said to be *connected* if there exists a path between every pair of nodes. Two nodes are *neighbors* if they share an edge, with the neighborhood of node i given by $\mathcal{N}_i = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$.

The *degree* of node i is defined as $d_i = |\mathcal{N}_i|$, which leads to the following definitions for the minimum, maximum, and average degrees of the graph: $d_{\text{MIN}}(\mathcal{G}) = \min_{i\in\mathcal{V}} d_i$, $d_{\text{MAX}}(\mathcal{G}) = \max_{i\in\mathcal{V}} d_i$, $d_{\text{AVG}}(\mathcal{G}) = \frac{1}{n}\sum_{i\in\mathcal{V}} d_i$. The structure of the graph is captured by the *degree matrix* $D = \operatorname{diag}(d_1, \ldots, d_n)$ and the *adjacency matrix* $A_n = \{a_{i,j}\} \in \{0, 1\}^{n\times n}$, where $a_{i,j} = 1$ if and only if $(i, j) \in \mathcal{E}$. Denote the eigenvalues of A_n by μ_i and those of the *Laplacian matrix* $L_n = D - A_n$ by λ_i; both sequences are assumed to be sorted in ascending order (i.e., $\mu_i \leq \mu_{i+1}$ and $\lambda_i \leq \lambda_{i+1}$ for $i = 1, \ldots, n - 1$). The second smallest Laplacian eigenvalue, λ_2, is the *algebraic connectivity* of the graph.

We now formally define the class of *random k-regular graphs* with $k \in \mathbb{N}$, along with the notion of *approximate* k-regular graphs exploited in this manuscript [38]. Note that a k-regular graph may exist if and only if the product nk is even, whereas approximate regular graphs exist for any pair k, n such that $k < n$.

Definition 1 *[38, Def. 1-3]Let $\mathbb{G}^{\Delta}_{n,k}$ be the class of undirected graphs on n whose node degrees lie in the interval $[k, k + \Delta]$, where $n, k, \Delta \in \mathbb{N}$ and $k \in [1, n - 1]$. Then, for any graph $\mathcal{G} \in \mathbb{G}^{\Delta}_{n,k}$, we say that:*

- *$\mathcal{G}$ is "k-regular" if $\Delta = 0$;*
- *$\mathcal{G}$ is "approximate k-regular" if $\Delta = 1$;*
- *$\mathcal{G}$ is "random" if it is uniformly chosen from $\mathbb{G}^{\Delta}_{n,k}$.*

If a graph is k-regular, the eigenvalues of the Laplacian matrix equal those of the adjacency matrix with inverted sign and shifted by k. As known, a random k-regular graph, has the largest eigenvalue of the adjacency matrix A_n equal to k. All the other eigenvalues, as proved by Friedman [34], are bounded by $2\sqrt{k-1}$ with *high probability*—the probability that the second-largest eigenvalue does not exceed the bound tends to 1 for large values of k—when the graph is picked at random among all k-regular graphs. After recalling the concept of *high probability*, as commonly employed in the analysis of randomized algorithms [39], we summarize the corresponding results in Proposition 3.

Definition 2 *[39] Let $p(m)$ be the probability of occurrence of an event that depends on a parameter $m \geq 0$. We say that the event occurs with high probability if* $\lim_{m\to\infty} p(m) = 1$.

Proposition 3 *[34, Th. 1.1] Given a connected random k-regular graph with n nodes, it holds:*

- *the eigenvalues $\mu_1 \leq \cdots \leq \mu_n$ of its adjacency matrix A_n satisfy $\mu_n = k$ and $\max\{|\mu_{n-1}|, |\mu_1|\} \leq 2\sqrt{k-1}$ with (sublinear) high probability given by $p(n) = 1 - \mathcal{O}\big(n^{-\sqrt{k}}\big)$;*
- *the second smallest eigenvalue λ_2 of the Laplacian matrix L_n (i.e., the algebraic connectivity of the corresponding graph) satisfies $\lambda_2 \geq \lambda_{2,lb} := k - 2\sqrt{k-1}$ with high probability $p(n) = 1 - \mathcal{O}\big(n^{-\sqrt{k}}\big)$.*

In this work, we consider *unstructured* networks whose graph topology is not fixed but may change over time. In particular, we allow agents to perform local modifications by opening or closing communication channels with other agents. These events occur at discrete times indexed by $t = 0, 1, 2, \ldots$, thus giving rise to a time-varying graph $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$, with a fixed set of nodes and a time-varying set of edges. We call this a *random graph process* (RGP) which, starting with an initial graph $\mathcal{G}(0)$, generates a sequence of graphs $\{\mathcal{G}(0), \mathcal{G}(1), \mathcal{G}(2), \ldots\}$. By construction, the graph $\mathcal{G}(t+1)$ generated at time $t+1$ depends only on $\mathcal{G}(t)$; however, from a given graph $\mathcal{G}(t)$, several different graphs may be generated, each with a different probability. An RGP can generate only a finite set of graphs since the number of nodes is finite and equal to n, denoted by $\mathbb{G}_n$. One can associate to an RGP a finite discrete-time Markov chain $\mathcal{M} = (\mathcal{X}, P, \pi_0)$ such that a walk in $\mathcal{X}$ corresponds to a realization of the RGP, where $\mathcal{X}$ is the finite *set of states* representing all the possible graphs with n nodes, i.e., each state $X_i \in \mathcal{X}$ uniquely corresponds to a graph in $\mathcal{G}_i \in \mathbb{G}_n$; P is the *transition probability matrix* representing the probability of transitioning from one state to another, i.e., $p_{i,j} \in [0,1]$ represents the probability of generating graph $\mathcal{G}_j$ from $\mathcal{G}_i$; π_0 is the *initial probability distribution*, i.e., $\pi_0(i) = 1$ if $\mathcal{G}(0) = \mathcal{G}_i$, $X(0) = X_i$ and $\pi_0(i) = 0$ otherwise. With this notation, a sequence of states $\{X(0), X(1), X(2), \ldots\}$ corresponds to a sequence of graphs $\{\mathcal{G}(0), \mathcal{G}(1), \mathcal{G}(2), \ldots\}$. A subset of $\mathcal{X}$ is called a *component* if all states within it are *recurrent*, i.e., from any state in the component it is possible to reach all others. A component $\mathcal{X}_{abs} \subseteq \mathcal{X}$ is defined as *absorbing* if no state outside $\mathcal{X}_{abs}$ can be reached from any state within $\mathcal{X}_{abs}$, i.e., for every $i \in \mathcal{X}_{abs}$ and every $j \notin \mathcal{X}_{abs}$, $p_{i,j} = 0$.

Property 4 *[40, Sec. 1.3] Let $(\mathcal{X}, P, \pi_0)$ be a Markov chain with only one absorbing component $\mathcal{X}_{abs} \subseteq \mathcal{X}$. A sequence of states $X(t)$ almost surely hits $\mathcal{X}_{abs}$ in finite time and remains within it at subsequent times, i.e.,* $\text{Prob}\left(\exists t^\star > 0,\ \forall t \geq t^\star : X(t) \in \mathcal{X}_{abs}\right) = 1$.

An absorbing component is said to be *ergodic* if it is *aperiodic*, i.e., the maximum common divisor among the lengths of all cycles within it is one. A Markov chain is *ergodic* if it has a single ergodic component.

Property 5 *[40, Sec. 1.7] Given an ergodic Markov chain $(\mathcal{X}, P, \pi_0)$, there exists a stationary probability distribution $\pi_s : \mathcal{X} \mapsto [0,1]$ such that the probability that the chain is in state $X_i \in \mathcal{X}$ is equal to $\pi_s(i)$, i.e.,* $\lim_{t\to\infty} \text{Prob}(X(t) = X_i) = \pi_s(i)$, $\forall X_i \in \mathcal{X}$.

We are now in the position to formally define the concept of *uniform k-regularity* for RGPs, which clarifies what kinds of graphs are of interest in this paper: a *uniformly k-regular* RGP generates connected random k-regular graphs as $t \to \infty$ according to Definition 1.

Definition 6 *An RGP whose initial graph $\mathcal{G}(0)$ is connected with n nodes is said to be "uniformly k-regular" if the associated Markov chain $\mathcal{M} = (\mathcal{X}, P, \pi_0)$ is ergodic with $\mathcal{X}^\star \subseteq \mathcal{X}$, where each state $X_i \in \mathcal{X}^\star$ uniquely corresponds to a connected k-regular graph, and the stationary probability distribution π_s satisfies $\pi_s(i) = |\mathcal{X}^\star|^{-1}$ for all $X_i \in \mathcal{X}^\star$.*

3 Problem statement and proposed protocols

This research studies the problem of controlling the algebraic connectivity $\lambda_2(t)$ of the network's graph $\mathcal{G}(t)$ such that it is made and maintained arbitrarily large, making the network more resilient against disconnections. To this end, two distributed protocols are proposed to dynamically reshape the network's graph by allowing the agents to open and close communication channels. As a result, the topology tends toward that of random k-regular graphs, which enjoy high values of the algebraic connectivity—at least $\lambda_2(t) \geq k - 2\sqrt{k-1}$ in the limit as $n \to \infty$ (see Proposition 3). As k is a free design parameter, the protocols are thus able to increase and maintain the algebraic connectivity above any desired threshold.

The two proposed protocols are:

- In Protocol 1, we detail the functioning of the distributed Coordinated Formation Of Random k-REGular connected Graphs (CFOR-REG) Protocol (see Section 3.1). This protocol, which exploits coordination between pairs of neighboring agents, is proven to steer the network topology toward a random k-regular graph;
- In Protocol 2, we describe the distributed Uncoordinated Formation of Approximate k-REGular Graphs (UFA-REG) Protocol (see Section 3.2). This protocol, which only relies on independent decision of the agents thereby benefiting from an easier communication scheme, is proven to shape the network topology into an approximate k-regular graph. Moreover, empirical results show that its algebraic connectivity well approximate that of random regular graphs.

3.1 *Distributed formation of connected random k-regular graphs via the CFOR-REG Protocol*

This section presents the CFOR-REG Protocol detailed in Protocol 1, which allows for the distributed formation of a random k-regular graph exploiting coordination between neighboring agents, where k is an arbitrary even natural number. At each iteration $t = 1, 2, 3, \ldots$, each node is active with probability $1 - \epsilon \in (0, 1)$ and picks one of its neighbors $j \in \mathcal{N}_i$ uniformly at random. Active nodes broadcast to their neighbors information about their degree and which neighbor they picked randomly; thus each node i can build the list of nodes R_i that have randomly picked i, i.e., $R_i = \{j \in \mathcal{N}_i : j \text{ picked } i\}$, for all $i \in \mathcal{V}$. Each matched pair (i, j) such that $i \in R_j$ and $j \in R_i$, ordered such that $d_i \geq d_j$, has at its disposal the set of operations displayed in Figure 1, where operations labeled with A, R, M, and S denote the addition, removal, move, and swap of some edges, respectively. These operations are executed by randomly selecting one of the following rules:

- **Rule 1**: if $|R_i| \geq 2$ and $|R_j| \geq 2$, the pair attempts to swap two neighbors by executing operation (S) to randomize the edges in the graph;
- **Rule 2**: if $d_i > d_j$ and $|R_i| \geq 2$, the pair either attempts (with probability $1-\beta$) to move an edge by executing operation (M) in order to balance the degrees of the nodes, or attempts (with probability β) to add an edge by executing operation (A1);
- **Rule 3**: if $d_i > d_j$ and $|R_i| \geq 2$, the pair attempts (with probability β) to remove an edge by executing operation (R1);
- **Rule 4**: the pair either removes an edge by executing operation (R2) if they have at least one common neighbor h and both have degrees greater than k, or adds an edge via operation (A2) if $d_i < k$ and a private neighbor of node j has a degree less than k.

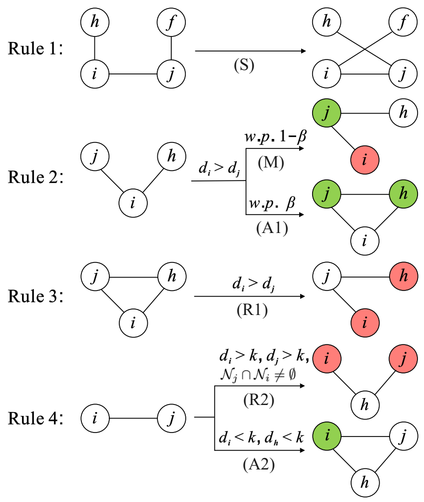

Fig. 1. Rules and operations of the CFOR-REG Protocol: the degree of the red nodes decreases, while the degree of the green nodes increases.

Protocol 1: CFOR-REG

Input: A connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $\epsilon, \beta \in (0, 1)$, and the desired integer degree $k \geq 2$

Output: A random k-regular graph

for $t = 1, 2, 3, \ldots$ **do**

  Each node $i \in \mathcal{V}$ activates with probability $1 - \epsilon$;

  Each active node $i \in \mathcal{V}_a$ picks a random $j \in \mathcal{N}_i$;

  For each node $i \in \mathcal{V}$ let $R_i = \{j \in \mathcal{N}_i : j \text{ picked } i\}$;

  foreach (i, j) *s.t.* $i \in R_j$, $j \in R_i$, $d_i \geq d_j$ **do**

    i picks at random a rule $r \in \{r_1, r_2, r_3, r_4\}$;

    if $r = r_1, |R_i| \geq 2, |R_j| \geq 2$ **then** `// Rule 1`

      i picks at random a node $h \in R_i \setminus \{j\}$;

      j picks at random a node $f \in R_j \setminus \{i\}$;

      if $(i, f) \notin \mathcal{E}, (j, h) \notin \mathcal{E}$ **then**

        $\mathcal{E} \leftarrow (\mathcal{E} \setminus \{(i, h), (j, f)\}) \cup \{(i, f), (j, h)\}$;

    else if $r = r_2, d_i > d_j, |R_i| \geq 2$ **then** `// Rule 2`

      i picks at random a node $h \in R_i \setminus \{j\}$;

      if $(j, h) \notin \mathcal{E}$ **then**

        i picks at random a value $\beta' \in [0, 1]$;

        if $\beta' > \beta$ **then**

          $\mathcal{E} \leftarrow (\mathcal{E} \setminus \{(i, h)\}) \cup \{(j, h)\}$;

        else

          $\mathcal{E} \leftarrow \mathcal{E} \cup \{(j, h)\}$;

    else if $r = r_3, d_i > d_j, |R_i| \geq 2$ **then** `// Rule 3`

      i picks at random a value $\beta' \in [0, 1]$;

      if $\beta' > 1 - \beta$ **then**

        i picks at random a node $h \in R_i \setminus \{j\}$;

        if $(j, h) \in \mathcal{E}$ **then**

          $\mathcal{E} \leftarrow \mathcal{E} \setminus \{(i, h)\}$

    else if $r = r_4$ **then** `// Rule 4`

      if $d_i > k, d_j > k, \mathcal{N}_j \cap \mathcal{N}_i \neq \emptyset$ **then**

        $\mathcal{E} \leftarrow \mathcal{E} \setminus \{(i, j)\}$;

      else if $d_i < k$ **then**

        $\mathcal{N}_{ji} := \{h \in \mathcal{N}_j \setminus (\mathcal{N}_i \cup \{i\})\}$;

        i picks at random a node $h \in \mathcal{N}_{ji}$;

        if $d_h < k$ **then**

          $\mathcal{E} \leftarrow \mathcal{E} \cup \{(i, h)\}$;

Theorem 7 *Consider an RGP due to the execution of the CFOR-REG Protocol starting from a connected graph $\mathcal{G}(0)$ with $n \geq k$ nodes and even $k \in \{4, 6, 8, \ldots\}$. Then:*

(i) *the graph $\mathcal{G}(t)$ remains connected at each time $t \geq 0$;*

(ii) *there almost surely exists a time $t^\star \in \mathbb{N}$ such that the graph $\mathcal{G}(t)$ is k-regular for all $t \geq t^\star$, i.e.,* $\text{Prob}(\exists t^\star > 0, \forall t \geq t^\star : \mathcal{G}(t) \textit{ is } k\textit{-regular}\,) = 1$;

(iii) *the RGP is "uniformly k-regular", i.e., it generates connected random k-regular graphs as $t \to \infty$;*

(iv) *the algebraic connectivity $\lambda_2(t)$ of the graph $\mathcal{G}(t)$ is, in expectation, greater than $k - 2\sqrt{k-1}$ as $t \to \infty$.*

PROOF. Statement (i) is trivial, which can be directly verified in a graphical way by looking at Figure 1, where all operations result in a connected subgraph. Hence, the entire graph remains connected.

Statement (ii) can be proved via Markov chain theory. Let $\mathcal{M} = (\mathcal{X}, P, \pi_0)$ be the Markov chain associated with the RGP, i.e. $\mathcal{X}$ represents the set of all possible graphs on n nodes; $P = \{p_{i,j}\}$ is the matrix whose elements represent the probability of transitioning between any two graphs; π_0 is the *initial probability distribution*, i.e., $\pi_0(i) = 1$ if $X_i \in \mathcal{X}$ is the state associated to graph $\mathcal{G}(0)$ and $\pi_0(i) = 0$ otherwise. Under this notation, statement (ii) is equivalent to asking that the subset of states corresponding to k-regular connected graphs, denoted by $\mathcal{X}^{reg}$, is the only absorbing component of the Markov chain (according to Property 4). To this end, consider any initial state $X(0) \notin \mathcal{X}^{reg}$, which does not correspond to a k-regular connected graph, i.e., $2|\mathcal{E}| \neq k|\mathcal{V}|$. Since k is assumed to be even, and for any number of nodes $|\mathcal{V}| = n$, there is always a number of edges satisfying $2|\mathcal{E}| = k|\mathcal{V}|$, a necessary condition for a k-regular graph. For simplicity, let us denote the initial graph by $\mathcal{G} = \mathcal{G}(0)$, and the graph reached by a random walk on the Markov chain after t transitions by $\mathcal{G}' = \mathcal{G}(t)$. Then, either one of the following two cases may occur.

Case 1: $2|\mathcal{E}| < k|\mathcal{V}|$, i.e., there are fewer edges than required, resulting in some nodes having degree less than k. We prove that $\mathcal{G}'$ can increase its edge count by considering the only two possible scenarios:

1) $\mathcal{G}$ is m-regular with $m < k$: then $d_i = m$ for all $i \in \mathcal{V}$. Moreover, there surely exists a pair of nodes $(i, j) \in \mathcal{E}$ such that $\mathcal{N}_{ij} = \mathcal{N}_i \setminus \{\mathcal{N}_j \cup \{j\}\} \neq \emptyset$. Note that if $\mathcal{N}_{ij} = \emptyset$ for all pairs of nodes $(i, j) \in \mathcal{E}$, then the graph must be complete, i.e., $m = n$, which contradicts the assumption $m < k$ since $k < n$. Thus, an extra edge can be added by executing operation (A2) in Rule 4.

2) $\mathcal{G}$ is not m-regular with $m \leq k$: then there must exist a pair of nodes $(i, j) \in \mathcal{E}$ such that $d_i > d_j$ and $\mathcal{N}_i \setminus \{\mathcal{N}_j \cup \{j\}\} \neq \emptyset$, i.e, i must have at least one neighbor that is not connected to j. Thus, an extra edge can be added by executing operation (A1) in Rule 2.

Case 2: $2|\mathcal{E}| > k|\mathcal{V}|$, i.e., there are more edges than needed, leading to some nodes having degree greater than k. We now prove that $\mathcal{G}'$ can decrease the number of edges. By [36, Lemma 4.8], the execution of Rules 1–2–3 can transform any graph $\mathcal{G}$ into a graph $\mathcal{G}'' = (\mathcal{V}, \mathcal{E}'')$ containing at least one triangle, i.e., there is a triplet (i, j, h) such that $\{(i, j), (j, h), (h, i)\} \subseteq \mathcal{E}''$. Thus, we consider $\mathcal{G}''$ as the new initial graph and analyze the only two possible scenarios:

1) $\mathcal{G}''$ is m-regular with $m > k$: then $d_i = m$ for all $i \in \mathcal{V}$. Moreover, there necessarily exists a pair of nodes $(i, j) \in \mathcal{E}$ such that $\mathcal{N}_j \cap \mathcal{N}_i \neq \emptyset$, since there is at least one triangle. Thus, an edge can be removed by executing operation (R2) in Rule 4.

2) $\mathcal{G}''$ is not m-regular with $m \geq k$: then either all triangles have nodes with the same degrees, or there is at least one triangle with nodes of different degrees. If the latter case holds, then an edge can be removed by executing operation (R1) in Rule 3. In the former case, we show that any triangle whose nodes have the same degree can be transformed into a triangle with nodes of different degrees. Consider a generic triangle composed of nodes (i, j, h) such that $d_i = d_j = d_h \geq 2$. Consider the shortest path from any node of this triangle to a node whose degree differs from that of the triangle nodes. Without loss of generality, let this path be between i and ℓ, and note that all nodes in this path have the same degree that is equal to d_i, while $d_\ell \neq d_i$. One of the following two cases must occur.

(2a) $d_\ell > d_i$: in this case, by executing Rule 2, node ℓ can always add an edge from one of its neighbors to a node on the shortest path, thus increasing its degree and, in turn, shortening the shortest path from i to a higher degree node. By induction, the degree of node i can eventually be increased, i.e., the degrees of the nodes in triangle (i, j, h) satisfy $d_i > d_j = d_h$.

(2b) $d_\ell < d_i$: in this case, we first note that $d_i \geq 3$ since i has two neighbors within the triangle and another neighbor on the shortest path. Therefore, the neighbor p of ℓ on the shortest path has at least one neighbor q outside that path. Since $d_p > d_\ell$ by assumption, by executing Rule 2 with operation (M), it is always possible to remove the edge (p, ℓ) and add the edge (q, ℓ), thus decreasing the degree of node p and, in turn, shortening the shortest path. By induction, the degree of node i can eventually be decreased, i.e., the degrees of the nodes in triangle (i, j, h) satisfy $d_i < d_j = d_h$.

Hence, in either case, edges of the network can be added or removed via the CFOR-REG Protocol until the network reaches an integer average degree of k. Assume now that the graph is still not k-regular, i.e., the set $\mathcal{V}_{>k}$ of nodes with degree greater than k and the set $\mathcal{V}_{<k}$ of nodes with degree less than k are not empty. Consider two nodes $i \in \mathcal{V}_{>k}$ and $j \in \mathcal{V}_{<k}$ with minimum distance $\delta_{ij} \in \mathbb{N}$ and the following cases:

(2c) $\delta_{ij} = 1$, i.e., nodes i and j are neighbors. In this case, node i has at least a neighbor h that is not a neighbor of j (as i has a higher degree than j). Therefore, node h can perform the operation (M) in Rule 2 to remove the edge with node i and add an edge to node j, thus causing both sets $\mathcal{V}_{>k}$ and $\mathcal{V}_{<k}$ to lose one node.

(2d) $\delta_{ij} > 1$, i.e., in the shortest path between i and j there are $\delta_{ij} - 1$ nodes of degree exactly k (since we are considering i and j of minimum distance). Let p and q be the first two nodes in the shortest path from i to j. In this case, node i has at least a neighbor h that is not a neighbor of j (as i has a higher degree than j). Thus, node i and its closest neighbor p on the shortest path to j can execute the swap operation (S) in Rule 1 and exchange neighbors h and q, thus reducing the distance δ_{ij} by 1.

Case (2d) can be repeated until case (2c) holds, causing both sets $\mathcal{V}_{>k}$ and $\mathcal{V}_{<k}$ to lose one node. Thus, these sets eventually become empty, i.e., the graph is reshaped into a k-regular graph. Once a k-regular graph is achieved, only the swap operation (S) in Rule 1 can be performed, thus keeping the degrees of all nodes equal to k. This proves that the subset of states $\mathcal{X}^{reg}$ corresponding to all k-regular connected graphs is the only absorbing component, thus completing the proof of statement (ii).

Statement (iii), according to Definition 6, corresponds to the case in which the Markov chain is ergodic on $\mathcal{X}^{reg}$, and the stationary probability distribution (which exists by Property 5) is uniform. As proved in statement (ii), once a random walk enters the absorbing component $\mathcal{X}^{reg}$, only the swap operation (S) in Rule 1 can be executed. This rule was originally presented in [41] as the "1-Flipper operation", in which it was proven that, starting from any connected k-regular graph on n nodes, in the limit as $t \to \infty$ this operation constructs all connected k-regular graphs with the same probability [41, Theorem 1]. This can be verified by noting that: 1) any k-regular connected graph can be constructed, i.e., all states in $\mathcal{X}^{reg}$ are recurrent; 2) there is a non-zero probability of remaining in the same graph, i.e., the transition probability matrix of the Markov chain has strictly positive diagonal entries and thus all states in $\mathcal{X}^{reg}$ are aperiodic; 3) for any two graphs $\mathcal{G}$ and $\mathcal{G}'$, the probability of constructing $\mathcal{G}'$ from $\mathcal{G}$ is the same as constructing $\mathcal{G}$ from $\mathcal{G}'$, i.e., the transition probability matrix of the Markov chain is symmetric and, in turn, the stationary probability distribution is uniform. From 1) and 2), it follows that the Markov chain is ergodic on $\mathcal{X}^{reg}$ and thus admits a stationary probability distribution π_s. From 3) it follows that $\pi_s(i) = |\mathcal{X}^{reg}|^{-1}$ for all graphs. Therefore, according to Definition 6, the RGP is uniformly k-regular, i.e., it generates connected random k-regular graphs as $t \to \infty$ according to Definition 1, completing the proof of statement (iii).

Statement (iv) holds because the RGP is uniformly k-regular and the algebraic connectivity of a random k-regular connected graph is bounded below in expectation by $k - 2\sqrt{k-1}$, see Proposition 3. □

3.2 Distributed uncoordinated formation of approximate k-regular graphs via the UFA-REG Protocol

This section presents the UFA-REG Protocol detailed in Protocol 2, which enables the distributed formation of an approximate k-regular graph, and does not require coordination between neighboring nodes. Consequently, the "swap" operation (S) of the CFOR-REG Protocol is not allowed now, since it would require coordination between a pair of neighboring nodes. In order to let each node locally modify the graph topology in a completely autonomous way while avoiding conflicting actions, we next introduce the concept of "*ownership of a graph's edges*".

In this protocol, we restrict the ability of nodes to remove or add edges by assigning each edge $(i,j) \in \mathcal{E}$ a unique owner (either node i or j). Thereafter, each edge can be removed only by its owner, and the owner of any newly added edge (i,j) is designated as the node responsible for its addition. We model this scenario by means of a directed *ownership graph* $\mathcal{G}_d = (\mathcal{V}, \mathcal{E}_d)$, which is a directed version of the undirected communication

Protocol 2: UFA-REG

Input: A connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $\epsilon, \beta \in (0,1)$, and the desired integer degree $k \geq 2$
Output: An approximate k-regular graph
for $t = 1, 2, 3, \ldots$ **do**
  Each node $i \in \mathcal{V}$ activates with probability $1 - \epsilon$;
  Pick at random a node $i \in \mathcal{V}_a$;
  while $d_i \leq k - 1$ **do** `// Rule 1`
    for $j \in \mathcal{N}_i$ **do**
      $\mathcal{N}_{ij} := \{s \in \mathcal{N}_j \setminus (\mathcal{N}_i \cup \{i\})\}$;
      if $\mathcal{N}_{ij} \neq \emptyset$ **then**
        i picks at random a node $s \in \mathcal{N}_{ij}$;
        $\mathcal{E} \leftarrow \mathcal{E} \cup \{(i,s)\}$;
        break `// exit the for loop`;
  i picks at random a rule $r \in \{r_2, r_3, r_4\}$;
  if $r = r_2$, $d_i \geq k + 1$ **then** `// Rule 2`
    if $\mathcal{N}_{i,out} \neq \emptyset$ **then**
      i picks at random a node $j \in \mathcal{N}_{i,out}$;
      if $d_j \geq k + 1$ **then**
        $\mathcal{E} \leftarrow \mathcal{E} \setminus \{(i,j)\}$;
  else if $r = r_3$, $\mathcal{N}_{i,out} \neq \emptyset$ **then** `// Rule 3`
    $\mathcal{N}_{i,out}^{\text{MAX}} := \{j \in \mathcal{N}_{i,out} : d_j = \max_{\ell \in \mathcal{N}_{i,out}} d_\ell\}$;
    i picks at random a node $j \in \mathcal{N}_{i,out}^{\text{MAX}}$;
    $d^\star = \max\{k, d_j - 1, d_i - 1\}$;
    $\mathcal{N}_{ij}^{\leq} := \{s \in \mathcal{N}_j \setminus \{\mathcal{N}_i \cup \{i\}\} : d_s \leq d^\star\}$;
    if $\mathcal{N}_{ij}^{\leq} \neq \emptyset$ **then**
      $\mathcal{N}_{ij}^{\text{MIN}} := \{s \in \mathcal{N}_{ij}^{\leq} : d_s = \min_{\ell \in \mathcal{N}_{ij}^{\leq}} d_\ell\}$;
      i picks at random a node $s \in \mathcal{N}_{ij}^{\text{MIN}}$;
      if $d_j \geq k + 1$ **then**
        $\mathcal{E} \leftarrow \mathcal{E} \cup \{(i,s)\} \setminus \{(i,j)\}$;
      else if $d_i \leq k$ **then**
        i picks at random a value $\beta' \in [0,1]$;
        if $\beta' > 1 - \beta$ **then**
          $\mathcal{E} \leftarrow \mathcal{E} \cup \{(i,s)\}$;
  else if $r = r_4$ **then** `// Rule 4`
    i picks at random a node $j \in \mathcal{N}_{i,out}$;
    $\mathcal{N}_{i,out} \leftarrow \mathcal{N}_{i,out} \setminus \{j\}$ and $\mathcal{N}_{j,out} \leftarrow \mathcal{N}_{j,out} \cup \{i\}$;

graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, defined as follows: for every edge $(i,j) \in \mathcal{E}$, either $(i,j) \in \mathcal{E}_d$ or $(j,i) \in \mathcal{E}_d$, indicating whether i or j owns the edge, respectively. With this notation, the set of out-neighbors of node i, denoted by $\mathcal{N}_{i,out} = \{j \in \mathcal{V} : (i,j) \in \mathcal{E}_d\}$, is precisely the subset of $\mathcal{N}_i$ for which node i owns the corresponding edges (and can therefore remove them).

At each iteration $t = 1, 2, 3, \ldots$, each node is active with probability $1 - \epsilon \in (0,1)$. It then locally modifies the network topology by executing one of the rules shown in Figure 2 (on the next page), where operations labeled A, R, and M denote the addition, removal, or movement of edges, respectively. These operations are performed by first applying Rule 1 and then randomly selecting one from Rules 2–3–4, which are explained next:

- **Rule 1**: if $d_i \leq k-1$, node i repeatedly executes operation (A1) to add edges to its 2-hop neighbors until its degree is at least k;
- **Rule 2**: if $d_i \geq k+1$, node i tries to remove an edge it owns from an out-neighbor whose degree is greater than or equal to $k+1$ by executing operation (R);
- **Rule 3**: node i either tries to move an edge by executing operation (M) to balance node degrees and randomize the edges, or tries (with probability β) to add an edge by executing operation (A2).
- **Rule 4**: node i relinquishes ownership of the edge (i,j), with $j \in \mathcal{N}_{i,\mathrm{out}}$, in favor of node j.

We observe that, unlike the CFOR-REG Protocol, all operations executed by nodes here are guaranteed to be non-conflicting. The only scenario worth mentioning is when two non-neighboring nodes i and j attempt to add the same edge (i,j). This situation does not lead to a conflict because the edge can indeed be added (albeit only once), thereby allowing both actions performed by the agents to succeed. Intuitively, these uncoordinated and conflict-free actions significantly speed up the edge mixing in the network when compared to the CFOR-REG Protocol. On the other hand, the UFA-REG Protocol may suffer from possible (even though very unlikely) disconnections and cannot ensure that the graph converges to an exact random regular graph. Nevertheless, we provide an empirical validation of the algebraic connectivity of the graphs obtained by the UFA-REG Protocol.

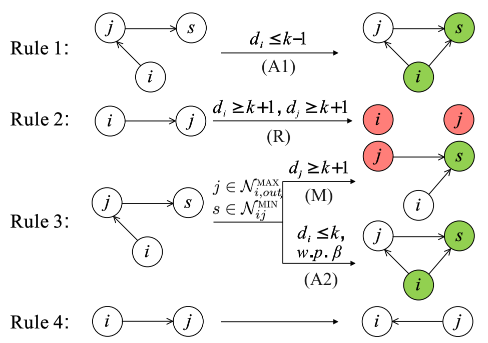

Fig. 2. Rules and operations of the UFA-REG Protocol: the degree of the red nodes decreases, while the degree of the green nodes increases.

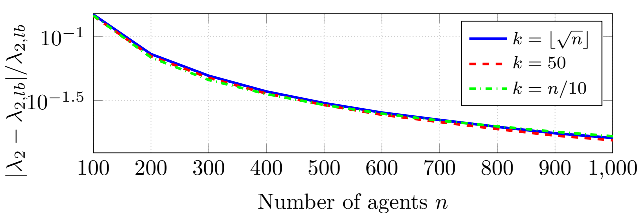

Fig. 3. Comparison of the algebraic connectivity for graphs generated by UFA-REG with the theoretical lower bound $\lambda_{2,lb}$ in Proposition 3 with $k = \{\lfloor\sqrt{n}\rfloor, 50, n/10\}$.

Theorem 8 *Consider an RGP due to the execution of the UFA-REG Protocol starting from a connected graph $\mathcal{G}(0)$ with $n \geq k$ nodes and even $k \in \{4, 6, 8, \ldots\}$. Then, there almost surely exists a time $t^\star \in \mathbb{N}$ such that the graph $\mathcal{G}(t)$ is approximate k-regular for all $t \geq t^\star$, i.e.,* $\mathrm{Prob}(\exists t^\star > 0, \forall t \geq t^\star : \mathcal{G}(t)$ *is app. k-regular*$) = 1$.

PROOF. The proof can be carried out in two steps. Firstly, the minimum degree d_{MIN} among all nodes in the network first converges within the interval $[k, \infty)$. This is because if $d_{\mathrm{MIN}} \leq k$, then d_{MIN} cannot decrease while the nodes execute the UFA-REG Protocol. If $d_{\mathrm{MIN}} < k$, then it eventually increases to at least k. The proof requires the examination of all possible cases and it is shown in detail in Lemma 10 of [42]. Secondly, once $d_{\mathrm{MIN}} \geq k$ is achieved. It can be shown—examining all possible cases—that if the maximum degree $d_{\mathrm{MAX}} \geq k+1$, then d_{MAX} cannot increase while the agents execute UFA-REG Protocol. Next, it can be shown that if $d_{\mathrm{MIN}} \geq k$ and $d_{\mathrm{MAX}} > k+1$, then d_{MAX} eventually decreases to at most $k+1$. This is shown with the necessary details in Lemma 11 of [42]. Thus, all the degrees converge to $[k, k+1]$ within a finite number of operations. □

We now empirically validate that the algebraic connectivity of a graph generated by the UFA-REG Protocol is greater in expectation than the lower bound for a random k-regular graph as characterized in Proposition 3. We run experiments on networks with an increasing number of agents where $n \in \{100, 200, \ldots, 1000\}$ and consider three different degrees of regularity $k = \{\lfloor\sqrt{n}\rfloor, 50, n/10\}$. Figure 3 shows the absolute relative distance—averaged over 1000 different instances of the problem—of the algebraic connectivity λ_2 from the lower bound $\lambda_{2,lb}$, i.e., $|\lambda_2 - \lambda_{2,lb}|/\lambda_{2,lb}$. The results confirm that such distance monotonically decreases with growing network size.

3.3 Validation of the empirical spectral distribution

We now validate the *empirical spectral distribution* (ESD) of the graphs generated by the proposed protocols by comparing it with that of random regular graphs. The eigenvalues μ_i of the adjacency matrix A_n associated with a random graph $\mathcal{G}$ are samples of independent, identically distributed random variables. In these terms, the ESD $P_{A_n} : \mathbb{R} \to [0,1]$ of the matrix A_n is an estimate of the *cumulative distribution function* $P(x) : \mathbb{R} \to [0,1]$ that generates its eigenvalues, $P_{A_n}(x) = \frac{1}{n}|\{i : \mu_i \leq x\}|$, where $|\cdot|$ denotes the cardinality of a set. In simple terms, the distribution $P(x)$ is the probability that an eigenvalue takes a value less than or equal to x, and the ESD $P_{A_n}(x)$ is an approximation of this probability given the realization A_n. Moreover, by the strong law of large numbers, the ESD $P_{A_n}(x)$ almost surely converges to $P(x)$ for $n \to \infty$. Another important concept is the relative likelihood that an eigenvalue is equal to a specific value, which is given by the *probability density function* $\rho : \mathbb{R} \to \mathbb{R}$, defined by $\lim_{n\to\infty} P_{A_n}(x) = P(x) = \int_{-\infty}^{x} \rho(x)dx$. A first important characterization of the ESD of random k-regular graph has been provided by McKay in [43], building upon [44], by considering the case of a fixed degree of regularity k in the limit of $n \to \infty$.

Proposition 9 *Let A_n be the adjacency matrix of a random k-regular connected graph with n nodes. When $n \to \infty$, the ESD of the normalized adjacency matrix $A_\sigma = A_n/\sigma$ with $\sigma = \sqrt{k-1}$ approaches the distribution with density $\rho_k(x) = \frac{k^2-k}{2\pi(k^2-kx^2+x^2)}\sqrt{4-x^2}$ for $|x| \le 2$.*

It can be verified that when also $k \to \infty$, the ESD of A_σ in Proposition 9 converges to a distribution with semi-circle density, which is commonly known as the *Wigner's semicircle law* [45].

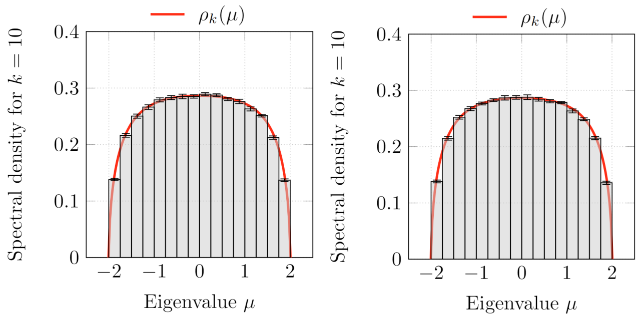

Fig. 4. ESD histogram of the normalized adjacency matrix of graphs with 1000 nodes generated by (left) CFOR-REG and (right) UFA-REG : the curve represents the density ρ_k expected for large regular graphs $n \to \infty$ (see Proposition 9).

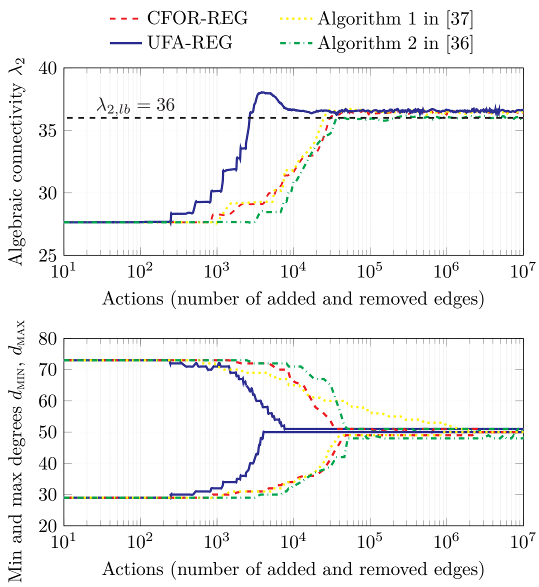

Fig. 5. Comparison with the state-of-the-art of the evolution of (top) the algebraic connectivity and (bottom) the maximum/minimum degrees, against the number of added and removed edges.

We run Monte Carlo simulations in large networks of $n = 1000$ nodes and $k = 10$, for which the expected spectral distribution is given in Proposition 9. The comparison is carried out by computing the eigenvalues of the normalized adjacency matrix A_σ and comparing their empirical density distribution with the distributions characterized in Proposition 9 for random k-regular graphs. Figure 4 shows the results averaged over 10 different instances. It can be noticed that for both protocols, the limiting distribution approaches the distribution of a random k-regular graph according to Proposition 9.

4 Numerical simulations

We now compare the performance of the proposed protocols with two other algorithms in the state-of-the-art: Algorithm 1 in [37] which allows one to choose the degree of regularity k arbitrarily but does not guarantee network connectivity; and Algorithm 2 in [36] which guarantees network connectivity but does not allow an arbitrary choice of the regularity. Instead, it is constrained k within an interval defined by the initial average degree, namely $k \in [d_{\text{AVG}}(\mathcal{G}), d_{\text{AVG}}(\mathcal{G}) + 2]$.

4.1 Performance comparison with the state-of-the-art

To make a fair comparison with Algorithm 2 in [36], we consider a network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $n = |\mathcal{V}| = 1000$ agents, whose communication graph has an initial average degree of $d_{\text{AVG}}(\mathcal{G}) = 49.2$. Then, we set the desired degree of regularity to $k = 50$, a high nodes' activation probability $1 - \varepsilon = 0.99$, and a low probability $\beta = 0.01$ of adding edges when not strictly needed. In this way, the expected graphs generated are as follows:

- the CFOR-REG Protocol, which produces a random k-regular graph with $k = 50$;
- the UFA-REG Protocol, which produces an approximate k-regular graph with $k = 50$;
- Algorithm 1 in [37], which produces an approximate random k-regular graph with $k = 50$;
- Algorithm 2 in [36], which produces a random k-regular graph with $k \in \{50, 51\}$.

Figure 5 shows the evolution of the algebraic connectivity λ_2 (top) and the minimum d_{MIN} and maximum d_{MAX} degrees (bottom) of the network while executing the proposed protocols and the algorithms from the state-of-the-art. In order to make the comparison fair, we plot these evolutions against the number of added and removed edges, which we call "*actions*": adding or removing an edge counts as one action, moving an edge counts as two actions, and exchanging two neighbors counts as four actions.

The numerical simulations reveal that the graphs generated by all protocols and algorithms increase their algebraic connectivity λ_2, approaching and then exceeding the lower bound $\lambda_{2,lb} := k - 2\sqrt{k-1} = 36$ for random k-regular graphs with $k = 50$ (see Proposition 3).

It can be noticed that the UFA-REG Protocol is the fastest one, allowing the network to achieve the desired algebraic connectivity by adding or removing fewer than $3 \cdot 10^3$ edges, while the other algorithms required around $3 \sim 5 \cdot 10^4$ actions, which is one extra order of magnitude. Moreover, the proposed protocols are also the fastest to achieve an (approximate) k-regular graph, requiring fewer than $5 \cdot 10^4$ actions, while Algorithm 1 in [37] needs about 10^6 actions, and Algorithm 2 in [36] did not converge to a regular graph even with 10^7 actions.

These results not only showcase the superior computational efficiency of the developed protocols but also suggest potential benefits for energy consumption. Since each action—whether adding, removing, moving, or exchanging an edge—carries an associated energy cost, the significant reduction in the number of operations required by our approach implies a lower overall energy demand. In scenarios where the energy cost per action can be estimated, the reported methodology offers a promising route for energy-efficient network self-organization in resource-constrained distributed systems.

4.2 Simulation in open networks

We now test the proposed protocols when the network is an "open" network, wherein agents are allowed to leave and join the network during the execution of the protocols [4–7].

Figure 6 shows the algebraic connectivity of an open network with initially $n = 1000$ agents and degree regularity parameter $k = 50$, obtained by executing the protocols proposed in this manuscript, the state-of-the-art algorithms from [36, 37], and a scenario with no self-organization. At each iteration $t > 0$, we assume there is a uniform probability $p(t) \in (0, 1)$ for the join or leave events, which increases over time according to:

$$p(t) = \begin{cases} 1.25\% & \text{if } t \leq 2.5 \cdot 10^4, \\ 2.50\% & \text{if } t \in (2.5 \cdot 10^4, 5.0 \cdot 10^4], \\ 5.00\% & \text{if } t \in (5.0 \cdot 10^4, 7.5 \cdot 10^4], \\ 10.0\% & \text{if } t \in (7.5 \cdot 10^4, 1.0 \cdot 10^5], \\ 20.0\% & \text{if } t > 1.0 \cdot 10^5. \end{cases} \tag{1}$$

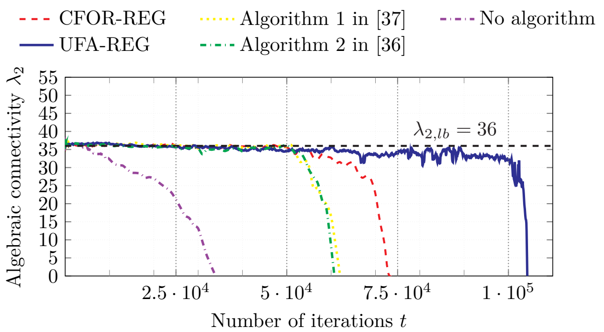

Fig. 6. Evolution of the algebraic connectivity λ_2 for an open network whose initial topology is a 50-regular graph with 1000 nodes.

In order to simulate a worst-case scenario, we let the leaving nodes to be those that are most likely to reduce network connectivity. To do so, we chose them by computing the Fiedler eigenvector of the Laplacian matrix. The sign of each entry in the Fiedler eigenvector indicates the side of the partition to which the corresponding node belongs when the graph is divided into two parts based on this eigenvector. Thus, the nodes that have the highest number of neighbors with an opposite Fiedler eigenvector sign entry are typically at the network's "center" and their removal may split the network with the highest probability. Instead, joining nodes will directly establish connections with $k = 50$ agents selected at random.

Figure 6 reveals that the UFA-REG is the best performing protocol, maintaining a high algebraic connectivity over most of the process even when the probability of join or leave events rises up to 10%, while all other algorithms fail at 5%. Among the others, the CFOR-REG protocol is the best performing, as it allows the network to remain connected for nearly 7.25×10^4 iterations. Algorithm 1 in [37] and Algorithm 2 in [36] postpone disconnection only until 6×10^4 iterations. On the other hand, without self-organization, the algebraic connectivity quickly decreases and the network disconnects with a probability of only 2.5%.

5 Conclusions

This manuscript presents two distributed protocols that enable the self-organization of any network topology into one with desired algebraic connectivity and bounded degrees, despite nodes leaving the network during the protocol execution. The strategy employed by the protocols is that of steering the graph topology toward an (approximate) k-regular graph, where the regularity degree $k \in \mathbb{N}$ is a free design parameter and the expected algebraic connectivity is at least $k - 2\sqrt{k-1}$.

References

[1] M. Zhu and G. Xie, "Model and simulation of krause model in dynamic open network," in *AIP Conference Proceedings*, AIP Publishing, vol. 1864, 2017.

[2] H. W. Abrahamson and E. Wei, "Primetime: A finite-time consensus protocol for open networks," in *62nd IEEE Conference on Decision and Control*, IEEE, 2023, pp. 5014–5019.

[3] M. Franceschelli and P. Frasca, "Proportional dynamic consensus in open multi-agent systems," vol. 2018-December, 2018, pp. 900–905.

[4] M. Franceschelli and P. Frasca, "Stability of open multiagent systems and applications to dynamic consensus," *IEEE Transactions on Automatic Control*, vol. 66, no. 5, pp. 2326–2331, 2020.

[5] D. Deplano, M. Franceschelli, and A. Giua, "Stability of paracontractive open multi-agent systems," in *63rd IEEE Conference on Decision and Control*, 2024.

[6] J. M. Hendrickx and S. Martin, "Open multi-agent systems: Gossiping with deterministic arrivals and departures," in *2016 54th annual allerton conference on communication, control, and computing (Allerton)*, IEEE, 2016, pp. 1094–1101.

[7] R. Vizuete, C. Monnoyer de Galland, P. Frasca, E. Panteley, and J. M. Hendrickx, “Trends and questions in open multi-agent systems,” in *Hybrid and Networked Dynamical Systems: Modeling, Analysis and Control*, Springer, 2024, pp. 219–252.

[8] Y. Fazea, Z. S. Attarbash, F. Mohammed, and I. Abdullahi, “Review on unstructured peer-to-peer overlay network applications,” in *2021 International Conference of Technology, Science and Administration (ICTSA)*, 2021, pp. 1–7.

[9] R. Patel, P. Frasca, J. W. Durham, R. Carli, and F. Bullo, “Dynamic partitioning and coverage control with asynchronous one-to-base-station communication,” *IEEE transactions on control of network systems*, vol. 3, no. 1, pp. 24–33, 2015.

[10] E. Restrepo, A. Lorìa, I. Sarras, and J. Marzat, “Consensus of open multi-agent systems over dynamic undirected graphs with preserved connectivity and collision avoidance,” in *61st IEEE Conference on Decision and Control*, IEEE, 2022, pp. 4609–4614.

[11] E. Restrepo and P. R. Giordano, “A distributed strategy for generalized biconnectivity maintenance in open multi-robot systems,” in *63rd IEEE Conference on decision and control (CDC 2024)*, 2024.

[12] Z. Xu, Y. Lv, and Y. Bao, “Adaptive collaborative formation control of unmanned vehicles in open environments,” in *2024 IEEE International Conference on Unmanned Systems (ICUS)*, IEEE, 2024, pp. 1023–1028.

[13] M. Kaheni, E. Usai, and M. Franceschelli, “Resilient and privacy-preserving multi-agent optimization and control of a network of battery energy storage systems under attack,” *IEEE Transactions on Automation Science and Engineering*, vol. 21, no. 4, pp. 5320–5332, 2024.

[14] M. Franceschelli, A. Pilloni, and A. Gasparri, “Multi-agent coordination of thermostatically controlled loads by smart power sockets for electric demand side management,” *IEEE Transactions on Control Systems Technology*, vol. 29, no. 2, pp. 731–743, 2021.

[15] A. Sharma and S. Chauhan, “A distributed reinforcement learning based sensor node scheduling algorithm for coverage and connectivity maintenance in wireless sensor network,” *Wireless Networks*, vol. 26, no. 6, pp. 4411–4429, 2020.

[16] N. Meena and B. Singh, “An efficient coverage and connectivity maintenance using optimal adaptive learning in wsns,” *International Journal of Information Technology*, vol. 15, no. 8, pp. 4491–4504, 2023.

[17] N. Bastianello, D. Deplano, M. Franceschelli, and K. H. Johansson, “Robust online learning over networks,” *IEEE Transactions on Automatic Control*, vol. 70, no. 2, pp. 933–946, 2025.

[18] D. Deplano, N. Bastianello, M. Franceschelli, and K. H. Johansson, “Optimization and learning in open multi-agent systems,” *arXiv preprint arXiv:2501.16847*, 2025.

[19] V. S. Varma, I.-C. Morărescu, and D. Nešić, “Open multi-agent systems with discrete states and stochastic interactions,” *IEEE Control Systems Letters*, vol. 2, no. 3, pp. 375–380, 2018.

[20] M. Xue, Y. Tang, W. Ren, and F. Qian, “Stability of multi-dimensional switched systems with an application to open multi-agent systems,” *Automatica*, vol. 146, p. 110 644, 2022.

[21] M. Fiedler, “Algebraic connectivity of graphs,” *Czechoslovak mathematical journal*, vol. 23, no. 2, pp. 298–305, 1973.

[22] 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, pp. 659–664, 2020.

[23] D. Klein and M. Randić, “Resistance distance,” *Journal of Mathematical Chemistry*, vol. 12, no. 1, pp. 81–95, 1993.

[24] Z. Bo and N. Trinajstić, “On resistance-distance and kirchhoff index,” *Journal of Mathematical Chemistry*, vol. 46, pp. 283–289, 2009.

[25] E. Lovisari, F. Garin, and S. Zampieri, “Resistance-based performance analysis of the consensus algorithm over geometric graphs,” *SIAM Journal on Control and Optimization*, vol. 51, no. 5, pp. 3918–3945, 2013.

[26] M. Pinsker, “On the complexity of a concentrator,” in *7th International Telegraffic Conference*, vol. 4, 1973, pp. 1–318.

[27] 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, pp. 593–612, 2005.

[28] C. Schneider, A. Moreira, J. A. Jr, S. Havlin, and H. Herrmann, “Mitigation of malicious attacks on networks,” *Proc. Natl. Acad. Sci.*, vol. 108, no. 10, pp. 3838–3841, 2011.

[29] M. Capalbo, O. Reingold, S. Vadhan, and A. Wigderson, “Randomness conductors and constant-degree lossless expanders,” in *Proceedings of the 34th annual ACM symposium on Theory of computing*, 2002, pp. 659–668.

[30] S. T. Arzo, R. Bassoli, F. Granelli, and F. H. Fitzek, “Multi-agent based autonomic network management architecture,” *IEEE Transactions on Network and Service Management*, vol. 18, no. 3, pp. 3595–3618, 2021.

[31] Y. Feng, F. Wang, Z. Liu, and Z. Chen, “An online topology control method for multi-agent systems: Distributed topology reconfiguration,” *IEEE Transactions on Network Science and Engineering*, 2024.

[32] L. Blume, D. Easley, J. Kleinberg, R. Kleinberg, and É. Tardos, “Which networks are least susceptible to cascading failures?” In *52nd IEEE annual symposium on foundations of computer science*, 2011, pp. 393–402.

[33] N. Alon, “Eigenvalues and expanders,” *Combinatorica*, vol. 6, no. 2, pp. 83–96, 1986.

[34] J. Friedman, “A proof of Alon’s second eigenvalue conjecture,” in *Proceedings of the 35th annual ACM symposium on Theory of computing*, 2003, pp. 720–724.

[35] A. Yazıcıoğlu, M. Egerstedt, and J. Shamma, “Decentralized formation of random regular graphs for robust multi-agent networks,” in *53rd IEEE Conference on Decision and Control*, 2014, pp. 595–600.

[36] A. Yazıcıoğlu, M. Egerstedt, and J. Shamma, “Formation of robust multi-agent networks through self-organizing random regular graphs,” *IEEE Transactions on Network Science and Engineering*, vol. 2, no. 4, pp. 139–151, 2015.

[37] Z. Dashti, D. Deplano, C. Seatzu, and M. Franceschelli, “Resilient self-organizing networks in multi-agent systems via approximate random k-regular graphs,” in *61st IEEE Conference on Decision and Control*, 2022, pp. 6448–6453.

[38] W. Zhao, D. Deplano, Z. Li, A. Giua, and M. Franceschelli, “Resilient networks for multi-agent systems based on graph self-organization into random approximate regular graphs,” in *20th IEEE International Conference on Automation Science and Engineering*, 2024, pp. 2975–2981.

[39] R. Motwani and P. Raghavan, “Randomized algorithms,” *ACM Computing Surveys (CSUR)*, vol. 28, no. 1, pp. 33–37, 1996.

[40] J. R. Norris, *Markov chains*. Cambridge university press, 1998.

[41] P. Mahlmann and C. Schindelhauer, “Peer-to-peer networks based on random transformations of connected regular undirected graphs,” in *Proceedings of the seventeenth annual ACM symposium on Parallelism in algorithms and architectures*, 2005, pp. 155–164.

[42] W. Zhao, D. Deplano, Z. Li, A. Giua, and M. Franceschelli, “Algebraic connectivity control and maintenance in multi-agent networks under attack,” *arXiv preprint arXiv:2406.18467*, 2024.

[43] B. McKay, “The expected eigenvalue distribution of a large regular graph,” *Linear Algebra and its applications*, vol. 40, pp. 203–216, 1981.

[44] H. Kesten, “Symmetric random walks on groups,” *Transactions of the American Mathematical Society*, vol. 92, no. 2, pp. 336–354, 1959.

[45] E. Wigner, “On the distribution of the roots of certain symmetric matrices,” *Annals of Mathematics*, vol. 67, no. 2, pp. 325–327, 1958.