(Article begins on next page)

A Complete Set of Connectivity-aware Local Topology Manipulation Operations for Robot Swarms

Koresh Khateri*[1], Karthik Soma*, Mahdi Pourgholi[2], Mohsen Montazeri,
Lorenzo Sabattini[3] and Giovanni Beltrame[1]

Abstract— The topology of a robotic swarm affects the convergence speed of consensus and the mobility of the robots. In this paper, we prove the existence of a complete set of local topology manipulation operations that allow the transformation of a swarm topology. The set is complete in the sense that any other possible set of manipulation operations can be performed by a sequence of operations from our set. The operations are local as they depend only on the first and second hop neighbors' information to transform any initial spanning tree of the network's graph to any other connected tree with the same number of nodes. The flexibility provided by our method is similar to global methods that require full knowledge of the swarm network. We prove the existence of a sequence of transformations for any tree-to-tree transformation, and derive sequences of operations to form a line or star from any initial spanning tree. Our work provides a theoretical and practical framework for topological control of a swarm, establishing global properties using only local information.

I. INTRODUCTION

According to Cayley's formula [1], there are N^{N-2} possible labeled spanning trees for a swarm network with N robots. Each of these determines different properties (e.g., coverage area, consensus rate, and mobility) of the swarm. In this paper, we investigate the possibility of transforming any initial spanning tree of the swarm's network, by only using local operations, to any other connected tree spanning all nodes of the system. This transformation is of importance because a fixed spanning tree restricts the relative movement of a swarm.

In a fixed topology, the mobility of the robots is constrained since initial neighbor robots have to remain neighbors throughout the mission even though they might be required at different locations that are farther than the communication range of the robots, whereas they could break their connection and still be connected with a multi-hop path.

The existence of a communication path between any pair of nodes in a robotic network defines the connectivity of that network. Continuous connectivity (i.e., strict connectivity from the start of the mission to the end) between the robots of a swarm is essential in several applications [2], [3] Assuming that communication is limited by distance, the existence of a communication link between two robots depends on their

[1] École Polytechnique de Montréal, 2900 Boul Édouard-Montpetit, Québec, CA (email: {koresh.khateri, karthik.soma, giovanni.beltrame}@polymtl.ca)
[2] Shahid Beheshti University, Tehran, Iran
[3] University of Modena and Reggio Emilia, Reggio Emilia, Italy
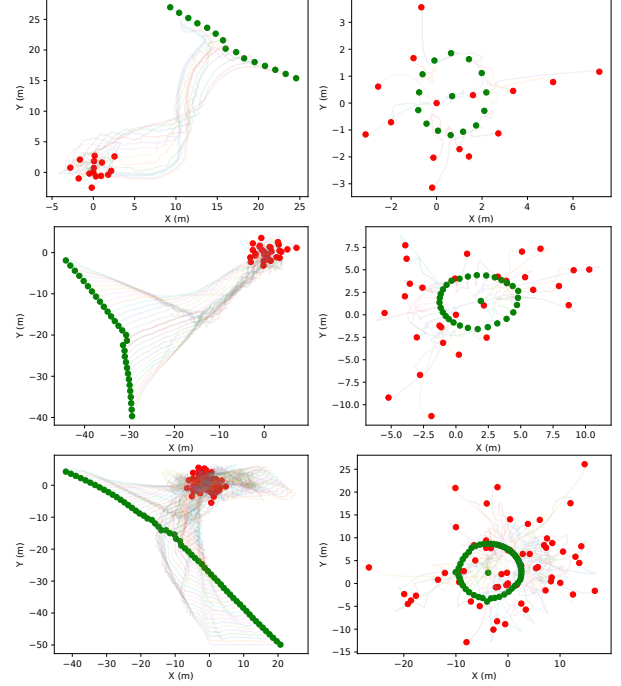*These authors contributed equally to this work

Fig. 1: Trajectory visualizations for line and star formations for 15, 30, and 60 robots, column-wise. Initial positions and final positions are represented by red and green colors, respectively.

position. Therefore, robot mobility causes the topology to dynamically change, potentially breaking links.

Connectivity maintenance is a supervisory control that blocks disconnecting action from a primary controller that pursues the primary goal of the swarm. In [4] a control barrier function based controller minimally tweaks the primary controller's command such that connectivity is guaranteed. We propose a set of operations to transform a spanning tree topology into any other spanning tree topology using only local, connectivity-aware operations which require data from immediate and 2-hop neighbors. Then the primary controller can use these operations to have more flexibility when topology manipulation is required. Note that data is not propagated from farther neighbors using decentralized averaging techniques like [5]. Although similar local manipulation operations have been proposed in the literature [6], [7], [8], [9], this is the first local complete set without using any global index of connectivity, to the best of the authors' knowledge.

II. RELATED WORK

The communication topology of a swarm can be fixed or vary throughout its mission. When enforcing a fixed topology, also known as local connectivity maintenance, every initially existing link between robots is maintained (e.g. by the use of virtual potential functions or control barrier functions) by enforcing an inter-robot distance that is smaller than the robots' communication range [10]. Local connectivity maintenance is a simple approach that requires minimal communication overhead and velocity constraints, but it restricts the relative mobility of two initially neighboring robots and makes it impossible to change the global topological properties of the swarm (e.g. consensus rate or coverage). With topology manipulation, the connectivity of the network must be taken into account either continuously from start to end [11], [12], [13] or it has to be re-established intermittently [14], [15].

Preserving connectivity has been an active research area, this subject is relevant in achieving almost all cooperative tasks in multi-robot and multi-agent systems such as flocking [16], [17], search and rescue [18], [19] and formation control [20], [21].

Published works on continuous connectivity maintenance can be classified into three main approaches:

- Local connectivity maintenance [22], [23], [24], which designs the control plans to avoid any disconnection of initially existing communication links;
- Global connectivity maintenance [25], [26], [11], [4], where the breaking of communication links is allowed, as long as the overall communication topology remains connected. Global connectivity maintenance is more flexible: it allows reconfiguring the network's graph to any possible connected graph with the same number of vertices. The criterion for allowing an edge disconnection is estimated or calculated based on a global index which requires complete information over the graph. In the decentralized version of this approach interactions are local, such that iterative distributed local averaging or consecutive local broadcast is used to collect information in order to estimate this global index.
- Connectivity-aware local topology manipulation operations: these methods are enhancing the local connectivity maintenance method by performing local topology manipulations while maintaining connectivity. A node permutation method to locally swap neighbors is proposed in [8]. A chained swarm relay method to provide maximum coverage from a ground station is investigated in [7]. A logical tree acts as a communication backbone in [9] where two topology manipulation operations called outward and inward are utilized to add robots to this backbone tree until they reach predefined targets. However, these methods do not provide the same flexibility compared to global connectivity maintenance methods.

From the literature on theoretical distributed dynamic networks, an approach [27] similar to our star formation algorithm is proposed for reconfigurable robots. In [28] the authors propose the use of rotation and slide operations which are similar to our to be defined leafization and leaf transfer operations for reconfigurable robots arranged in 2-dimensional grids. They have shown these two operations are universal in a centralized mode as the operations can make changes to the topology in the x and y axes respectively. They also propose a line formation algorithm that unlike our method requires global information.

In this paper, we investigate the possibility of defining connectivity-aware local topology manipulation operations in such a way as to provide the same flexibility as with global methods. We demonstrate that there is a complete set of operations that can manipulate the swarm's topology in any possible connected form.

The main reason for developing connectivity-aware local topology manipulation operations is the fact that the decentralized estimation of global connectivity indices is not scalable and it is very sensitive to noise [7], [10]. Despite the flexibility that a global method provides, the convergence time needed to estimate a global index restricts the speed of robots [10]. This phenomenon is caused by practical constraints like finite bandwidth and/or communication delays. The superiority of global methods in some works [29], [11] is due to assuming continuous, infinite bandwidth and cost-less communication. It is worth noting that in a global method communications are on $(N-1)-$hop path, in the worst case. Thus, it can have a significant impact on a large-scale network. However, in a local method communication paths are one or two hops. Therefore, the communication delay is insignificant and can be ignored. In a real-world situation, there is a trade-off between graph flexibility and the maximum allowed speed of robots.

The algebraic connectivity which is the second smallest eigenvalue λ_2 of the Laplacian matrix L, associated with the communication graph, is a connectivity index that is employed to indicate the connectivity of the network. λ_2 is shown to be a concave function of the Laplacian matrix [30]. Therefore, optimization approaches to calculating the control inputs had been proposed in order to maximize it [31], [32]. However, calculating the eigenvalues of a matrix requires full knowledge of the matrix. A centralized approach to calculate eigenvalues and eigenvectors is used in [32]. Power iteration is used in designing a decentralized algorithm to estimate algebraic connectivity in [33]. Based on this decentralized estimator, several global connectivity maintenance methods to apply control inputs such that λ_2 is preserved positive are proposed [11], [29], [34]. A connectivity maintenance supervisor considering time delays using control barrier functions to preserve λ_2 more than a threshold is proposed in [35].

Another approach to global connectivity maintenance is by designing a controller such that only links with an alternative $k-$hop path are allowed to disconnect. The length of a path in a graph with N nodes is less than $N-1$. Thus, global connectivity of the network could be maintained with $k =$

$N-1$ [36] and $k = 1$ corresponds to local connectivity. In [37] a centralized controller is proposed to maintain alternative paths of different sizes. Based on this view, authors in [38] have proposed a decentralized version to preserve k-hop connectivity that uses a modified dynamic source routing method to find alternative paths with a predetermined length. Thus, the locality level of the search for an alternative path is selected by the designer.

III. PRELIMINARIES

A limited connectivity range swarm of N robots is modeled with a distance-dependent graph. Let $G(t) = (V_G, E_G(t))$ be a graph with the set of vertices $V_G = \{v_1, \cdots, v_N\}$ and the edges $E_G(t) = \{(i, j) \in V_G \times V_G \mid A^G_{i,j}(t) > 0\}$ where $A^G(t)$ is the adjacency matrix. In this paper, $A^G_{i,j}(t) = 1$, if there is an edge (i, j) at time t. Otherwise, $A^G_{i,j}(t) = 0$. A graph G is considered as limited range if $A_{i,j} = 0$ when $\|x_i - x_j\| > R$, where x_i, x_j and R are the position of nodes i, j, and the communication range, respectively. The neighbors of node v_i at time t are $N^G_i(t) = \{v_j \in V_G \mid A^G_{i,j}(t) > 0\}$.

As in [39], the degree matrix associated with a graph G is defined as:

$$D^G_{i,j}(t) = \begin{cases} \sum_{k \in N_i} A^G_{i,k}(t) & i = j \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

and the Laplacian matrix is $L^G(t) = D^G(t) - A^G(t)$. The second smallest eigenvalue of the Laplacian matrix λ_2 is an index of the graph connectivity. The degree of a node degree(node) is defined as the number of edges connected to that node.

Vert(G) is a function that returns the vertices of a graph.

A tree $T(t) = (V_T, E_T(t))$ is a graph in which any two vertices are connected by exactly one path. A spanning tree of a connected graph G is a subgraph of G, which is a tree and contains all vertices of G. Let "\", "|.|" be the set complement operator and the size of a set, respectively. A leaf of a tree T is a node with only one edge connected to it. A super-leaf is a sub-tree S_j with j as its root where, $A_{i \in \{V_{S_j} \setminus j\}, k \in \{V_T \setminus V_{S_j}\}} = 0$ and $|N_j \setminus V_{S_j}| = 1$. Which means nodes in $V_{S_j} \setminus j$ do not have any edge with nodes in $V_T \setminus V_{S_j}$ and node j is connected to only one of the nodes in $V_T \setminus V_{S_j}$.

The following statements are equivalent for an undirected graph:

(a) The graph is connected.
(b) There is at least one path between each pair of nodes in the graph.
(c) $\lambda_2 > 0$.
(d) The graph has at least one spanning tree.

IV. MAIN RESULTS

In this section, we only deal with the existence of a complete set of local operations. The implementation of such operations is shown for the case of line formation and star formation in the supplementary material [40]. We define our set of operations that are able to manipulate connected trees

to transform them into any other possible connected tree of the same size. Through the unique Prüfer bijection of trees to strings, we show the completeness of our set of operations. A Prüfer sequence is a bijection between the set of labeled connected trees on N vertices with the set of sequences of length $N - 2$. The manipulation operations are provided to transform the Prüfer sequence of the tree to any other Prüfer sequence of length $N - 2$. Lemma 4.3 proves the possibility of transforming a spanning tree into any other spanning tree of the same size. Finally, Theorem 4.1 shows equality, with respect to the freedom of movement of robots, of the proposed method with the traditional global connectivity maintenance methods.

Consider labeling nodes of a tree with natural numbers $i \in \mathbb{N}$, assume the leaf with the lowest number to be l_1 and let the node connected to l_1 be node a, note that because l_1 is a leaf, a is the unique node connected to l_1, then consider that after deleting l_1, l_2 is the leaf with the lowest label number, which is connected to node b. By repeating this process until there remain only two nodes, a sequence of labels of the connected nodes to the lowest leaves could be created, which is called the Prüfer sequence of the tree (i.e. $\{a, b, \cdots\}$ here).

Lemma 4.1: [41] Every labeled tree on N nodes has one and only one Prüfer sequence of length $N - 2$.

In this paper, we have defined these four topology manipulation operations for changing the topology of a tree as follows:

Definition 4.1: Leafization $L(j, k)$: this operation is to transform a non-leaf node into a leaf node. Consider node j to be a non-leaf node with a neighbor $k \in N_j(t)$ and $|N_j(t)| > 1$. With the leafization operation, all the edges $\{(j, p \in \{N_j \setminus k\})\}$ will be disconnected and new edges $\{(k, p \in \{N_j \setminus k\})\}$ will be established.

Definition 4.2: Leaf transfer $LT(l, j, k)$: This operation is to transfer a leaf from one node to another node. Consider node l as a leaf connected to node j while node j is a neighbor of node k. Leaf transfer of leaf l from node j to node k is to disconnect the edge (l, j) and establish a new link (l, k).

Definition 4.3: Super leafization $SL(S_j, k)$: Consider S_j to be a sub-tree rooted in j (i.e. $A_{i \in \{Vert(S_j) \setminus j\}, k \in \{Vert(T) \setminus Vert(S_j)\}} = 0$, where $A_{i,k}$ means the i, kth element of the adjacency matrix). Consider node k such that $k \in \{N_j \setminus Vert(S_j)\}$ and $|N_j \setminus Vert(S_j)| > 1$. With the super-leafization operation, all the edges $\{(j, p \in N_j \setminus \{k \cup Vert(S_j)\})\}$ will be disconnected and new edges $\{(k, p \in N_j \setminus \{k \cup Vert(S_j)\})\}$ will be established.

Definition 4.4: Super leaf transfer $SLT(S_l, j, k)$: This operation is to transfer a super leaf connected to one node to another node. Consider super node S_l as a super leaf connected to node j while node j is a neighbor of node k. Super-leaf transfer of S_l from node j to node k is to disconnect the edge (l, j) and establish a new link (l, k).

Remark 4.1: The $SL(S_j, k)$ and $SLT(S_l, j, k)$ operations are generalizations of $L(j, k)$ and $LT(l, j, k)$ operations,
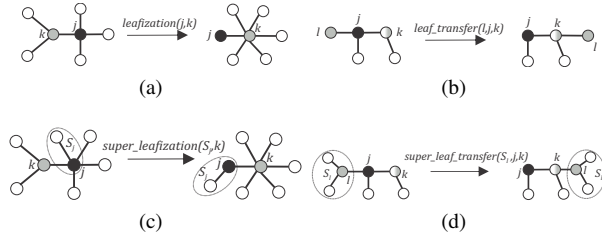
Fig. 2: (a) $L(j,k)$, (b) $LT(l,j,k)$, (c) $SL(S_j,k)$, and (d) $SLT(S_l,j,k)$ operation.

respectively. $SL(S_j,k)$ and $SLT(S_l,j,k)$ are still local operations as only the edges between the root of the super node and the neighbors are affected. All the other edges in the super nodes are left untouched.

An example of these operations is illustrated in Figure 2.

Lemma 4.2: Under the defined tree manipulation operations the result remains a tree.

Proof: Regarding the $L(j,k)$ operation: Let $N_j(0) = \{j_1, j_2, \cdots\}$ be the set of neighbors of node j before the operation. Consider a pair of nodes s and d that are connected by a path including node j, before the operation. Let the unique initial path be $\{s, \cdots, j_x, j, j_y, \cdots, d\}$ then, after the leafization operation, this pair of nodes are connected uniquely, by the path $\{s, \cdots, j_x, k, j_y, \cdots, d\}$. Because, with the operation all the neighbors of j will be connected to k as defined in the operation's definition and $\{j,k\}$ is the unique path between nodes j and k. Thus the path $\{s, \cdots, j_x, k, j_y, \cdots, d\}$ is the only path between s and d. Hence, there is a unique path between each pair of nodes after the operation. Therefore, the initial-tree remains a tree and is connected.

The proof for LT operation is trivial and the proof for SL and the SLT operation are omitted, as they are generalizations of the L and LT operations. ∎

Please note that the following lemma is only showing the existence of a series of operations for transforming any tree and the Prüfer sequence and labeling nodes are only used for proving the lemma. For implementing our method, we need to design algorithms based on our manipulation operations like what we have illustrated in supplementary material [40]

Lemma 4.3: Every tree in the set of all possible connected trees with N nodes is transformable to any other connected tree of the same size, by using the defined topology manipulation operations.

Proof: Let us call the initial tree, resulting-tree(m), and the final tree as the tree we start from, the result after m steps and the tree we want to end up with, respectively. Consider the final-tree's Prüfer sequence to be $\{p_1, p_2, \cdots, p_{N-2}\}$ and the sequence of leafs with lowest label while composing this Prüfer sequence be $\{l_1, l_2, \cdots, l_{N-2}\}$. Beginning with any arbitrary initial tree, if l_1 is not a leaf of the initial tree, it could be transformed into a leaf, by using the leafization operation. In a connected tree, there is a unique path between any pair of nodes. Thus, there is a unique path between l_1 and node p_1. Then, using a sequence of leaf transfer

operations, l_1 can move through that path to establish an edge with p_1. After this step, p_1 will be the first element of the final-tree's Prüfer sequence, as desired. At step m, if the neighbors of l_m in the resulting-tree($m-1$) (i.e., $N_{l_m}(m-1)$) contains any node in the set $\{l_1, \cdots, l_{m-1}\}$ (i.e., $N_{l_m}(m-1) \cap \{l_1, \cdots, l_{m-1}\} \neq \emptyset$) then consider the subtree, containing l_m and all the subtrees rooted in nodes $N_{l_m}(m-1) \cap \{l_1, \cdots, l_{m-1}\}$, if this subtree is not a super leaf of the resulting-tree($m-1$), it could be transformed into a super leaf, by using the super_leafization operation. Then, there is a unique path between l_m and node p_m. Thus, using a sequence of super leaf transfer operations, l_m can move through that path to establish an edge with p_l. This makes p_m the mth item in the final-tree's Prüfer sequence. As stated in Lemma 4.1, the complete set of trees on N nodes are uniquely described by Prüfer sequences. Therefore, the possibility of transforming an arbitrary initial tree into a tree with a desired Prüfer sequence by only using certain operations means the possibility of achieving any desired tree using those operations. ∎

The following theorem shows that all achievable benefits from global connectivity maintenance are possibly achievable using a modified local method, in which the initial tree is maintained by local maintenance of the initial edges of the tree, while edges are only allowed to break or be created by using the topology manipulation operations.

Theorem 4.1: Global connectivity maintenance, for distance-dependent networks, is reducible to local topology manipulation with defined tree manipulation operations.

Proof: Global connectivity maintenance needs an initially connected graph. A spanning tree of this graph could be selected as the initial tree to be maintained connected in the local topology manipulation scheme. At any instance or after reaching any desired configuration (i.e breaking edges and creating new ones), with the global connectivity maintenance, a new connected graph is connecting the nodes of the network, which also contains a spanning tree. This former spanning tree is reachable from the initial tree with our topology manipulation operations, according to Lemma 4.3, and it is connected, based on Lemma 4.2. Therefore, any configuration reachable with global connectivity maintenance is also achievable with local tree maintenance with tree manipulation operations. ∎

Remark 4.2: The tree manipulation operations need information, only from immediate (i.e. 1-hop) neighbors and 2-hop neighbors. Therefore, the operations are considered to be local.

Remark 4.3: Theorem 4.1, only shows the possibility of the reduction of global connectivity maintenance to a modified local connectivity maintenance. It does not provide a sequence of operations needed to transform an initial tree into a desired final tree. To find the sequence in which the operations should be done we need to have the target spanning tree be definable with local rules which do not require any global information. Otherwise, a one-time (i.e., not continuously every moment during the mission and only

when a change of topology is needed) propagation of the structure of the target tree is needed which can be done decentralized. This point is explained more for the case of line and star topologies which are locally definable in Section V and supplementary materials [40].

To implement the topology manipulation operations, it is needed to establish new connections with second neighbor robots. During the operations, Due to the communication range of robots, as considered to be R, the neighbor's distance must be reduced until it reaches $\frac{R}{2} - \delta$, δ being a positive small constant. After this distance reduction, robots will be able to communicate and establish new connections with their second neighbors. A method to perform such a local inter-distance reduction using gradients of potential functions is investigated in [8] for a local swap operation. We have used the same controller and the proof of connectivity preservation during the inter-distance reduction is similar.

V. EXPERIMENTS

In this section, we show how we can use the operations defined in Section IV to perform line and star formation from any initially connected random topology. The knowledge of Prüfer sequences and labeling of nodes are not used for forming lines and star formation. We choose the line and star formation because the sequences required for a line/star topology from any initial topology can be determined by local information such as the number of neighbors of each node and the roles of the neighbor robots (free to participate in an operation or busy). Unlike the Theorem 4.1 we use parallel operations to make faster lines and stars. In particular, to form lines we use leaf transfers and super leaf transfers, and to form stars we use leafizations and super leafization. For more information about the rules refer to the supplementary material [40]. For other symmetric topologies that have a local structure, similar algorithms could be developed because the possibility of the existence of such an algorithm is proven in Lemma 4.3.

we have designed these experiments in the Buzz [42] programming language with a set of Khepera IV robots using ARGoS3 [43] as our physics-based simulator. Our experiments show that we can transform any initial topology into a line or a star, with swarms of varying sizes (i.e., number of robots), namely 15, 30, and 60. The robots have a communication range R_{range}. We run each experiment for 5 randomly generated initial positions.

We require the robots to be initially connected and to be aware of their immediate and second neighbors on an initial spanning tree. Please note that this initial spanning tree could be given to all robots before they start their mission or it can be calculated in a decentralized way [9].

After this step, the robots perform local operations on the spanning tree to form line or star topologies. When the robots have to perform the transformation, the robots involved come to a reduced distance R_{transfer}. This gives the robots the ability to form connections with multiple neighbors and break links if needed. Robots maintain a distance R_{mission} while not performing the operations.

Fig. 3: Plots for the time taken to complete the tree transformation and arrangement completion, from randomly initial trees to the line and star formation, for 15, 30, and 60 robots.

The robots make use of range and bearing measurements to know the relative positions of their neighbors and use potential-based controllers similar to [22]. This allows the robots to maintain the required distances R_{transfer} and R_{mission}.

Figure 3 shows the time taken to transform the topology for line and star cases, labeled as topology transformed, and the time required to arrange the robots, labeled as arrangement finished. This latter is the physical positioning of the robots, e.g. the straightening of the line formation and the equalization of angles between the points of the star 1.

Figures 4 and 5 depict λ_2 and coverage area. We plot the time evolution of λ_2 of the tree, as a connectivity index and a parameter specifying the consensus rate. This is done to demonstrate the connectivity awareness of our method which has to stay greater than zero over the experiment.

In line formation, the value of λ_2 of the tree reduces with time and reaches a constant value when it is straightened out. Whereas in star formation, the value of λ_2 of the tree increases and reaches the same value no matter the number of nodes in the system. The coverage area has been shown to decrease for the star and to increase for the line case. This shows the trade-off between λ_2 and the coverage area. This is why topology manipulation is required to provide flexibility.

We also provide the plots for λ_2 of the graph and some progress indices for line in the form of evolution of degree(robot) = 1 and degree(robot) = 2 with time and for star in the form of evolution of degree(robot) = 1 and degree(robot) $\geq$ 2 with time in the supplementary material [40]. This is to show that the manipulation operations are changing the initial topology closer to a line and star with time.

It is worth noting the importance of choosing the two topologies line and star. For the line, the main advantage is the coverage it can provide. This is useful when the robots are searching, exploring, or they need communication with a base station. For star, the main advantages are the ability to make decisions faster (e.g., consensus). We show that from any random topology we can form the tree with maximum and minimum λ_2. Based on some environmental conditions or user requirements, the swarm can make a decentralized decision to change the topology of the swarm and consequently manipulate λ_2 (e.g., when the swarm wants to forage it can form a line as they need to maximize their coverage, and when they want to decide where to store food it can form a star as they need to make faster decisions.)
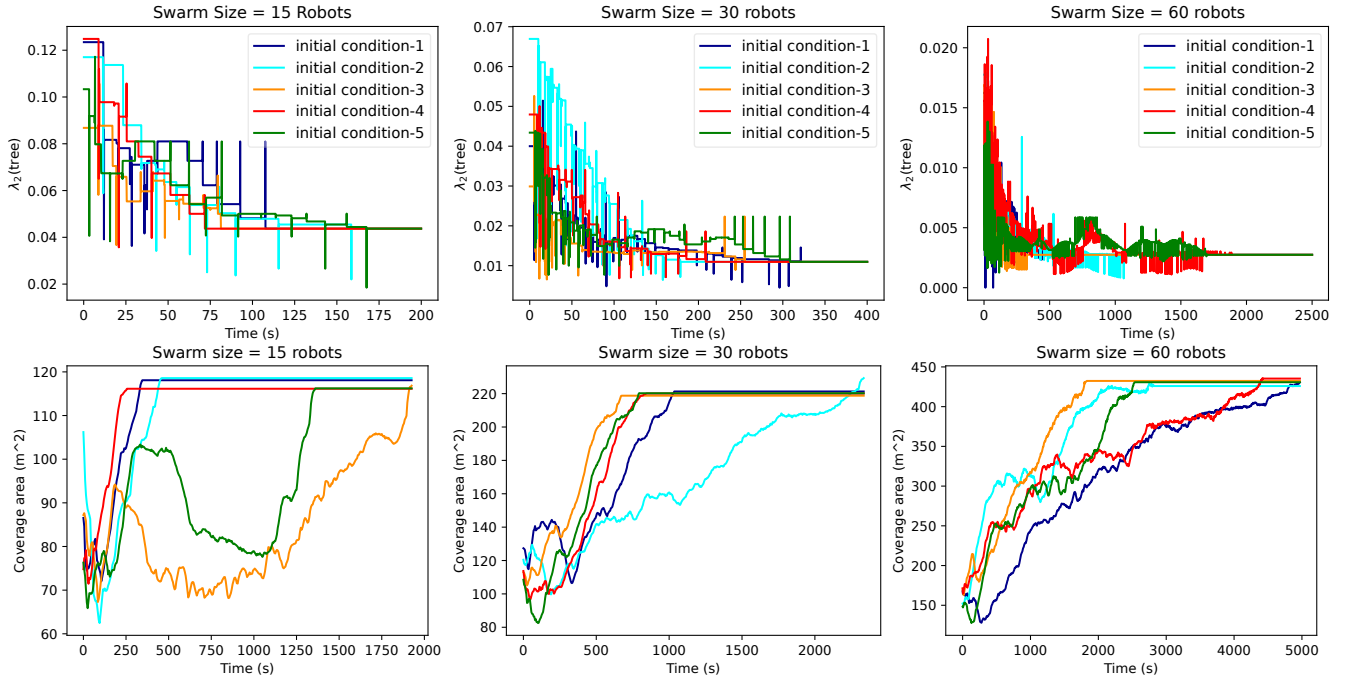
Fig. 4: Line formation for 15, 30, and 60 robots column wise. From top to bottom in each column the evolution of λ_2 of the spanning tree and coverage area over time is plotted.
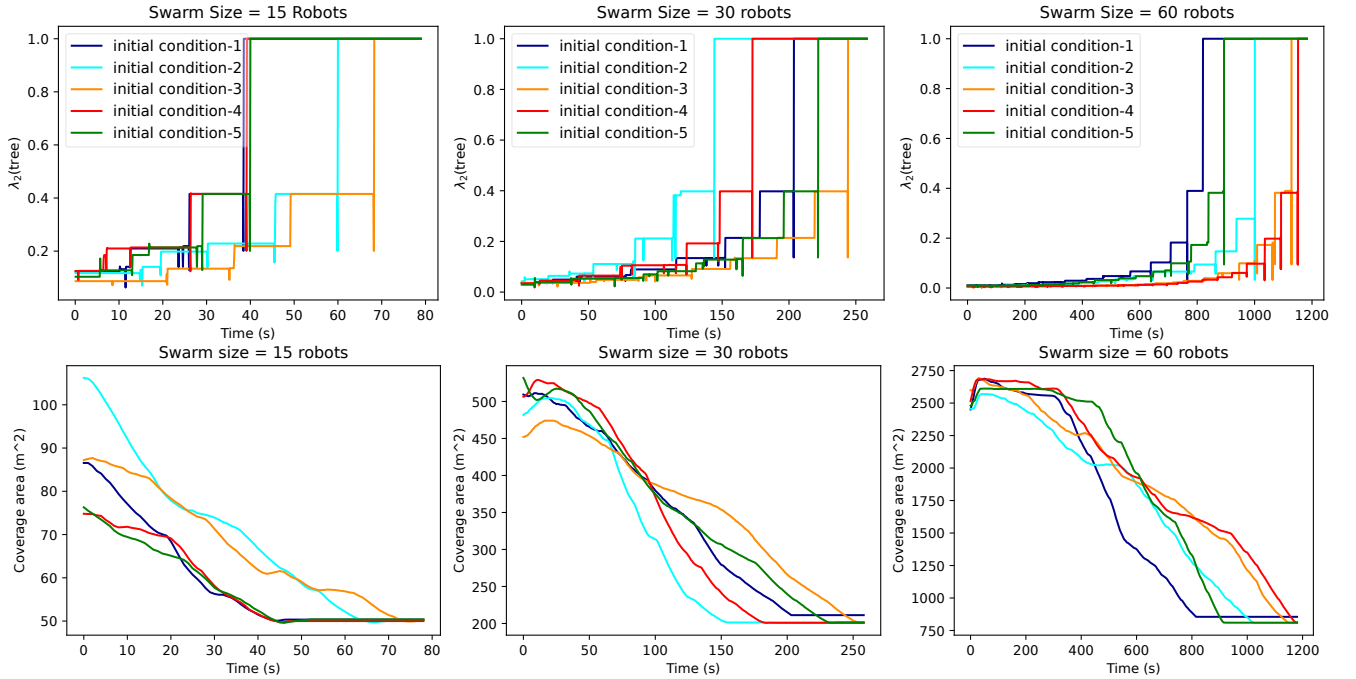

Fig. 5: Star formation for 15, 30, and 60 robots column wise. From top to bottom in each column the evolution of λ_2 of the spanning tree and coverage area over time are plotted.

VI. CONCLUSION

The possibility of introducing a complete set of connectivity-aware local topology manipulator operations have been proven to be achievable in this paper. This result has shown that estimating a global index of connectivity is not necessary for topology manipulation in its most flexible sense. This is important because estimating global connectivity indices in a decentralized fashion is sluggish and not scalable. As an application for this local method, transforming a swarm in such a way that achieves extreme contradicting properties (i.e., maximum consensus rate and coverage area) was shown to be possible. Future research in this area can consist of finding another complete set of operations with better performance on certain applications and applying this local method to other swarm missions.

REFERENCES

[1] P. W. Shor, "A new proof of cayley's formula for counting labeled trees," *Journal of Combinatorial Theory, Series A*, vol. 71, no. 1, pp. 154–158, 1995.

[2] C. Liu, Z. Zhang, and Q. Zeng, "Distributed connectivity maintenance for flying ad-hoc networks considering bridging links," *Physical Communication*, vol. 48, p. 101409, 2021.

[3] Y. Dong and J. Huang, "Consensus and flocking with connectivity preservation of uncertain euler–lagrange multi-agent systems," *Journal of Dynamic Systems, Measurement, and Control*, vol. 140, no. 9, p. 091011, 2018.

[4] B. Capelli and L. Sabattini, "Connectivity maintenance: Global and optimized approach through control barrier functions," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 5590–5596.

[5] Y. Zhang, S. Li, and J. Weng, "Distributed estimation of algebraic connectivity," *IEEE Transactions on Cybernetics*, 2020.

[6] S. Yi, W. Luo, and K. Sycara, "Distributed topology correction for flexible connectivity maintenance in multi-robot systems," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 8874–8880.

[7] V. S. Varadharajan, D. St-Onge, B. Adams, and G. Beltrame, "Swarm relays: distributed self-healing ground-and-air connectivity chains," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5347–5354, 2020.

[8] K. Khateri, M. Pourgholi, M. Montazeri, and L. Sabattini, "A connectivity preserving node permutation local method in limited range robotic networks," *Robotics and Autonomous Systems*, vol. 129, p. 103540, 2020.

[9] N. Majcherczyk, A. Jayabalan, G. Beltrame, and C. Pinciroli, "Decentralized connectivity-preserving deployment of large-scale robot swarms," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4295–4302.

[10] K. Khateri, M. Pourgholi, M. Montazeri, and L. Sabattini, "A comparison between decentralized local and global methods for connectivity maintenance of multi-robot networks," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 633–640, 2019.

[11] L. Sabattini, C. Secchi, N. Chopra, and A. Gasparri, "Distributed control of multirobot systems with global connectivity maintenance," *IEEE Transactions on Robotics*, vol. 29, no. 5, pp. 1326–1332, 2013.

[12] D. V. Dimarogonas and K. H. Johansson, "Decentralized connectivity maintenance in mobile networks with bounded inputs," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*. IEEE, 2008, pp. 1507–1512.

[13] A. Ajorlou, A. Momeni, and A. G. Aghdam, "A class of bounded distributed control strategies for connectivity preservation in multi-agent systems," *IEEE Transactions on Automatic Control*, vol. 55, no. 12, pp. 2828–2833, 2010.

[14] M. Saboia, L. Clark, V. Thangavelu, J. A. Edlund, K. Otsu, G. J. Correa, V. S. Varadharajan, A. Santamaria-Navarro, T. Touma, A. Bouman *et al.*, "Achord: Communication-aware multi-robot coordination with intermittent connectivity," *arXiv preprint arXiv:2206.02245*, 2022.

[15] Y. Kantaros and M. M. Zavlanos, "Distributed intermittent connectivity control of mobile robot networks," *IEEE Transactions on Automatic Control*, vol. 62, no. 7, pp. 3109–3121, 2017.

[16] H. Liang, H. Cao, and Y. Fu, "Decentralized adaptive flocking control algorithm with avoiding collision and preserving connectivity for crowded uuv swarm with uncertainties and input saturation," *Ocean Engineering*, vol. 237, p. 109545, 2021.

[17] Z. Feng and G. Hu, "Connectivity-preserving flocking for networked lagrange systems with time-varying actuator faults," *Automatica*, vol. 109, p. 108509, 2019.

[18] E. F. Flushing, M. Kudelski, L. M. Gambardella, and G. A. Di Caro, "Connectivity-aware planning of search and rescue missions," in *Safety, Security, and Rescue Robotics (SSRR), 2013 IEEE International Symposium on*. IEEE, 2013, pp. 1–8.

[19] S. Hayat, E. Yanmaz, T. X. Brown, and C. Bettstetter, "Multi-objective uav path planning for search and rescue," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 5569–5574.

[20] J. Fu, G. Wen, X. Yu, and Z.-G. Wu, "Distributed formation navigation of constrained second-order multiagent systems with collision avoidance and connectivity maintenance," *IEEE Transactions on Cybernetics*, 2020.

[21] Z. Peng, N. Gu, Y. Zhang, Y. Liu, D. Wang, and L. Liu, "Path-guided time-varying formation control with collision avoidance and connectivity preservation of under-actuated autonomous surface vehicles subject to unknown input gains," *Ocean Engineering*, vol. 191, p. 106501, 2019.

[22] G. Wen, Z. Duan, H. Su, G. Chen, and W. Yu, "A connectivity-preserving flocking algorithm for multi-agent dynamical systems with bounded potential function," *IET Control Theory & Applications*, vol. 6, no. 6, pp. 813–821, 2012.

[23] M. Ji and M. Egerstedt, "Distributed coordination control of multiagent systems while preserving connectedness," *IEEE Transactions on Robotics*, vol. 23, no. 4, pp. 693–703, 2007.

[24] M. A. Hsieh, A. Cowley, V. Kumar, and C. J. Taylor, "Maintaining network connectivity and performance in robot teams," *Journal of Field Robotics*, vol. 25, no. 1-2, pp. 111–131, 2008.

[25] A. Gasparri, L. Sabattini, and G. Ulivi, "Bounded control law for global connectivity maintenance in cooperative multirobot systems," *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 700–717, 2017.

[26] L. Siligardi, J. Panerati, M. Kaufmann, M. Minelli, C. Ghedini, G. Beltrame, and L. Sabattini, "Robust area coverage with connectivity maintenance," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 2202–2208.

[27] O. Michail, G. Skretas, and P. G. Spirakis, "Distributed computation and reconfiguration in actively dynamic networks," in *Proceedings of the 39th Symposium on Principles of Distributed Computing*, ser. PODC '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 448–457. [Online]. Available: https://doi.org/10.1145/3382734.3405744

[28] ——, "On the transformation capability of feasible mechanisms for programmable matter," *Journal of Computer and System Sciences*, vol. 102, pp. 18–39, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0022000018300436

[29] L. Sabattini, N. Chopra, and C. Secchi, "Decentralized connectivity maintenance for cooperative control of mobile robotic systems," *The International Journal of Robotics Research*, vol. 32, no. 12, pp. 1411–1423, 2013.

[30] R. Merris, "Laplacian matrices of graphs: a survey," *Linear algebra and its applications*, vol. 197, pp. 143–176, 1994.

[31] M. C. De Gennaro and A. Jadbabaie, "Decentralized control of connectivity for multi-agent systems," in *Decision and Control, 2006 45th IEEE Conference on*. IEEE, 2006, pp. 3628–3633.

[32] Y. Kim and M. Mesbahi, "On maximizing the second smallest eigenvalue of a state-dependent graph laplacian," in *American Control Conference, 2005. Proceedings of the 2005*. IEEE, 2005, pp. 99–103.

[33] P. Yang, R. A. Freeman, G. J. Gordon, K. M. Lynch, S. S. Srinivasa, and R. Sukthankar, "Decentralized estimation and control of graph connectivity for mobile sensor networks," *Automatica*, vol. 46, no. 2, pp. 390–396, 2010.

[34] C. Secchi, L. Sabattini, and C. Fantuzzi, "Decentralized global connectivity maintenance for interconnected lagrangian systems with communication delays," *IFAC Proceedings Volumes*, vol. 45, no. 19, pp. 78–83, 2012.

[35] B. Capelli, H. Fouad, G. Beltrame, and L. Sabattini, "Decentralized connectivity maintenance with time delays using control barrier functions," *arXiv preprint arXiv:2103.12614*, 2021.

[36] M. M. Zavlanos, M. B. Egerstedt, and G. J. Pappas, "Graph-theoretic connectivity control of mobile robot networks," *Proceedings of the IEEE*, vol. 99, no. 9, pp. 1525–1540, 2011.

[37] M. M. Zavlanos and G. J. Pappas, "Controlling connectivity of dynamic graphs," in *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*. IEEE, 2005, pp. 6388–6393.

[38] K. Khateri, M. Pourgholi, M. Montazeri, and L. Sabattini, "Decentralized local-global connectivity maintenance for networked robotic teams," *European Journal of Control*, vol. 51, pp. 110–121, 2020.

[39] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, 2010. [Online]. Available: https://doi.org/10.1515/9781400835355

[40] K. Soma, K. Khateri, M. Pourgholi, M. Montazeri, L. Sabattini, and G. Beltrame, "Supplementary material for a complete set of connectivity aware local topology manipulation operations for swarms," https://mistlab.ca/papers/2022/TopologyManipulation/.

[41] H. Prufer, "Neuer bewis eines satzes uber permutationnen," *Arch. Math. Phys.*, vol. 27, pp. 742–744, 1918.

[42] C. Pinciroli and G. Beltrame, "Buzz: An extensible programming language for heterogeneous swarm robotics," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 3794–3800.

[43] C. Pinciroli, V. Trianni, R. O'Grady, G. Pini, A. Brutschy, M. Brambilla, N. Mathews, E. Ferrante, G. Di Caro, F. Ducatelle, M. Birattari, L. M. Gambardella, and M. Dorigo, "ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems," *Swarm Intelligence*, vol. 6, no. 4, pp. 271–295, 2012.