Distributed Topology Correction for Flexible Connectivity Maintenance in Multi-Robot Systems

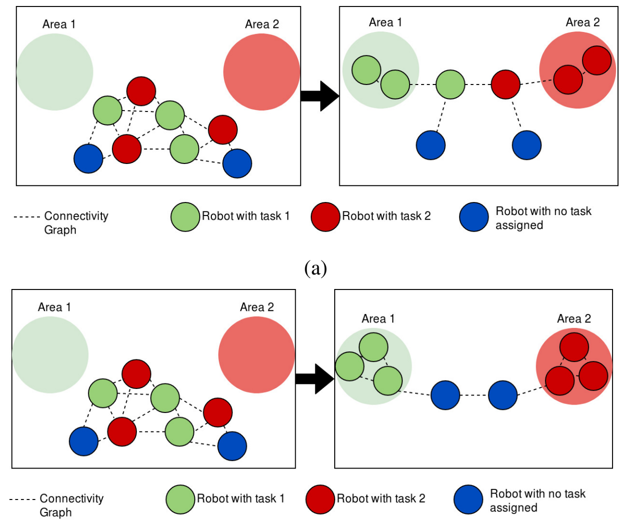
Sha Yi Wenhao Luo Katia Sycara

***Abstract*— Multi-robot systems can perform task-related collaborative behaviors while maintaining connectivity within the system. However, some robots may fail to execute tasks or converge relatively slowly due to connectivity constraints. We consider the case that some robots may not have tasks assigned at a certain time frame, and they may help the task robots to achieve their goals by forming a connectivity graph with flexible topology. Therefore, we introduce a *topology correction controller* to provide flexibility for the task robots to perform task behaviors by modifying the topology of the connectivity graph for a faster convergence rate. We propose a distributed approach of blending weighted rendezvous and weighted flocking to form the correction controller. We prove that this scheme can guarantee a faster convergence rate and provide flexible connectivity graph topology. We then present our result of a system of up to thirty robots in various cluttered environments and show that our approach of behavior combination is robust and scalable.**

I. Introduction

Multi-robot systems have been widely studied for their ability to perform collective behaviors to accomplish complicated tasks [1]–[4]. Each robot within the system usually has a limited range of communication [5] and can only exchange information with its neighbors in the connectivity graph. Maintaining connectivity of the whole system is essential [6] since collaborative behaviors rely on inter-robot connections [7], and it takes an extensive amount of work to restore the connection once it is lost [8]. In this networked robotics system, distributed algorithms, where each robot reasons and controls using only the local neighbor information, are also important for computational efficiency and scalability [7].

Most work focuses on maintaining connectivity based on existing control laws [7], [9], or studies the connectivity from a given behavior or state [10], [11]. These are based on the assumption that every robot is assigned with a set of behaviors to perform in sequence [12] or in parallel. However, in some of these situations, part of the robot system might not complete the given tasks due to connectivity constraints. An example is shown in Figure 1. The robots are given preassigned tasks, and the goal is to move to the assigned locations. However, due to connectivity constraints, some of the robots within the system might not reach the designated areas. In this case, it would be beneficial if more robots who have no assigned active task could be added to task robots' system and serve as connecting points of maintaining connectivity when there is no active task assigned to them. It would be desired if those robots could move to change the topology of the connectivity graph such that the convergence rate of other robots' task-related behaviors is also improved. Therefore, our goal is to design such a controller for those robots with no active tasks to help improve other task robots' behaviors under connectivity constraints. We define *topology correction controller* to be the controller used by these robots that aim at correcting the topology of the connectivity graph to improve the flexibility for the task robots. In real-world applications, for example, exploration in a complicated environment, search and rescue, where robots need to be equipped with expensive sensing or other special capabilities, one may explicitly define a group of robots with correction controller to serve as connecting robots for the system, where these robots can be equipped with only communication capabilities to lower the cost of the whole system. In other cases, the correction controller can be used implicitly when tasks are allocated dynamically, and some robots may serve as connecting robots to correct the topology of the connectivity graph when there is no active task assigned to them during a specific time frame.



(a)

(b)

Fig. 1: Left figures: green robots are assigned to task 1, red robots are assigned to task 2, and blue robots are not assigned any task. The tasked robots need to reach their corresponding goal areas while maintaining connected. Resulting configurations when (a) the blue robots do not actively move to support the task robots. Two task robots fail to reach the designated areas due to connectivity constraints; (b) the blue robots actively move to form connecting points so that the task robots can reach their goal locations.

The challenges of designing a correction controller involve 1) best utilizing the non-task robots in favor of the

The authors are with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA. Email: {shayi, wenhao katia}@cs.cmu.edu.

This work was funded by AFOSR award FA9550-18-1-0097 and AFRL/AFOSR award FA9550-18-1-0251.

performance of task robots; 2) formalizing the controller of non-task robots to influence the topology of connectivity graph, which determines convergence and performance; 3) speeding up the convergence rate of the controller for the whole system.

In this paper, we propose a *weighted rendezvous* algorithm where the weights are assigned based on the structure of the connectivity graph to correct the topology of the graph. We will also present its variations of *weighted behavior combination* to compensate for limitations during execution. To guarantee connectivity of the whole system, we make use of *Minimum Connectivity Constraint Spanning Tree* [13], [14] to find the optimal connectivity constraints for the system and provide the final control output based on the selected constraints. The main contribution is as follows: i) propose correction controller that gives flexible topology in favor of improving convergence rate, thus enhancing the performance of the system; ii) provide a theoretical analysis of the guaranteed improvement of the task performance and convergence rate; iii) evaluate with experimental results of the effectiveness and efficiency of our proposed topology correction controller.

II. Related Work

Multi-robot systems can accomplish collaborative tasks with predefined control laws towards the goal region such as flocking strategy [15], [16] or from a sequence of behavior library [12]. Such collaborative performance relies on communication between neighbors in the connectivity graph [7] to exchange robot states and information. Maintaining the system's connectivity is essential [6] and previous works [17], [18] have introduced methods on adding more robots to preserve or restore connectivity or enhance the robustness of connectivity for the whole system.

Connectivity maintenance in multi-robot systems with predefined control laws has been extensively studied in the literature [7], [10], [19]. Various research works [20], [21] present gradient-based methods and consider connectivity as attraction forces between robots to maintain flexible connectivity for the system as it executes predefined tasks. Barrier certificate [22], [23] proposed to achieve collision avoidance can also be used to formulate connectivity constraints [13], [14], [24], [25], e.g. [14] provides a way of selecting the optimal active connectivity constraints to minimize the influence of connectivity constraints on the original robot controller. This method is extensively used in our approach to guarantee connectivity within the whole system while minimizing the influence of connectivity constraints.

To measure the connectivity of the communication graph, [26] stated the relationship between the second smallest eigenvalue of the Laplacian matrix of the graph, further discussed in detail in [11], which also stated the relationship between the convergence rate and stability of the system. In [11], [15], it is shown that the convergence rate of rendezvous and flocking is directly related to the second smallest eigenvalue of the Laplacian matrix, which is also determined directly by the degrees of the graph vertices, as well as other topological properties. We will discuss this in detail in Section IV-B. While the influence of connectivity graph topology on the robot controller has been extensively studied, it has in general ignored the influence of the robot control output on the graph topology. Therefore, in this paper, we discuss how to modify the graph topology with control output to form a more flexible connectivity graph for the system.

III. Problem Definition

Consider a robotic team of n robots with positions denoted as $x_i \in \mathbb{R}^2$ where $i \in \{1, 2, \dots, n\}$, and single integrator dynamics of $\dot{x}_i = u_i$. This can be mapped to unicycle dynamics during real-world applications when needed [27]. Velocity limitation exists for the robots as $||u_i|| \leq u_{max}$. Each robot can communicate with all robots within a limited Euclidean distance R, i.e. robot i ids connected and can communicate with robot j if $||x_i - x_j|| \leq R$. This forms a spatially induced connectivity graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where each vertex $v_i \in \mathcal{V}$ represents robot i, and each edge $e_{ij} \in \mathcal{E}$ between v_i and v_j exists when robot i and robot j are connected as defined above. The connectivity graph $\mathcal{G}$ is undirected, i.e. $e_{ij} = e_{ji}$. Robot j is considered as a neighbor of robot i if they are connected, i.e. $e_{ij} \in \mathcal{E}$. We denote the neighbor set of robot i to be $\mathcal{N}_i \subseteq \mathcal{V}$. Additionally, the robots maintain a safety distance with each other of $d_{ij} \geq r$ where $d_{ij} = ||x_i - x_j||$, and r is the safety radius between robots to avoid collision. Similar to the inter-robot collision avoidance, the robots also maintain safety distance with the environment obstacles.

Suppose some robots in the multi-robot system are assigned tasks and need to perform specific behaviors. We denote the set of task robots to be $\mathcal{V}_t \subset \mathcal{V}$. Accordingly, we denote the set of robots that have no task behavior assigned to be $\mathcal{V}_c \subset \mathcal{V}$ with $\mathcal{V}_t \cup \mathcal{V}_c = \mathcal{V}$. We assume the task allocation for the task robots has been done beforehand, and we do not consider how these are designed and allocated. The performance is measured based on the completion of tasks for $\mathcal{V}_c$. Note that in this paper, we assume that the number of robots with tasks is always less than the total number of robots within the system, i.e., $|\mathcal{V}_c| \geq 1$. Due to the connectivity constraints, some of the task robots might not reach the designated location, as shown in the example in Figure 1a. In this paper, our objective is to design the controller for robots with no tasks in set $\mathcal{V}_c$ to provide flexibility for the task robots in $\mathcal{V}_t$ to perform the task behaviors and improve the convergence rate.

IV. Preliminaries

A. Minimum Connectivity Constraint

To guarantee connectivity and avoid collisions, we make use of the previous work on barrier certificate [22] and behavior mixing with minimum connectivity constraint spanning tree [14] that distributedly computes the control output of the robots given safety and connectivity constraints. In particular, the resulting control output $\mathbf{u} = [u_1, \dots, u_n] \in \mathbb{R}^{2\times n}$ is obtained by minimally modifying the given task-related control input $\hat{\mathbf{u}} = [\hat{u}_1, \dots, \hat{u}_n] \in \mathbb{R}^{2\times n}$, i.e. minimizing $\sum_{i=1}^{n} ||u_i - \hat{u}_i||^2$ subject to collision avoidance and connectivity maintenance constraints. Safety and connectivity constraints are enforced on pair-wise distance between robots:

$$h^s_{i,j}(\mathbf{x}) = ||x_i - x_j||^2 - r^2 \geq 0$$
$$h^c_{i,j}(\mathbf{x}) = R^2 - ||x_i - x_j||^2 \geq 0, \forall i, j$$

Safety and connectivity barrier certificate are define as [22]:

$$\mathcal{B}^s = \{\mathbf{u} \in \mathbb{R}^{2\times n} : \dot{h}^s_{i,j}(\mathbf{x}) + \gamma h^s_{i,j}(\mathbf{x}) \geq 0, \forall i > j\}$$
$$\mathcal{B}^c(\mathcal{G}^c) = \{\mathbf{u} \in \mathbb{R}^{2\times n} : \dot{h}^c_{i,j}(\mathbf{x}) + \gamma h^c_{i,j}(\mathbf{x}) \geq 0, \forall e_{i,j} \in \mathcal{E}^c\}$$

where γ is the barrier gain, $\mathcal{G}^c = (\mathcal{V}^c, \mathcal{E}^c)$ is a desired connectivity graph to maintain, e.g. a spanning tree that invokes minimum connectivity constraints [14] in a distributed manner [28]. These constraints over controllers ensure the forward invariance of the condition defined above, i.e. robots stay collision free and connected at all time. The control output under constraints is calculated as:

$$\mathbf{u}^* = \arg\min_{\mathcal{G}^c, \mathbf{u}} \sum_{i=1}^{n} ||u_i - \hat{u}_i||^2 \quad (1)$$
$$\text{s.t.} \quad \mathcal{G}^c \subset \mathcal{G} \quad \text{is connected}$$
$$\mathbf{u} \in \mathcal{B}^s \bigcap \mathcal{B}^c(\mathcal{G}^c)$$

This gives us guarantee on the safety and connectivity constraints with any $\hat{\mathbf{u}}$, and serves as a post-processing after computing the desired control output to be introduced in later sections. The task-related controller is always assumed known in [14], [25] and some tasks may not be achieved due to connectivity constraints. In this paper, we propose correction controller for those untasked robots, so that the control deviation defined in (1) for the task robots is further minimized and hence improving the tasks performance.

B. Graph Laplacian and Convergence

Here we briefly discuss the relation between Graph Laplacian of the connectivity graph and the convergence rate of the system. Consider the *graph Laplacian* matrix L of the connectivity graph $\mathcal{G}$. The graph Laplacian is defined as $L = D - A$ where A is the adjacency matrix where each element $a_{ij} = 1$ if an edge exists between v_i and v_j, and zero otherwise. $D = \text{diag}(deg(v_1), \dots, deg(v_n))$ is the degree matrix. Each $deg(v_i)$ denotes the degree of v_i, where $deg(v_i) = \sum_{i\neq j} a_{ij}$, and zero for off-diagonal elements. For $\mathcal{V}' \subseteq \mathcal{V}$, $deg(\mathcal{V}') = [deg(v_i), \dots, deg(v_j)]$ where $v_i, \dots, v_j \in \mathcal{V}'$. By definition, L is symmetric if the graph is undirected. Laplacian matrix L is essential for evaluating the convergence of consensus algorithms [11]. The second smallest eigenvalue, denoted as $\lambda_2(L)$, describes the *algebraic connectivity* of the graph and $\lambda_2(L) \geq 0$ always holds when the graph is connected [26]. Let $\delta = \min deg(\mathcal{V})$ to be the minimum degree of the vertices in graph $\mathcal{G}$ with n vertices. A lower bound for the algebraic connectivity $\lambda_2(L)$ exists [29]

$$\lambda_2(L) \geq 2\delta - n + 2 \quad (2)$$

It is known that a continuous-time consensus is globally exponentially reached with a speed that is faster or equal to $\lambda_2(L_s)$ where $L_s = (L + L^T)/2$ for a strongly connected balanced digraph [11]. For undirected graph with a symmetric Laplacian matrix, we have $L = L^T$, thus $L = L_s = (L + L^T)/2$. In our setting, the speed of convergence is directly influenced by $\lambda_2(L)$. Thus, the goal of our correction controller is to move towards vertices with small degrees so as to improve the convergence rate of the whole system.

V. Methodology

In this section, we introduce our method, *topology correction*, where the robots with no task assigned will be able to improve the topology of the connectivity graph to provide flexibility for the task robots to perform their tasks. To simplify our description, the controller of the robots with no task for correcting topology is named as *correction controller*. Our method has two steps:

1) Robots with no task will execute the correction controller which will be discussed in detail later;
2) The system is guaranteed connectivity with the minimum connectivity constraint spanning tree from section IV-A.

As introduced in section IV-B, increasing the minimum degree δ of the graph can achieve a faster convergence rate and also provide a better topology of connectivity graph. The intuition is that to maintain flexible topology of the connectivity graph, those robots with fewer edges connected are more likely to get disconnected, thus need more "help" from the other robots. By controlling the robots with no tasks to go towards the robots with smaller degrees, we can achieve better topology that will provide a faster convergence rate of the whole system. Therefore, the correction controller we will be discussing aims to correct the connectivity graph's topology by *maximizing the minimum degree* δ.

A. Weighted Rendezvous

To form a better topology in favor of improving the convergence rate of the system , it is straight forward to think of *rendezvous* behavior that non-task robots could keep up with the task robots. Following the notation introduced in Section III, control law used for rendezvous [11] is

$$\dot{x}_i(t) = \sum_{j\in\mathcal{N}_i} a_{ij}(x_j(t) - x_i(t)) \quad (3)$$

where $\mathcal{N}_i$ denotes the neighbor set of robot i and a_{ij} is the element in adjacency matrix. Following the discussion in section IV-B, we propose the *weighted rendezvous* as follows

$$u^r_i(t) = \dot{x}_i(t) = \sum_{j\in\mathcal{N}_i} w^r_{ij}(x_j(t) - x_i(t)) \quad (4)$$

where the weight w^r_{ij} in the weight array $\mathbf{w}^r_i = \left[w^r_{i1}, \dots, w^r_{ij}, \dots, w^r_{i|\mathcal{N}_i|}\right]$, where $j \in \mathcal{N}_i$.

1) Degree-based Weighted Rendezvous: Following the discussion in section IV-B, we may set the weights with respect to the degree of neighboring vertices. The weights should be larger on vertices with smaller degrees and smaller on vertices with larger degrees aiming at balancing the degrees in order to increase the minimum degree of the graph. Therefore, the desired weights w^r_{ij} in the weight array $\mathbf{w}^r_i$ are calculated as

$$w^r_{ij} = \max\left(deg(\mathcal{N}_i)\right) - deg(v_j) + \epsilon, v_j \in \mathcal{N}_i \quad (5)$$

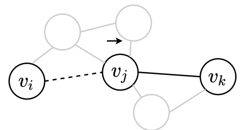


Fig. 2: The moment when vertex v_j is moving to v_k resulting in v_i and v_j getting disconnected from each other. We only consider the vertices involving possible edge changes (get connected or disconnected), while the grey vertices are other irrelevant vertices whose edges remain unchanged.

ϵ is a small value of $1 \gg \epsilon > 0$ and serves as a *correction factor* to guarantee that weights are not all zeros with regular graph (the graph that every vertex has same degree). The weight array is then normalized so that $\sum_{j\in\mathcal{N}_i} w^r_{ij} = 1$. We will then prove that this controller will correct the topology of the connectivity graph by *modifying the minimum degree* δ *of graph* $\mathcal{G}$.

As mentioned above, our goal is to increase the minimum degree δ of the connectivity graph. Thus, we are not interested in the cases where the original control output $\hat{\mathbf{u}}$ is already causing the minimum degree δ to increase. Before we start the proof, we list several cases where δ is already increasing: 1) adding an edge: Consider a situation when two robots come closer that a new connection is made, while in the meantime, no edges is disconnected. This will only cause δ to increase; 2) removing an edge e_{ij} where $deg(v_i) > \delta, deg(v_j) > \delta$: when two robots i and j move away from each other with a distance larger than R, the connection edge e_{ij} is removed. Notice that degrees are integers, we have $deg(v_i) \geq \delta+1$. Removing an edge e_{ij} will result in $deg(v_i) \geq \delta, deg(v_j) \geq \delta$, which does not change the value of δ.

Thus, the only case that is of interest is shown in Figure 2, which describes when v_i having minimum degree $deg(v_i) = \delta$, v_j moves towards its other neighbor v_k, causing edge e_{ij} to disconnect. This is the only case that will cause a decrease of δ and in the following proof of Preposition 1, we focus on this case. Note that with the minimum connectivity guarantee in section IV-A, the graph always remain connected. In the following discussion, we denote $\delta(t)$ and $deg(v_i, t)$ as the δ and $deg(v_i)$ at time t respectively.

Preposition 1. *With the control law defined in equation* (4) *with weights in equation* (5)*, the minimum degree* δ *of connectivity graph increases, i.e.* $\delta(t') \geq \delta(t)$ *with* $t' > t$ *where* t' *is the next time instance that degree of vertices changes.*

Proof. We prove by contradiction. As described above, the only case that will result in $\delta(t) > \delta(t')$ is shown in Figure 2. Without loss of generality, consider the edge that gets disconnected is $e_{ij} \in \mathcal{E}$. When v_j moves towards v_k, v_j disconnects with v_i where $deg(v_i, t) = \delta(t)$ at time t. According to equation (4), robot j moves towards robot k at $x_k(t)$ with velocity $\dot{x}_j(t)$ resulted from $w^r_{jk} > w^r_{ji}$. Thus

$$\begin{aligned} w^r_{jk} &= \max(deg(\mathcal{N}_j, t)) - deg(v_k, t) + \epsilon \\ &> \max(deg(\mathcal{N}_j, t)) - deg(v_i, t) + \epsilon = w^r_{ji} \end{aligned}$$

Then we have

$$deg(v_i, t) = \delta(t) > deg(v_k, t) \tag{6}$$

This violates the initial assumption that v_i with $deg(v_i, t) = \delta(t)$ has the minimum degree. Therefore, the case described above will not happen and any other scenarios will not result in decrease of δ, thus $\delta(t') \geq \delta(t)$, with $t' > t$. □

2) Distance-based Weighted Rendezvous: The degree-based approach can trigger edge changes to increase the convergence rate by gradually modifying δ in the graph. We notice that this degree-based approach will result in a sudden change of velocity. To further optimize our objective, we propose the *distance-based weighted rendezvous* based on the degree-based approach that provides the same guarantees and better performance.

We first define a *density score* of each vertex v_i as

$$c(v_i) = deg(v_i) + \sum_{j\in\mathcal{N}_i} \frac{R - d_{ij}}{deg(v_i)(R - r)} \tag{7}$$

$c(v_i)$ describes the level of density around v_i. An intuition on this is that when the neighboring robots are close to robot i, the distance d_{ij} is relatively small, then density score of v_i is higher, and vice versa. Similarly, w^r_{ij} in weight array $\mathbf{w}^r_i$ are calculated as

$$w^r_{ij} = \max(c(\mathcal{N}_i)) - c(v_j) + \epsilon, v_j \in \mathcal{N}_i \tag{8}$$

where ϵ is the same *correction factor* as in Equation (5). $\mathbf{w}^r_i$ is then normalized such that $\sum_{j\in\mathcal{N}_i} w^r_{ij} = 1$. We will then prove that this formulation gives the same guarantee as equation (5).

Lemma 2. *For any two connected vertices* $v_i, v_j \in \mathcal{V}$*,* $e_{ij} \in \mathcal{E}$ *, if* $deg(v_i) > deg(v_j)$*, then* $c(v_i) \geq c(v_j)$*.*

Proof. We prove by analyzing the relationship of $deg(v_i)$ and $c(v_i)$. With $d_{ij} = ||x_i - x_j||$ and $e_{ij} \in \mathcal{E}$, we have

$$r \leq d_{ij} \leq R \tag{9}$$

where as defined in section III, r is the safety radius and R is the connectivity radius. By applying the inequality in equation (9), we have

$$c(v_i) = deg(v_i) + \sum_{j\in\mathcal{N}_i} \frac{R - d_{ij}}{deg(v_i)(R - r)} \geq deg(v_i) \tag{10}$$

Notice that $deg(v_i) = |\mathcal{N}_i|$, we also have

$$c(v_i) \leq deg(v_i) + \sum_{j\in\mathcal{N}_i} \frac{R - r}{deg(v_i)(R - r)} = deg(v_i) + 1 \tag{11}$$

Combining the two equations above we get

$$deg(v_i) \leq c(v_i) \leq deg(v_i) + 1 \tag{12}$$

Since degree of vertex is integer, the inequality $deg(v_i) > deg(v_j)$ is the same as $deg(v_i) \geq deg(v_j) + 1$. Thus we have

$$c(v_i) \geq deg(v_i) \geq deg(v_j) + 1 \geq c(v_j) \tag{13}$$

Then we may conclude that $c(v_i) \geq c(v_j)$ holds when $deg(v_i) > deg(v_j)$. □

Our final statement is as follows. Lemma 2 proves that density score preserves the inequality of degree values. Following the same procedure of the proof for Prepostion 1, we may conclude that δ increases similarly for the density score defined in equation (7). Thus, the following theorem holds.

Theorem 3. *With the control law defined in equation* (4) *with weights defined in equation* (8)*, the minimum degree* δ *is increasing, i.e.* $\delta(t') > \delta(t), t' > t$.

The proof is similar with Preposition 1. With this distance-based weighted rendezvous, the non-task robots will be able to provide a more flexible topology for the whole system.

B. Weighted Behavior Combination

We noticed that during execution, when task robots need to reach locations close to each other, non-task robots might be blocking their way. To solve this problem, we propose *weighted behavior combination* that utilizes the combination of weighted rendezvous and a weighted flocking behavior introduced below.

1) Weighted Flocking: The control law is as follow:

$$u_i^f = \dot{x}_i = \sum_{j \in \mathcal{N}_i} w_{ij}^f u_j \tag{14}$$

The weights are designed so that the non-task robots flocks with the neighboring task robots when needed:

$$w_{ij}^f = \begin{cases} \frac{1}{|\mathcal{V}_t \cap \mathcal{N}_i|}, & \text{if } v_j \in \mathcal{V}_t \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

$|\mathcal{V}_t \cap \mathcal{N}_i|$ is the number of task robots in the neighbors of v_i.

2) Weighted Behavior Combination: Denote the output velocity of weighted rendezvous as u_i^r, weighted flocking as u_i^f for robot i. We then calculate the weights of performing weighted flocking for each non-task robot

$$\alpha_i = \frac{1}{2u_{max}} \left|\left| u_i^r - \frac{1}{|\mathcal{V}_t \cap \mathcal{N}_i|} \sum_{j \in \mathcal{V}_t \cap \mathcal{N}_i} u_j \right|\right| \cdot \gamma, v_i \in \mathcal{V}_c \tag{16}$$

where $\gamma \in (0, 1]$ is an user-defined value that describes the extent the user would want to avoid the cluttering problem due to weighted rendezvous. α_i describes the amount of influence that weighted rendezvous has on the neighboring task robots. For example, if the α_i is large, the influence of the weighted rendezvous behavior of the non-task robot on the task robot controller is large. In this case, it is preferred to lower the weighted rendezvous controller's weight for the non-task robots. The weighted behavior combination controller is then designed as

$$\hat{u}_i = (1 - \alpha_i) u_i^r + \alpha_i u_i^f \tag{17}$$

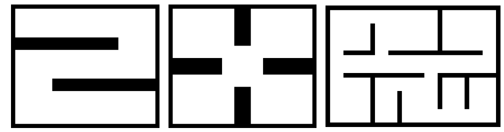
$\hat{u}_i$ will be passed to Equation (1) and outputs the optimized control input u_i that guarantees connectivity of the whole system. This method is also fully decentralized since weighted rendezvous and weighted flocking only take neighbor information to calculate the degree and density score. This distributed consensus-based method is scalable with an increasing number of robots in the system, shown in the next section.

VI. Results

A. Task Robot Controller

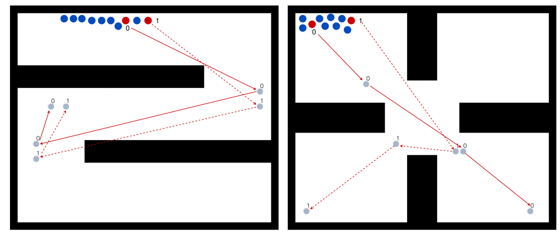
In our current scenario, we assume the task robots run a PD controller to reach designated goals. Note that the task robots do not need to be limited to this controller. As mentioned in Section III, the robot positions are denoted as x_i with single integrator dynamics $\dot{x}_i = u_i$. Consider the goal location for robot i is x_i^g, the controller used by the task robots is

$$u_i = \arg\min ||x_i - x_i^g||^2 \tag{18}$$



(a) (b) (c)

Fig. 3: (a) size 30×36, (b) size 30×36, and (c) size 27×39, are three maps with the black areas as walls or obstacles, and white areas as the free space.



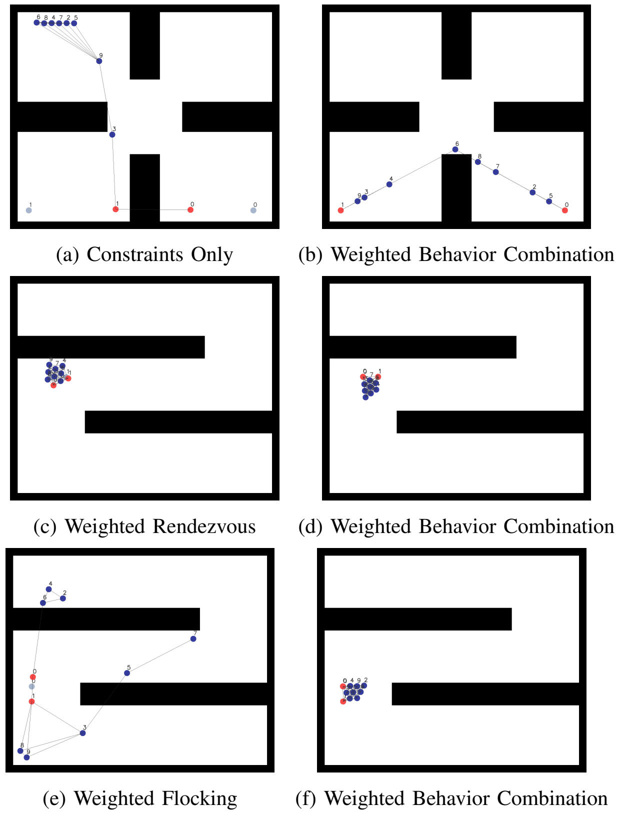
(a) Map 3a tasks (b) Map 3b tasks

Fig. 4: Tasks of Map 3a and Map 3b with $n = 10$ robots. Two red robots are assigned tasks, others are not assigned tasks in this time frame. Two sequences of tasks (randomly chosen within an area) are assigned to two robots.

B. Experiments

We tested our correction controller with the maps in Figure 3. In each map, the task robots have a sequence of locations to visit. The non-task robots execute the correction controller to correct the topology of the connectivity graph to provide flexibility for the task robots. We tested and compared the performance of 1) connectivity constraints only without any correction controller, 2) weighted rendezvous, 3) weighted flocking, 4) weighted behavior combination in the maps shown in Figure 3. Due to space limitation, the result in Map 3c will not be included in the paper but only in the video attachment in the supplementary material. A sequence of tasks are randomly chosen within an area that is feasible for the robots and are preassigned to specific robots before execution. The robots will execute the next task once the system has converged or the previous location is reached.

Figure 4 shows the sequence of tasks given that two of the robots in the system need to visit the goal locations in sequence while maintaining connectivity throughout the process. We tested the performance of the other robots without assigned tasks executing different controllers. Figure 5a shows that with only connectivity constraints, the robots without tasks might move to maintain connectivity but cannot provide flexible topology for the task robots to execute tasks. Figure 5c and 5e shows that robots with weighted rendezvous controller may block the way of the task robots. The robots with weighted flocking are more likely to leave robots behind, resulting in failing of some tasks. With weighted behavior combination, the robots without task assigned can provide a flexible topology of connectivity graph so that the task robots can reach all the goal locations.

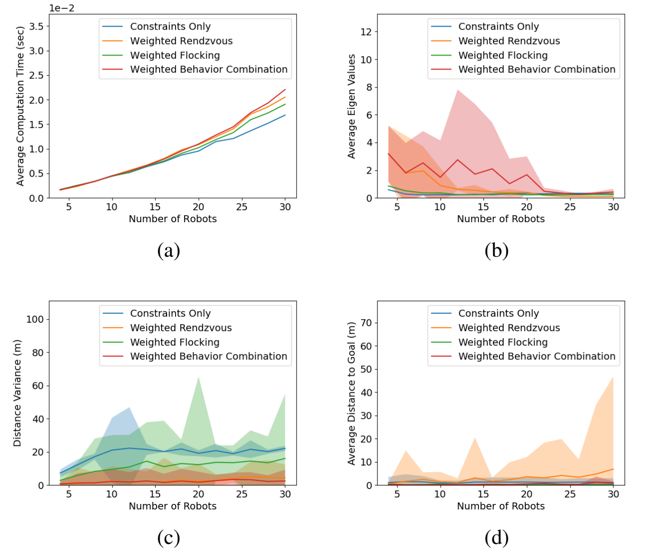


(a) Constraints Only (b) Weighted Behavior Combination

(c) Weighted Rendezvous (d) Weighted Behavior Combination

(e) Weighted Flocking (f) Weighted Behavior Combination

Fig. 5: Results of (a) robots without tasks are moving with only connectivity constraints executing task 3 in Map 3b, (c) robots with weighted rendezvous controller to execute task 3 in Map 3a, (e) robots with weighted flocking controller to execute task 2 in Map 3a . While (b) (d) (e) are executing the same tasks as in (a) (c) (e) corresponding with weighted behavior combination.



(a) (b)

(c) (d)

Fig. 6: (a) Average computation time for each iteration; (b) Average eigenvalues (c) Variance of the robot locations at each time of convergence; (d) Average distance to goal for the task robots at each time of convergence

We also measure the quantitative results of our experiments. The experiments include 15 runs with random initial locations and various robots in the system, ranging from $n = 4$ to $n = 30$. The method is written in python and tested on Intel Xeon CPU E5-2660 with cores of 2.60GHz. We present our result on average computation time in Figure 6a, average eigenvalues representing convergence rate in Figure 6b, variance of distance in Figure 6c and average distance to goal locations in Figure 6d at the time of convergence. Since the performance is highly related to the tasks and environment, the last three plots' variance are relatively large. In Figure 6a, the variance is too small given the time scale. The computation time shows our algorithms are scalable to a large number of robots and can run in real-time. In Figure 6b, weighted rendezvous and weighted behavior combination outperform other methods with a larger value of positive eigenvalue λ_2. This indicates that our method effectively increases network connectivity and ensures network connectivity ($\lambda_2 = 0$ equivalent to network disconnection). Note that we do not show the negative part because the eigenvalue does not go negative throughout the process. As shown in Figure 6c, we take into account the variance in locations of the robots after convergence. This is used to measure the robots' ability without tasks to keep up with the task robots not to be left behind, and our approach gives the best performance with the smallest distance variance. As shown in Figure 6d, we also compute the average distance to goal locations for the task robots to measure the performance of task robots and flexibility given by robots without tasks, where weighted behavior combination performs the best. Although with a larger number of robots, the average distance to the goal of weighted flocking is smaller than the weighted behavior combination, the distance variance is much larger. To that end, we conclude that weighted rendezvous can improve the convergence rate and provide flexibility, and weighted behavior combination has the best performance overall.

VII. Conclusion and Discussion

In this paper, we considered the system where some robots are assigned tasks at a specific time frame, and others are not. We study the problem of controlling those robots without tasks to maintain a flexible connectivity graph for the robots with an assigned controller to achieve domain tasks. We propose weighted rendezvous to reconfigure robots' motion with no assigned tasks (i.e., untasked robots) to provide flexibility for the other tasked robots to reach their goal. This is achieved by correcting the topology of the current connectivity graph with those untasked robots that can lead to a provable faster convergence rate for the tasked robots' controller in a distributed and scalable fashion. We also proposed the weighted behavior combination that improves the performance in general tasks to solve the cluttering problem due to rendezvous. Results have shown that our weighted behavior combination gives the best performance overall. Future work includes experiments on real robots for verification, examining the number of extra robots needed to maintain connectivity depends on tasks and environments, influences of package loss during communication.

References

[1] W. Burgard, M. Moors, C. Stachniss, and F. E. Schneider, "Coordinated multi-robot exploration," *IEEE Transactions on robotics*, vol. 21, no. 3, pp. 376–386, 2005.

[2] W. Luo and K. Sycara, "Adaptive sampling and online learning in multi-robot sensor coverage with mixture of gaussian processes," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 6359–6364.

[3] G. Hollinger, S. Singh, J. Djugash, and A. Kehagias, "Efficient multi-robot search for a moving target," *The International Journal of Robotics Research*, vol. 28, no. 2, pp. 201–219, 2009.

[4] G. Kantor, S. Singh, R. Peterson, D. Rus, A. Das, V. Kumar, G. Pereira, and J. Spletzer, "Distributed search and rescue with robot and sensor teams," in *Field and Service Robotics*. Springer, 2003, pp. 529–538.

[5] L. Moreau, "Stability of multiagent systems with time-dependent communication links," *IEEE Transactions on automatic control*, vol. 50, no. 2, pp. 169–182, 2005.

[6] 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.

[7] M. M. Zavlanos and G. J. Pappas, "Distributed connectivity control of mobile networks," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1416–1428, 2008.

[8] Z. Mi, Y. Yang, and G. Liu, "Hero: A hybrid connectivity restoration framework for mobile multi-agent networks," in *IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 1702–1707.

[9] G. Hollinger and S. Singh, "Multi-robot coordination with periodic connectivity," in *IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 4457–4462.

[10] 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.

[11] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[12] S. Nagavalli, N. Chakraborty, and K. Sycara, "Automated sequencing of swarm behaviors for supervisory control of robotic swarms," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 2674–2681.

[13] W. Luo and K. Sycara, "Voronoi-based coverage control with connectivity maintenance for robotic sensor networks," in *International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*. IEEE, 2019, pp. 148–154.

[14] W. Luo, S. Yi, and K. Sycara, "Behavior mixing with minimum global and subgroup connectivity maintenance for large-scale multi-robot systems," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 9845–9851.

[15] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Transactions on automatic control*, vol. 51, no. 3, pp. 401–420, 2006.

[16] M. M. Zavlanos, A. Jadbabaie, and G. J. Pappas, "Flocking while preserving network connectivity," in *Decision and Control, 46th IEEE Conference on*. IEEE, 2007, pp. 2919–2924.

[17] J. Banfi, N. Basilico, and S. Carpin, "Optimal redeployment of multirobot teams for communication maintenance," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 3757–3764.

[18] L. Guerrero-Bonilla, D. Saldana, and V. Kumar, "Design guarantees for resilient robot formations on lattices," *IEEE Robotics and Automation Letters*, vol. 4, no. 1, pp. 89–96, 2018.

[19] N. Michael, M. M. Zavlanos, V. Kumar, and G. J. Pappas, "Maintaining connectivity in mobile robot networks," in *Experimental Robotics*. Springer, 2009, pp. 117–126.

[20] T. Nestmeyer, P. R. Giordano, H. H. Bülthoff, and A. Franchi, "Decentralized simultaneous multi-target exploration using a connected network of multiple robots," *Autonomous Robots*, vol. 41, no. 4, pp. 989–1011, 2017.

[21] P. Robuffo Giordano, A. Franchi, C. Secchi, and H. H. Bülthoff, "A passivity-based decentralized strategy for generalized connectivity maintenance," *The International Journal of Robotics Research*, vol. 32, no. 3, pp. 299–323, 2013.

[22] U. Borrmann, L. Wang, A. D. Ames, and M. Egerstedt, "Control barrier certificates for safe swarm behavior," *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 68–73, 2015.

[23] W. Luo, W. Sun, and A. Kapoor, "Multi-robot collision avoidance under uncertainty with probabilistic safety barrier certificates," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[24] W. Luo and K. Sycara, "Minimum k-connectivity maintenance for robust multi-robot systems," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 7370–7377.

[25] L. Wang, A. D. Ames, and M. Egerstedt, "Multi-objective compositions for collision-free connectivity maintenance in teams of mobile robots," in *IEEE 55th Conference on Decision and Control (CDC)*. IEEE, 2016, pp. 2659–2664.

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

[27] D. Pickem, P. Glotfelter, L. Wang, M. Mote, A. Ames, E. Feron, and M. Egerstedt, "The robotarium: A remotely accessible swarm robotics research testbed," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 1699–1706.

[28] R. G. Gallager, P. A. Humblet, and P. M. Spira, "A distributed algorithm for minimum-weight spanning trees," *ACM Transactions on Programming Languages and systems (TOPLAS)*, vol. 5, no. 1, pp. 66–77, 1983.

[29] N. M. M. De Abreu, "Old and new results on algebraic connectivity of graphs," *Linear algebra and its applications*, vol. 423, no. 1, pp. 53–73, 2007.