Deadlock Resolution in Multi-Agent Systems using Hierarchical Control

Kunal Garg Sera Hamilton Chuchu Fan

***Abstract*— Multi-agent robotic systems often require control design for a multi-objective problem, such as maintaining a safe distance from other agents as well as obstacles, maintaining network connectivity for building team knowledge and completing team objectives for performance. Such problems are intractable in the centralized framework for large-scale systems. Thus, a distributed framework is necessary where each agent only requires its neighbors' information while being able to contribute towards completing the team objective. However, a decentralized control framework often leads to a sub-optimal solution, resulting in the system getting stuck in local minima or a deadlock. This paper addresses the issue of deadlock resolution via a hierarchical control framework. We propose a high-level planner for temporary goal assignment and a low-level controller that drives the agents to their assigned goals. The proposed framework is distributed in nature, making it scalable to large-scale multi-agent systems. We perform extensive simulation and experimental case studies to demonstrate the efficacy and need for such a hierarchical control framework.**

I. Introduction

Multi-agent systems (MAS) have a lot of potential applications in today's world such as warehouse operations [1], [2], self-driving cars [3], coordinated navigation of drones in a dense forest for search-and-rescue missions [4], among others. For such safety-critical MASs, it is important to design controllers that not only guarantee safety in terms of collision and obstacle avoidance but are also scalable to large-scale multi-agent problems. Furthermore, in certain applications of MAS such as coverage [5] and formation control [6], it is also required that the underlying graph topology remain connected for sharing information and building team knowledge. Existing methods for multi-agent coordination and motion planning are incapable of solving such problems that consider all these aspects, i.e., safety, connectivity, and performance, in terms of reaching a goal destination or following a given trajectory, in a scalable manner.

Connectivity: Assessing the connectivity of a MAS in a centralized framework is a relatively easy problem. One only needs to compute the second smallest eigenvalue of the corresponding Laplacian matrix of the graph topology to check the connectivity [7]. However, maintaining the connectivity of a network in a centralized manner is not scalable as it requires extensive exchange of information among the nodes. As shown in [8], it is possible to assess and maintain the connectivity of the global network in a distributed manner using local information. While there is a lot of work on maintaining the initial graph topology of MAS (see [9], [10]), it is restrictive to not allow the network topology to change as it can lead to suboptimal performance. There is very little work on allowing graph topology to change while maintaining connectivity in a scalable manner [8]. Furthermore, to the best of the authors' knowledge, there is no work in the literature that reconfigures the network topology to assist with deadlock resolution and completing the global team objective in a distributed manner.

The authors are with the Department of Aeronautics and Astronautics at MIT, {kgarg, seras, chuchu}@mit.edu.

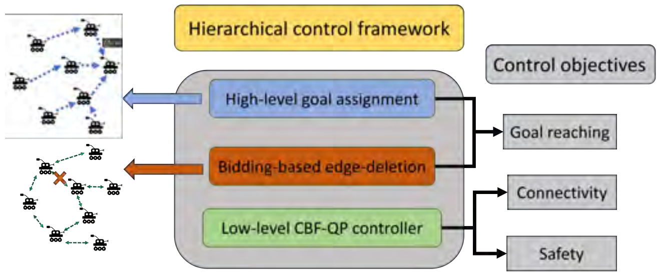
Hierarchical control framework
Control objectives
High-level goal assignment
Goal reaching
Bidding-based edge-deletion
Connectivity
Low-level CBF-QP controller
Safety

Fig. 1: Overview of the hierarchical control framework.

Safety: For the safety objective, in recent years, control barrier functions (CBFs) have become very popular in both single and multi-agent problems for safety-critical systems [11]–[15] where the existence of a CBF guarantees the existence of a control input that can keep the system safe. The CBFs are used in a quadratic programming (QP) framework for online control synthesis. However, as discussed in [16], CBF-based methods are myopic in nature and only using CBFs for a multi-task problem such as maintaining safety and connectivity, and reaching a destination location for performance, can potentially lead to infeasibility of the underlying QP. They may also lead to the system getting stuck in an undesirable equilibrium point away from the desired destination [17]. A high-level planner for addressing some of these issues has been studied for single-agent systems [18], [19] but not for multi-agent systems.

Multi-agent control: Other multi-agent motion planning methods include path planning methods that generate safe paths for each agent [20]. Such methods include but are not limited to variants of rapidly exploring random trees (RRT) [21], [22], solving mixed integer linear program (MILP) for computing safe paths for agents [23]–[25] or model predictive control (MPC)-based methods [26]. However, all of these methods suffer from scalability issues and do not consider connectivity maintenance as an objective. Moreover, predictive methods such as MPC are also difficult to implement in a distributed manner without making strong assumptions about other agents' behaviors or using additional compatibility constraints [26]. Furthermore, MPC suffers from the curse of dimensionality with an increasing number of agents in MAS, and its computational complexity does not permit its usage in real-time control synthesis (as illustrated in numerical experiments). More recently, learning-based methods have shown promising results in safe multi-agent coordination [14], [27], [28]. However, the focus of these works is collision avoidance, not deadlock resolution.

Contributions In this paper, we present a hierarchical control framework that uses a high-level planner that helps to resolve deadlocks so that the MAS does not get stuck in local minima and a CBF-QP-based low-level controller that helps guarantee safety, connectivity maintenance, and performance. To avoid deadlocks, we propose a heuristic temporary goal assignment for the high-level planner that uses each agent's local observations. We employ a bidding-based mechanism [8] at the low level to assist with the completion of the multi-objective problem. In particular, we remove edges from the graph topology that help the MAS move closer to its destination. Finally, we illustrate the efficacy of the proposed hierarchical control framework on hardware experiments where a nominal CBF-QP gets stuck in deadlocks.

II. Problem formulation

In this work, we design a distributed control framework for large-scale robotic systems with multiple objectives. First, we start with describing the dynamics of the individual robots (referred to as agents henceforth), and then, we list the individual as well as the team objective for the system. The agent dynamics are given by $\dot{z}_i = f(z_i) + g(z_i)u_i$, where f, g are locally Lipschitz continuous functions with $\mathcal{X} \subset \mathbb{R}^{n_x}$ denoting agents' workspace and $\mathcal{U}_i \subset \mathbb{R}^{n_u}$ the control constraint set. We use $p_i \subset z_i$ to denote the position (or location) of the $i-$th agent in the global coordinates.

The workspace $\mathcal{X}$ consists of stationary obstacles $\mathcal{O}_l \subset \mathbb{R}^2$ for $l \in \{1, 2, \dots, M\}$, denoting walls, blockades and other obstacles in the path of the moving agents. Each agent has a limited sensing radius $R_s > 0$ and the agents can only sense other agents or obstacles if it lies inside its sensing radius. The time-varying connectivity graph $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$ dictates the communication network among the agents, i.e., if there is an edge between agent i and j in $\mathcal{G}(t)$ at a time instant $t \geq 0$, then the agents (i, j) can exchange information necessary for completing the team objective. Here, $\mathcal{V} = \{1, 2, \dots, N\}$ denotes the set of nodes and $\mathcal{E}(t) \in \mathbb{R}^{N \times N}$ denotes the set of edges where $(i, j) \in \mathcal{E}(t)$ if $\|p_i(t) - p_j(t)\| \leq R_s$. The corresponding time-varying adjacency matrix, given as $\mathcal{A}_{ij}(t) = 1$ if $\|p_i(t) - p_j(t)\| \leq R_s$ and 0 otherwise. The set of neighbors for agent i is denoted as $\mathcal{N}_i(t) := \{j \mid \mathcal{A}_{ij}(t) = 1\}$. The graph topology $\mathcal{G}(t)$ is said to be connected at time $t \geq 0$ if there is a path between each pair of nodes (i, j) at t. One method of checking the connectivity of the graph topology is through its Laplacian matrix, defined as $L(\mathcal{A}(t)) := D(t) - \mathcal{A}(t)$, where $D(t)$ is the degree matrix defined as $D_{ij}(t) = \sum_j \mathcal{A}_{ij}(t)$ when $i = j$ and 0 otherwise. From [29, Theorem 2.8], we know that the graph topology $\mathcal{G}(t)$ is connected at time t if and only if the second smallest eigenvalue of the Laplacian matrix is positive, i.e., $\lambda_2(L(\mathcal{A}(t))) > 0$. Now we have all the elements to introduce the problem statement.

Problem 1. *Consider the multi-agent system with initial conditions $\{x_i(0)\}$ such that the underlying graph $\mathcal{G}(t)$ is connected at $t = 0$, safety parameters $d_m, r_{obs} > 0$, a sensing radius $R_s > 0$, a set of stationary obstacles $\{\mathcal{O}_j\}$, and goal locations $\{p_{gi}\}$. Design a control policy π_i using information available in agent i's sensing radius, such that*

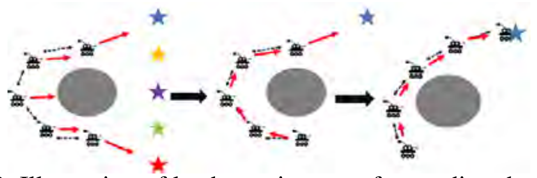
Fig. 2: Illustration of leader assignment for evading deadlock situation due to presence of an obstacle.

1. ***Safety**: Agents maintain a safe distance from other agents and obstacles at all times, i.e., $\|p_i(t) - p_j(t)\| \geq 2d_m$ and $\min_l \min_{p \in \mathcal{O}_l} \|p_i(t) - p\| \geq d_m$ for all $t \geq 0$;*
2. ***Connectivity**: The graph $\mathcal{G}(t)$ remains connected at all times, i.e., $\lambda_2(\mathcal{L}_2(t)) > 0$ for all $t \geq 0$;*
3. ***Performance**: Agents reach their respective goal locations, i.e., $\|p_i(t) - p_{gi}\| \to 0$.*

Let $\mathcal{M}_i(t) := \{j \mid \min_{p \in \mathcal{O}_j} \|p_i(t) - p\| \leq R_s\}$ denote the set of obstacles that are in agent i's sensing region at time t.

III. Hierarchical control architecture

A. High-level planner

In a multi-agent motion planning problem without a high-level *supervisor*, the presence of obstacles can lead to deadlock situations as illustrated in the first image of Figure 2. Here, the agents at the end of the network chain try to go directly toward their respective goal locations while trying to maintain connectivity as well. Such situations can be evaded by assigning a temporary goal location via *leader* assignment where the assigned leader acts as temporary goal $x_{gi,temp}$ for agent i. In a simple scenario consisting of a small number of agents and obstacles such as the one illustrated in Figure 2 with just 5 agents and 1 obstacle, it is possible to assign the leader manually. We propose a mechanism for choosing a leader and temporary goal assignment (see Algorithm 1).

1) Leader-based goal assignment: We use a leader-based goal assignment to resolve a deadlock due to the presence of an obstacle. This assignment is triggered when the average speed of the MAS $\frac{\sum_i |\dot{p}_i|}{N} < u_{min}$, i.e., it falls below a minimum threshold $u_{min} > 0$. Since the graph topology is connected, the average speed (which requires global information) can be computed through consensus updates. In this case, the agent closest to its goal and with *maximum mobility* is assigned as the leader of the MAS. That, is the leader is chosen as

$$\text{lead} = \arg\min_{i \in \mathcal{A}} \frac{\|p_i - p_{gi}\|}{\|\dot{p}_i|_{p_{gi,temp}}\|}, \tag{1}$$

where $\dot{p}_i|_p$ denotes the derivative of the position of the $i-$th agent under the temporary goal $p_{gi,temp}$. The moving speed $|\dot{p}_{lead}\|$ of the leader depends on whether the leader is moving towards its *actual* goal locations or a *temporary* goal location, as described later. The follower assignment is carried out as follows. Let $\mathcal{A}_{lead}(t)$ be the set of agents that have been assigned as a leader with the set initiated as $\mathcal{A}_{lead}(t, 0) = \{\text{lead}\}$. Then, the $k-$th follower with $k \geq 1$ is chosen as

$$\text{follow}_k = \arg\min_{j \in \mathcal{A} \setminus \mathcal{A}_{lead(t,k-1)}} \min_{i \in \mathcal{A}_{lead}(t,k-1)} \|p_i - p_j\|, \tag{2}$$

and this follower is added to the set of the leaders, i.e., $\mathcal{A}_{lead}(t,k) = \mathcal{A}_{lead}(t,k-1) \cup \{\text{follow}_k\}$. The leader for the $k-$th follower is given as $\text{lead}_k = \arg\min_{i\in\mathcal{A}_{lead}(t,k-1)} \|p_{\text{follow}_k} - p_i\|$. The process is repeated till each agent is assigned an agent to follow. Next, for each agent i that is a given minimum distance away from its goal, i.e., if $\|p_i - p_{gi}\| \geq d_{min}$, their temporary goal is chosen as the location of their leaders, i.e., $p_{gi,temp} = p_{i_{\text{lead}}}$.

2) Temporary goal for leader: In the leader-follower mode, the lead agent may be stuck in a deadlock. To tackle this, a temporary goal for the lead agent $p_{g\text{lead},temp}$ is assigned for the leader as

$$p_{g\text{lead},temp} = \underset{p\in\mathbb{B}\oplus p_{lead}}{\arg\max}\ \|\dot{p}_{lead}|_p\|,$$

where $\mathbb{B} \oplus p \subset \mathbb{R}^2$ is a unit circle centered at $p \in \mathbb{R}^2$. Intuitively, the temporary goal is assigned in a way that leads to the maximum moving speed of the leader.

Algorithm 1: Leader-follower assignment

```
Data: $E, \{p_i\}, \{p_{gi}\}, \{|\dot{p}_i|\}, u_{min}, d_{min}$
Result: $p_{g,temp}$, lead
if $\frac{\sum_i |\dot{p}_i|}{N} < u_{min}$ then
    lead $= \arg\min \frac{\|p_i - p_{gi}\|}{\|\dot{p}_i|_{p_{gi,temp}}\|}$
    $\mathcal{A}_{lead(t,0)} = \{\text{lead}\}$
    for $k \in [1, N-1]$ do
        choose follow$_k$ per (2)
        $\mathcal{A}_{lead(t,k)} = \mathcal{A}_{lead(t,k-1)} \cup \{\text{follow}_k\}$
        $k_{\text{lead}} = \arg\min_{i\in\mathcal{A}_{lead}(t,k-1)} \|p_{\text{follow}_k} - p_i\|$
        if $\|p_k - p_{gk}\| \geq d_{min}$ then
            $p_{gk,temp} = p_{k_{\text{lead}}}$
        end
    end
    if $|\dot{p}_{lead}| < u_{min}$ and $\|p_{lead} - p_{glead}\| \geq d_{min}$ then
        $p_{g\text{lead},temp} = \underset{p\in\mathbb{B}\oplus p_{lead}}{\arg\max} \|\dot{p}_{lead}|_p\|$
    end
end
```

Terminal goal re-assignment: If any of the agents reach their goals and obstruct the rest of the agents, the agent that is stuck due to another agent that has already reached its goal location swaps its goal location with the latter.

We prove that the proposed algorithm does not get stuck in any deadlocks in Appendix I, where deadlocks are defined as the existence of a non-zero length period where the average speed of the MAS is zero.

B. QP-based low-level planner

The high-level planner provides the leader and the goal information to the controller at the low level. The low-level controller synthesizes an input u_i to keep the system safe from obstacles and other agents and drive the system trajectories toward its goal location. Safety is encoded using control barrier functions (CBFs) while convergence to the goal via control Lyapunov functions (CLFs). For defining the CLFs and CBFs, we use a unicycle dynamical model for the agents given as $\dot{x}_i = v_i \cos\theta_i, \quad \dot{y}_i = v_i \sin\theta_i, \quad \dot{\theta}_i = \omega_i$ where $z_i = (x_i, y_i, \theta_i)$ denotes the states and $u_i = (v_i, \omega_i)$ the control input of the $i-$th agent for $i \in \{1, 2, \ldots, N\}$. We define $p_i = (x_i, y_i)$ as the position vector of the $i-$th agent and $\mathcal{X} \subset \mathbb{R}^3$ be the workspace of the MAS.

Recall that the safety constraint for the $l-$th obstacle is defined as $x_i \in S_l = \{x = (p, \cdot) \mid \min_{p_o\in\mathcal{O}_l} \|p - p_o\| \geq d_m\}$. To this end, we define a quadratic barrier function as $B_{l,p}(p_i) = \frac{1}{2}\min_{p_o\in\mathcal{O}_l} \|p_i - p_o\|^2 - \frac{1}{2}r_{obs}^2$ so that $B_{l,p}(p_i) \geq 0$ implies that the agent i is at a safe distance r_{obs} from the $l-$th obstacle.

The steering input ω_i does not appear in the time-derivative of the function $B_{l,p}$ along the $i-$th agents trajectories. With this choice of CBF, while it is possible to maintain safety (using $u_i = 0$ as input), it is not possible to guarantee that the agents reach their goal locations. Thus, we introduce an additional barrier function for evading the obstacle, defined as $B_{l,\theta}(\theta_i) = \frac{1}{2}|\theta_i - \theta_o\| - \frac{1}{2}\theta_{threshold}^2$, where $\theta_o = \tan^{-1}\left(\frac{y_o - y_i}{x_o - x_i}\right)$ with $(x_o, y_o) = \arg\min_{p_o\in\mathcal{O}_l} \|p_i - p_o\|$ and $\theta_{threshold} > 0$, so that the input ω_i helps steer the agent i away from the obstacle $\mathcal{O}_l$. For a convex-shaped obstacle with a smooth boundary (i.e., defined as a level-set of a smooth function), the functions $B_{l,p}$ and $B_{l,\theta}$ are also smooth. Similarly, barrier functions for safety with neighbors $B_{ij,p}, B_{ij,\theta}$ and for connectivity $\tilde{B}_{ij,p}, \tilde{B}_{ij,\theta}$ can be defined. Furthermore, for convergence to the goal, we can define quadratic Lyapunov functions $V_p(p_i) := \frac{1}{2}\|p_i - p_{gi}\|^2$ and $V_\theta(\theta_i) := \frac{1}{2}|\theta_i - \theta_{ig}|^2 - \theta_{g,thres}^2$ where $\theta_{ig} = \tan^{-1}\left(\frac{y_{gi} - y_i}{x_{gi} - x_i}\right)$.

Using the CBF theory for safety (see e.g., [12]), we use the following CBF conditions $\dot{B}_{l,p}(p_i) \geq -\alpha_{l,p}\ B_{l,p}(p_i), \dot{B}_{l,\theta}(\theta_i) \geq -\alpha_{l,\theta}\ B_{l,\theta}(\theta_i)$ for all $l \in \mathcal{M}_i$, $\dot{B}_{ij}(x_i) \geq -\alpha_{ij}\ B_{ij}(x_i)$ and $\dot{\tilde{B}}_{ij}(x_i) \geq -\tilde{\alpha}_{ij}\ \tilde{B}_{ij}(x_i)$ for all $j \in \mathcal{N}_i$, in the respective safe sets. Here $\alpha_{l,p}, \alpha_{ij}, \tilde{\alpha}_{ij} \in \mathbb{R}$ are scalar variables. Note that the time derivative of B_{ij} (and $\tilde{B}_{ij}$) reads $\dot{B}_{ij} = L_f B_{ij}(x_i) + L_g B_{ij}(x_i)u_i + \frac{\partial B_{ij}}{x_j}\dot{x}_j$, and requires $\dot{x}_j$. The evaluation of $\dot{x}_j$ requires (x_j, u_j) which can be obtained through active communication between neighboring agents. In a distributed infrastructure with a limited sensing radius, it is reasonable to assume active communication with (possibly a small number of) neighbors.

Note that the CBF and CLF are linear inequalities in the input variable u_i. Also, assume that the control input constraints are given as $\mathcal{U}_i = \{u \mid A_u u \leq b_u\}$ with appropriately sized matrix and vector A_u, b_u. Noting this, and using the relaxed CLF constraint from [30] and slack-variable variation of CBFs from [31], we set a QP with the CBF-CLF conditions and the input constraints as the linear inequality constraints to minimize a quadratic cost consisting of the input u_i and the slack terms introduced for feasibility. In particular, a CBF condition with slack term takes the form $L_f B_{l,p}(p_i) + L_g B_{l,p}(p_i)u_i \geq -\alpha_{l,p}\ B_{l,p}(p_i)$ with $\alpha_{l,p} \in \mathbb{R}$ a variable of optimization along with the input u_i. We set up the following QP($\tilde{\mathcal{N}}_i$) with $z_i = (u_i, \{\alpha_{l,p}\}, \{\alpha_{l,\theta}\}, \{\alpha_{ij,p}\}, \{\alpha_{ij,\theta}\}, \{\tilde{\alpha}_{ij,p}\}, \{\tilde{\alpha}_{ij,\theta}\}, \delta_p, \delta_\theta)$

for each agent i:[1]

$$\min_{z_i} \quad \frac{1}{2} z_i^T Q z_i + F_i^T z_i, \tag{3a}$$

$$\text{s.t. } \dot{B}_{l,p}(x_i) \geq -\alpha_{l,p} B_{l,p}(x_i), \forall l \in \mathcal{M}_i \tag{3b}$$

$$\dot{B}_{l,\theta}(x_i) \geq -\alpha_{l,\theta} B_{l,\theta}(x_i), \forall l \in \mathcal{M}_i \tag{3c}$$

$$\dot{B}_{ij,p}(x_i) \geq -\alpha_{ij,p} B_{ij,p}(x_i), \forall j \in \mathcal{N}_i \tag{3d}$$

$$\dot{B}_{ij,\theta}(x_i) \geq -\alpha_{ij,\theta} B_{ij,\theta}(x_i), \forall j \in \mathcal{N}_i \tag{3e}$$

$$\dot{\tilde{B}}_{ij,p}(x_i) \geq -\tilde{\alpha}_{ij,p} \tilde{B}_{ij,p}(x_i), \forall j \in \tilde{\mathcal{N}_i} \tag{3f}$$

$$\dot{\tilde{B}}_{ij,\theta}(x_i) \geq -\tilde{\alpha}_{ij,\theta} \tilde{B}_{ij,\theta}(x_i), \forall j \in \tilde{\mathcal{N}_i} \tag{3g}$$

$$\dot{V}_p(x_i) \leq -\lambda_p \ V_p(p_i) + \delta_p \tag{3h}$$

$$\dot{V}_\theta(\theta_i) \leq -\lambda_\theta \ V_\theta(\theta_i) + \delta_\theta, \tag{3i}$$

$$A_u u_i \leq b_u, \tag{3j}$$

where Q is a positive definite matrix weighing cost on various optimization variables and $F_i = [-u_{i,prev}, -1, -1, \ldots, -1, 1]^T$ where $u_{i,prev}$ is the last computed control input of agent i. The choice of the linear term $F^T z$ helps in choosing an input u_i that is *close* to the previously computed input so that the input signal does not have large jumps, while -1 scaling on slack variables $\alpha_i, \alpha_{ij}, \tilde{\alpha}_{ij}$ penalizes their negative values and promotes their positive values, and $+1$ scaling on δ penalizes its positive values and promotes its negative values. The argument $\tilde{\mathcal{N}_i}$ in QP($\tilde{\mathcal{N}_i}$) denotes the set of neighbors for which connectivity constraint is imposed in (3f)-(3g). This formulation is required for the bidding mechanism in Section III-C. Note that the QP (3) for agent i requires the control actions of neighbors $j \in \mathcal{N}_i$. Some works address this problem and propose methods of solving CBF-QPs for MRS in a distributed fashion (see e.g., [32]). Based on [31], the following result can be stated about the feasibility of the QP and continuity of its solution, which is required for using the existence and uniqueness of the closed-loop solutions for Nagumo's theorem [33] to guarantee safety via forward invariance arguments.

Theorem 1. *Assume that the QP* (3) *is feasible at the boundary of the intersection of the sets* $S_i, S_{ij}, \tilde{S}_{ij}$. *Then, the QP* (3) *is feasible for all* $x_i \in S_i \cup S_{ij} \cup \tilde{S}_{ij} \setminus \{x_{gi}\}$. *Furthermore, if the strict complementary slackness condition ([34]) holds, then the solution* z_i^* *of QP* (3) *is continuous on the set* $\text{int}\left(S_i \cup S_{ij} \cup \tilde{S}_{ij} \setminus \{x_{gi}\}\right)$ *and the resulting control input* u_i^* *keeps the agent* i *safe from its neighbors and obstacles and connected to its neighbors at all times.*

Remark 1. *It is possible to use more sophisticated CBFs and CLFs for the safety and convergence requirements, respectively. In particular, learning-based approaches as adopted in [28] can be used to learn distributed certificates for MAS.*

The multi-constrained QP can lead to agents getting stuck in local minima. In particular, under conflicting constraints of maintaining connectivity with a neighbor situated at one location and another neighbor situated at a diagonally opposite end. Such deadlock situations become a hindrance for the multi-agent team in completing their task. In many such situations, it might be possible to delete an edge while maintaining the connectivity of the graph, such that leaving a neighbor out of the connectivity constraint enables the QP to find a solution that moves the team along their desired paths. Motivated by this, we propose a novel mechanism of removing edges based on an auction mechanism in [8] that can help the MAS achieve its team objective.

[1]The actual CBF-CLF inequalities are omitted for the sake brevity.

C. New auction mechanism for edge deletion

The authors in [8] presented a bidding mechanism to remove edges from the graph topology $\mathcal{G}$ while maintaining the connectivity. For the sake of brevity, we summarize the three major steps of the auction mechanism:

1) Computing the set of *safe* neighbors such that removing an edge with *any one* of such neighbors does not break the local (and consequently, global) connectivity;
2) Bidding to break edge with one such safe neighbor; and
3) Using a max-consensus update (see [8]) to find the winning bid and remove an edge from the graph topology.

We use the same auction mechanism with specific bidding designed for resolving deadlocks and completing the team objective. We propose a bidding method where each agent computes a low-level controller assuming that a particular connection does not exist. Then, based on whether that controller helps achieve the team objective or not, the agent bids a higher or a lower value. The bidding method is described in more detail below.

QP-based bidding The second step in the auction mechanism requires each agent i to choose an edge from the set $\mathcal{S}_i$ and a corresponding bid b_i for the auction. To this end, we propose a bidding mechanism based on the solution of the QP. In particular, for each agent i, let QP($\mathcal{N}$) denote an instance of a QP where the connectivity constraints are imposed for each neighbor $j \in \mathcal{N}$ and its corresponding solution as $u_i^*(\mathcal{N})$. At a time t, we first compute the set of safe neighbors $\mathcal{S}_i(t)$. Then, we compute the set of control inputs $\{u_i^*(\mathcal{N}_i \setminus j_i)\}_{j_i \in \mathcal{S}_i}$. Using this, we propagate the system dynamics one step in future to compute $\{x(t+1, \mathcal{N} \setminus j_i\}_{j_i \in \mathcal{S}_i}$ and the corresponding distances of the agent i from its goal location $\{d_i(t+1, \mathcal{N} \setminus j_i)\}_{j_i \in \mathcal{S}_i}$ where $d_i = \|x_i - x_{gi}\|$. Then the selection function g and the bid values b_i are defined as follows when $\mathcal{S}_i \neq \emptyset$:

$$g(\mathcal{S}_i) = \underset{j_i \in \mathcal{S}_i}{\arg\max} \left(\lambda_2(\mathcal{A}^i \setminus (i, j_i)) \frac{\|d_i(t)\|}{\|d_i(t+1, \mathcal{N}_i \setminus j_i)\|} \right), \tag{4a}$$

$$b_i = K_i \min\left(100, \max_{j_i \in \mathcal{S}_i} \left(\frac{\|\dot{p}_i|_{u_i^*(\mathcal{N}_i \setminus j_i)}\|}{\|\dot{p}_i|_{u_i^*(\mathcal{N}_i)}\|} \right)\right), \tag{4b}$$

where we choose $K_i = 1 + \epsilon$ if $i = g(\mathcal{S}_{g(\mathcal{S}_i)})$ and $K_i = 1$ otherwise, with $\epsilon > 0$, to handle tiebreaks. The selection function g chooses a safe neighbor based on whether removing an edge helps move the agent i closer to its destination. The bid value is based on the norm of the control input and so, if removal of an edge leads to a high norm of the resulting control input as compared to the case when *all* of the neighbor are considered, then the bid is high.

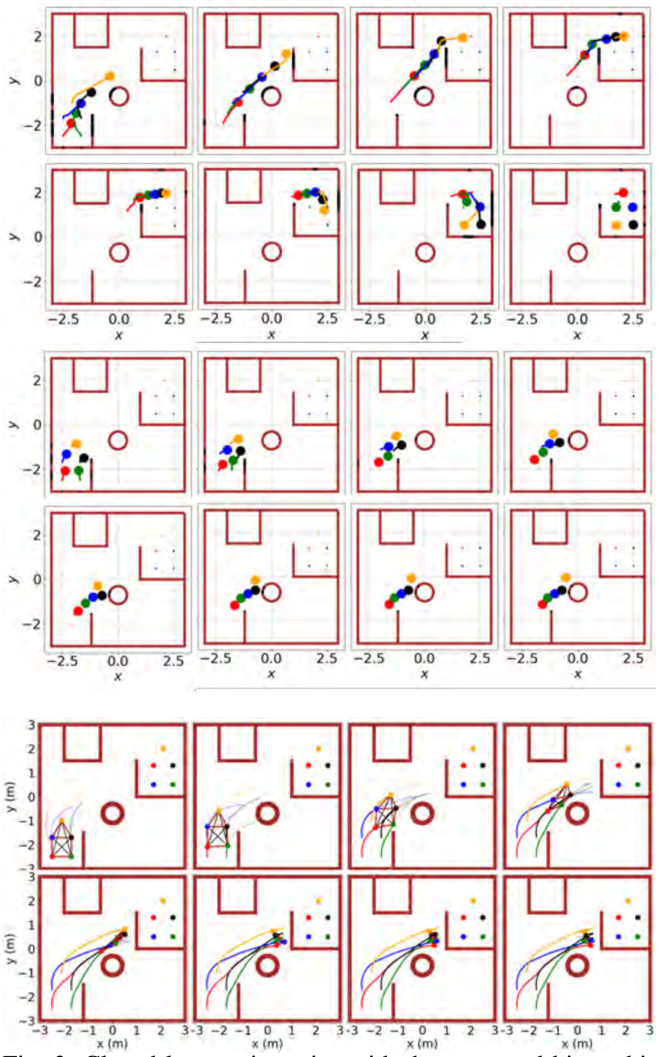
Fig. 3: Closed-loop trajectories with the proposed hierarchical controller (top figures), only CBF-QP controller (middle figures) and MPC (bottom figures) for apartment scenario.

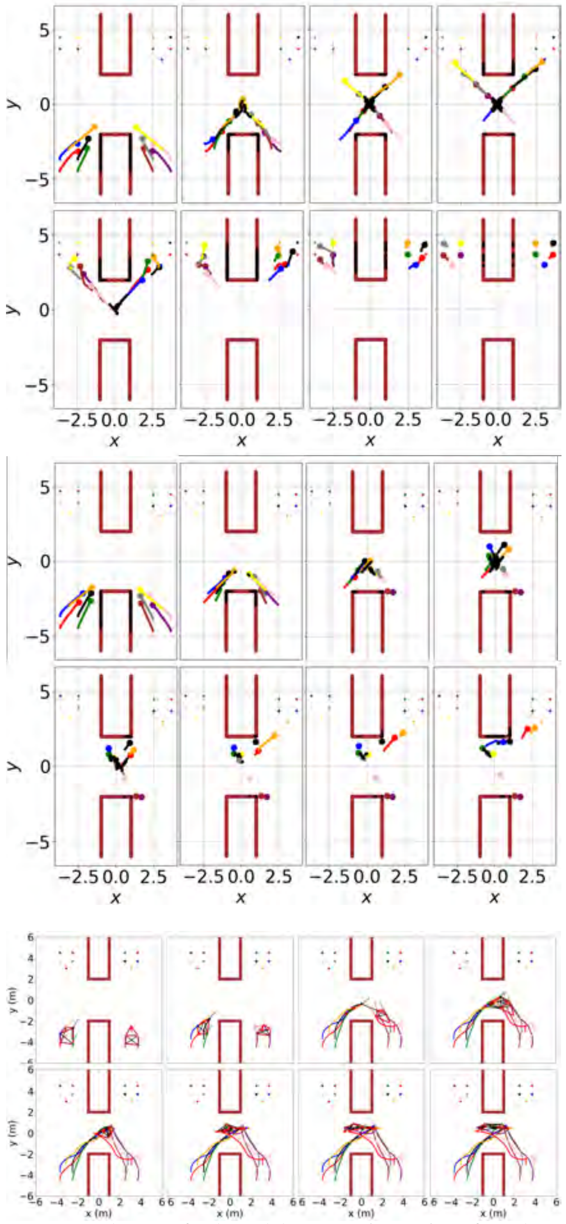
Fig. 4: Snapshots of closed-loop trajectories for the third numerical case study on crossing scenario with the proposed hierarchical controller (top figures), only CBF-QP controller (middle figures), and MPC (bottom figures).

Furthermore, if the set of safe neighbors is empty, then the bid value is set to -1. Here, a constant scaling factor K_i is added to the bid to reward the removal of an edge (i, j) if it benefits both agent i as well as agent j. If $j = g(\mathcal{S}_i)$ and $i = g(\mathcal{S}_j)$, i.e., both agents i, j choose to remove the edge (i, j), then the removal of this common edge benefits both the agents and it is rewarded by boosting the bid value.

IV. Evaluations

We perform several numerical and hardware case studies to illustrate the efficacy of the proposed method. For the numerical case studies, the main objective is to illustrate 1) deadlock resolution capabilities as opposed to a nominal CBF-QP without a high-level planner, and 2) scalability to many agents. Robot experiments illustrate the proposed method can be used on real robots. In each of the case studies, the simulation/experiment parameters are as follows: $R = 1.7, r = 0.1, dt = 0.1$. We use CBF-QP and a centralized model predictive control (MPC) as baselines for performance comparison. All the numerical experiments are performed on an Intel Xeon Platinum 8260 CPU @ 2400MHz with 4GB RAM. We use `Casadi` with `snopt` solver for MPC and `osqp` solver in `Python` for QPs.

Numerical case studies: We consider three scenarios for numerical case studies, namely, an apartment scenario consisting of 5 agents, a crossing scenario consisting of 10 agents, and a narrow passage scenario inspired from [35] consisting of 24 agents,

Apartment scenario: The first scenario represents a warehouse or an apartment scenario where the agents are required to reach another part of the workspace while remaining inside the boundary of the workspace and avoiding collisions with walls and other obstacles. The agents start in one corner of the workspace and propagate their way through the obstacle environment to reach their respective destinations. Figure 3 shows that the proposed framework (top figures) leads to the

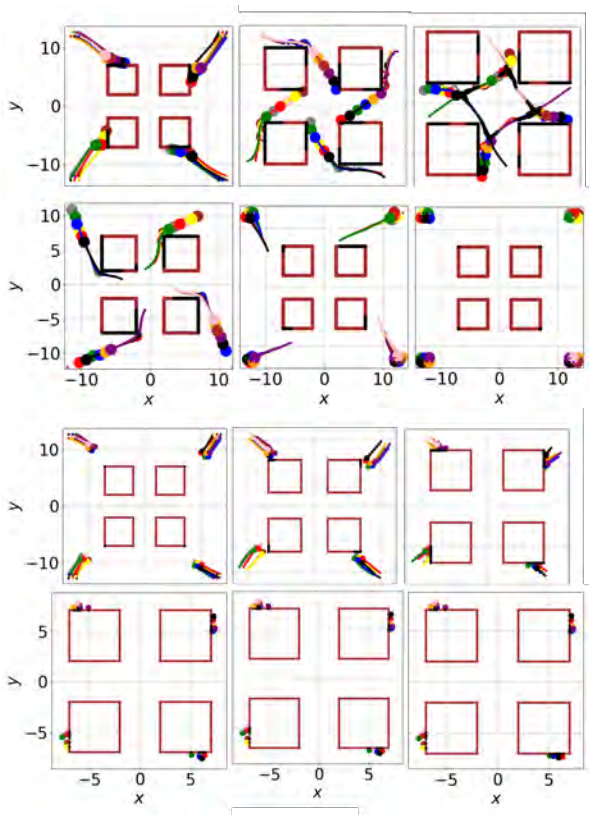

Fig. 5: Closed-loop trajectories with the proposed hierarchical controller (top figures) and just CBF-QP controller (bottom figures) for narrow passage scenario.

successful completion of the task via temporary goal assignment for evading deadlocks. Only CBF-QP (middle figures) cannot lead to the completion of the task and the agents get stuck behind an obstacle. Furthermore, as can be seen from the bottom figures in Figure 3, MPC (bottom figure) also fails to complete the task. Furthermore, each MPC step takes about 20 seconds of computational time, which makes it impossible to use for real-time implementation.

Crossing scenario: Next, we generate a crossing scenario to increase the inter-agent interaction between different groups. In this scenario, two groups of agents are initialized at the bottom left and bottom right corners of the workspace with their goals located on diagonally opposite corners. Figure 4 plots the closed-loop trajectories under the proposed hierarchical control framework as well as only the CBF-QP controller. Once again, the CBF-QP controller leads to a deadlock while the proposed method completes the task. MPC also fails to complete the task as can be seen from the bottom figures in Figure 4.

Narrow scenario: To illustrate the scalability of the proposed method, we consider the *narrow passage* case study in [35]. Here, we choose 6 agents at each corner to exchange their locations with the agents initialized at the diagonally opposite corner. There are four square obstacles situated in the center of the workspace that create narrow passages for these agents. In addition to the safety and goal-reaching requirements in [35], we also impose that the groups at each corner should maintain the connectivity of their underlying graph topology at all times. Figure 5 illustrates the performance of the proposed framework where the agents resolve various deadlocks on their way and safely navigate through a crowded narrow passage. When only the CBF-QP controller is used, the agents get stuck in a deadlock. In the interest of space, we do not include the plots for MPC here, but it also fails to complete the task and the agents get stuck in a deadlock (the results are reported in the video).

Computational time: We report the computational time for the proposed method, for the CBF-QP controller, and for MPC for the above three scenarios in Table I. While the proposed method and the CBF-QP method take 10 ms for each step for each agent, MPC takes a much longer time for the last two scenarios and hence, cannot be used for real-time implementation.

Hardware experiments: We validate our approach in hardware experiments on the first scenario from the numerical case studies, using four Turtlebot3 ground robots. We use an off-board ground computer for sending control commands to the robots via ROS, while the state information of the robots is obtained using a Vicon motion capture system. We perform experiments on the apartment scenario. Figure 6 shows the snapshots from the experiments under the proposed framework as well as the CBF-QP controller. The experiments confirm the numerical simulation results where the CBF-QP leads to deadlocks while the proposed method is capable of completing the tasks. Also, the additional steps in the proposed method do not lead to computational overhead and the proposed hierarchical control framework can be used for real-time deadlock resolution. A video of all the numerical and hardware experiments is available here: https://tinyurl.com/icra23GHF.

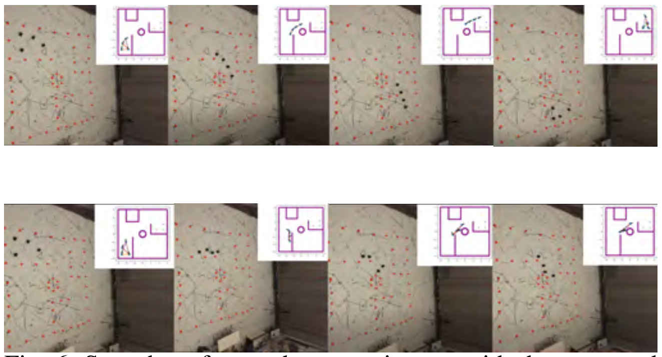

Fig. 6: Snapshots from robot experiments with the proposed hierarchical control (top figures) and only CBF-QP controller (bottom figures).

TABLE I: Per agent per step computation time.

Scenario	# Agents	Method	Average time (s)
Apartment	5	Proposed method	0.01
		CBF-QP	0.0092
		MPC ($T = 10$)	0.1
		MPC ($T = 20$)	0.3
Crossing	10	Proposed method	0.0051
		CBF-QP	0.0036
		MPC ($T = 10$)	0.3
		MPC ($T = 20$)	0.8
Narrow	24	Proposed method	0.0089
		CBF-QP	0.0059
		MPC ($T = 10$)	0.45
		MPC ($T = 20$)	1.9

V. Conclusions

In this paper, we presented a distributed hierarchical control framework for a multi-agent system that uses a high-level planner to assign temporary goals to agents when they get stuck in a deadlock. In addition, we proposed a novel QP-based bidding mechanism to break redundant edges in the graph topology that helps the multi-agent achieve its team objective. We illustrate the efficacy of the proposed method through extensive numerical case studies and also provide robot experiments that showcase the performance of the hierarchical controller as compared to CBF-QP which gets stuck in deadlocks and MPC which is very slow for real-time implementation.

One of the main limitations of the heuristic goal assignment method used as the high-level planner in the proposed framework is that it is not scalable to a large MAS operating in a dense obstacle environment. Future work involves using reinforcement learning-based methods for learning when and what temporary goal should be assigned to each agent for deadlock resolution. Furthermore, the current QP-based low-level controller uses very simple quadratic CBFs which might not be very effective in handling highly dense obstacle scenarios. Neural barrier certificates provide a viable alternative to handle complex scenarios and we will study how we can encode both safety and connectivity in a single certificate for handling complex environments and large-scale MAS.

References

[1] B. Li and H. Ma, "Double-deck multi-agent pickup and delivery: Multi-robot rearrangement in large-scale warehouses," *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3701–3708, 2023.

[2] P. R. Wurman, R. D'Andrea, and M. Mountz, "Coordinating hundreds of cooperative, autonomous vehicles in warehouses," *AI magazine*, vol. 29, no. 1, pp. 9–9, 2008.

[3] J. Dinneweth, A. Boubezoul, R. Mandiau, and S. Espié, "Multi-agent reinforcement learning for autonomous vehicles: a survey," *Autonomous Intelligent Systems*, vol. 2, no. 1, p. 27, 2022.

[4] Y. Tian, K. Liu, K. Ok, L. Tran, D. Allen, N. Roy, and J. P. How, "Search and rescue under the forest canopy using multiple uavs," *The International Journal of Robotics Research*, vol. 39, no. 10-11, pp. 1201–1221, 2020.

[5] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.

[6] F. Mehdifar, C. P. Bechlioulis, F. Hashemzadeh, and M. Baradarannia, "Prescribed performance distance-based formation control of multi-agent systems," *Automatica*, vol. 119, p. 109086, 2020.

[7] C. Godsil and G. F. Royle, *Algebraic graph theory*. Springer Science & Business Media, 2001, vol. 207.

[8] 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.

[9] B. S. Park and S. J. Yoo, "Connectivity-maintaining and collision-avoiding performance function approach for robust leader–follower formation control of multiple uncertain underactuated surface vessels," *Automatica*, vol. 127, p. 109501, 2021.

[10] M. M. Zavlanos and G. J. Pappas, "Potential fields for maintaining connectivity of mobile networks," *IEEE Transactions on robotics*, vol. 23, no. 4, pp. 812–816, 2007.

[11] L. Lindemann and D. V. Dimarogonas, "Control barrier functions for multi-agent systems under conflicting local signal temporal logic tasks," *IEEE control systems letters*, vol. 3, no. 3, pp. 757–762, 2019.

[12] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European control conference (ECC)*. IEEE, 2019, pp. 3420–3431.

[13] R. Cheng, M. J. Khojasteh, A. D. Ames, and J. W. Burdick, "Safe multi-agent interaction through robust control barrier functions with learned uncertainties," in *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE, 2020, pp. 777–783.

[14] Z. Qin, K. Zhang, Y. Chen, J. Chen, and C. Fan, "Learning safe multi-agent control with decentralized neural barrier certificates," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=P6_q1BRxY8Q

[15] H. Parwana, A. Mustafa, and D. Panagou, "Trust-based rate-tunable control barrier functions for non-cooperative multi-agent systems," in *2022 IEEE 61st Conference on Decision and Control (CDC)*. IEEE, 2022, pp. 2222–2229.

[16] W. Xiao, C. A. Belta, and C. G. Cassandras, "Sufficient conditions for feasibility of optimal control problems using control barrier functions," *Automatica*, vol. 135, p. 109960, 2022.

[17] M. F. Reis, A. P. Aguiar, and P. Tabuada, "Control barrier function-based quadratic programs introduce undesirable asymptotically stable equilibria," *IEEE Control Systems Letters*, vol. 5, no. 2, pp. 731–736, 2020.

[18] K. Garg, R. K. Cosner, U. Rosolia, A. D. Ames, and D. Panagou, "Multi-rate control design under input constraints via fixed-time barrier functions," *IEEE Control Systems Letters*, vol. 6, pp. 608–613, 2021.

[19] U. Rosolia and A. D. Ames, "Multi-rate control design leveraging control barrier functions and model predictive control policies," *IEEE Control Systems Letters*, vol. 5, no. 3, pp. 1007–1012, 2020.

[20] H. Ma, W. Hönig, L. Cohen, T. Uras, H. Xu, T. S. Kumar, N. Ayanian, and S. Koenig, "Overview: A hierarchical framework for plan generation and execution in multirobot systems," *IEEE Intelligent Systems*, vol. 32, no. 6, pp. 6–12, 2017.

[21] V. R. Desaraju and J. P. How, "Decentralized path planning for multi-agent teams with complex constraints," *Autonomous Robots*, vol. 32, pp. 385–403, 2012.

[22] J. Netter, G. P. Kontoudis, and K. G. Vamvoudakis, "Bounded rational rrt-qx: Multi-agent motion planning in dynamic human-like environments using cognitive hierarchy and q-learning," in *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, 2021, pp. 3597–3602.

[23] J. Chen, J. Li, C. Fan, and B. C. Williams, "Scalable and safe multi-agent motion planning with nonlinear dynamics and bounded disturbances," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 13, 2021, pp. 11 237–11 245.

[24] H. Ma, D. Harabor, P. J. Stuckey, J. Li, and S. Koenig, "Searching with consistent prioritization for multi-agent path finding," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 7643–7650.

[25] R. J. Afonso, M. R. Maximo, and R. K. Galvão, "Task allocation and trajectory planning for multiple agents in the presence of obstacle and connectivity constraints with mixed-integer linear programming," *International Journal of Robust and Nonlinear Control*, vol. 30, no. 14, pp. 5464–5491, 2020.

[26] L. Dai, Q. Cao, Y. Xia, and Y. Gao, "Distributed mpc for formation of multi-agent systems with collision avoidance and obstacle avoidance," *Journal of the Franklin Institute*, vol. 354, no. 4, pp. 2068–2085, 2017.

[27] C. Yu, A. Velu, E. Vinitsky, J. Gao, Y. Wang, A. Bayen, and Y. Wu, "The surprising effectiveness of ppo in cooperative multi-agent games," *Advances in Neural Information Processing Systems*, vol. 35, pp. 24 611–24 624, 2022.

[28] S. Zhang, K. Garg, and C. Fan, "Neural graph control barrier functions guided distributed collision-avoidance multi-agent control," in *7th Annual Conference on Robot Learning*, 2023.

[29] M. Mesbahi and M. Egerstedt, "Graph theoretic methods in multiagent networks," in *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, 2010.

[30] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2016.

[31] K. Garg, E. Arabi, and D. Panagou, "Fixed-time control under spatiotemporal and input constraints: A quadratic programming based approach," *Automatica*, vol. 141, p. 110314, 2022.

[32] X. Tan and D. V. Dimarogonas, "Distributed implementation of control barrier functions for multi-agent systems," *IEEE Control Systems Letters*, vol. 6, pp. 1879–1884, 2021.

[33] F. Blanchini, "Set invariance in control," *Automatica*, vol. 35, no. 11, pp. 1747–1767, 1999.

[34] S. M. Robinson, "Perturbed kuhn-tucker points and rates of convergence for a class of nonlinear-programming algorithms," *Mathematical programming*, vol. 7, pp. 1–16, 1974.

[35] J. Van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *2008 IEEE international conference on robotics and automation*. Ieee, 2008, pp. 1928–1935.

Appendix I
Proof of no deadlocks

Recall that a deadlock is defined as the existence of an interval $\tau \subset \mathbb{R}$, given as $\tau = [a, b)$ where $a, b \in \mathbb{R}$, where the average speed of the MAS $\frac{1}{N}\sum_i |\dot{p}_i|(t) = 0$ for all $t \in \tau$ and the Lebesgue measure of the interval $\mu(\tau) > 0$, where $\mu(\tau) = |\tau| := b - a$. Let $a \in \mathbb{R}$ be a time instant when the average speed of the MAS is zero, i.e., $\frac{1}{N}\sum_i |\dot{p}_i|(a) = 0$. Per the leader-assignment condition in Algorithm 1, the condition $\frac{1}{N}\sum_i \|\dot{p}_i(t)\| < u_{min}$ triggers a leader assignment step at time t. The leader is chosen as the agent that has the minimum value of $\frac{\|p_i - p_{gi}\|}{\|\dot{p}_i|_{p_{gi,temp}}\|}$. There are two cases possible: $\|\dot{p}_i|_{p_{gi,temp}}\| = 0$ for all $i \in \mathcal{V}$ or there exists at least one $i \in \mathcal{V}$ such that $\|\dot{p}_i|_{p_{gi,temp}}\| > 0$. The latter case leads to a non-zero average speed of the MAS, leading to $b = a$ in $\tau = [a, b)$, and thus, a deadlock cannot occur in this case. Next, we prove that

$$\|\dot{p}_i|_{p_{gi,temp}}\| = 0$$

for all $i \in \mathcal{V}$ is only possible at $t = 0$. Note that $\|\dot{p}_i|_{p_{gi,temp}}\| = 0$ for all $i \in \mathcal{V}$ is possible only if *all* the agents are occluded with obstacles and none of the agents have any free space to move. This is only possible at $a = 0$, i.e., the MAS is initialized in a location that is occluded by obstacles such that it cannot move. For $a > 0$, since the MAS can reach such a location where the average speed becomes lower than the leader-assignment threshold, there exists at least one agent that has *free* space around it to move and hence, $\|\dot{p}_i|_{p_{gi,temp}}\| = 0$ is not possible for all $i \in \mathcal{V}$ for any $a > 0$ and hence, the MAS cannot get stuck in a deadlock.