Multi-UAV Deployment in Obstacle-Cluttered Environments with LOS Connectivity

Yuda Chen[1], Shuaikang Wang[1], Jie Li[2] and Meng Guo[1]

***Abstract*— A reliable communication network is essential for multiple UAVs operating within obstacle-cluttered environments, where limited communication due to obstructions often occurs. A common solution is to deploy intermediate UAVs to relay information via a multi-hop network, which introduces two challenges: (i) how to design the structure of multi-hop networks; and (ii) how to maintain connectivity during collaborative motion. To this end, this work first proposes an efficient constrained search method based on the minimum-edge RRT* algorithm, to find a spanning-tree topology that requires a less number of UAVs for the deployment task. Then, to achieve this deployment, a distributed model predictive control strategy is proposed for the online motion coordination. It explicitly incorporates not only the inter-UAV and UAV-obstacle distance constraints, but also the line-of-sight (LOS) connectivity constraint. These constraints are well-known to be nonlinear and often tackled by various approximations. In contrast, this work provides a theoretical guarantee that all agent trajectories are ensured to be collision-free with a team-wise LOS connectivity at all time. Numerous simulations are performed in 3D valley-like environments, while hardware experiments validate its dynamic adaptation when the deployment position changes online.**

I. Introduction

Fleets of unmanned aerial vehicles (UAVs) have been widely deployed to accomplish collaborative missions for exploration, inspection and rescue [1], [2], [3], [4]. In many applications, the UAVs need to keep a reliable communication with a ground station to transfer data and monitor status of the fleet [5]. This however can be challenging due to the limited communication range and obstructions from obstacles. An effective solution as also adopted in [5], [6] is to deploy more UAVs as relay nodes to form a multi-hop communication network. Two key challenges arise with this approach, i.e., (i) how the multi-hop network should be designed, including the topology and the positions of each node; (ii) how to control the UAV fleet to form this network, while maintaining the team-wise connectivity and avoiding collisions in an obstacle-cluttered environment.

A. Related Work

Regarding the design of communication network, linear graphs are proposed in [7], [8] for exploration and surveillance tasks in complex environments. Such structure is simple to design but rather limited, as it only allows the tailing agent to perform a task. Another common topology is called spanning tree as adopted in [9], [10], [11], where all agents as leaves of this tree can execute multiple tasks simultaneously. However, these works often assume a redundant number of agents without estimating or minimizing the number of agents that is required to accomplish the tasks. Furthermore, the communication model considered in this work is often simplified to be range-based, i.e., any two agents within a certain distance can exchange information. However, obstructions due to obstacles can severely degrade the communication quality and the transmission rate. Thus, as also motivated in [12], [13], [14], the line-of-sight (LOS) model is a more practical choice for approximating the inter-agent communication in obstacle-cluttered environments. More concretely, the work in [13] proposes the notion of visibility-weighted minimum spanning tree (MST), which leads to a more robust path compared to [10] under LOS constraints. Lastly, more realistic communication models for multi-agent system are proposed in [5], [15], [4], which take into account reflection and diffraction effects with a higher-order model. Whereas essential for certain applications, such models are often highly nonlinear and non-analytical, yielding them difficult to be incorporated into the collaborative motion strategy. Thus, despite of being a conservative approximation, the considered LOS constraint can ensure a high quality of inter-agent communication.

On the other hand, it is a non-trivial problem to maintain the designed communication network during collaborative motion, especially when avoiding collisions between agents and obstacles. Numerous methods have been proposed to address these constraints. For instances, artificial forces are designed in [16], [17], e.g., repulsive forces for obstacles and other agents; and attractive forces for the targets and connected agents. Lyapunov-based nonlinear controllers are proposed in [18], [19] mostly for free-spaces. Control barrier

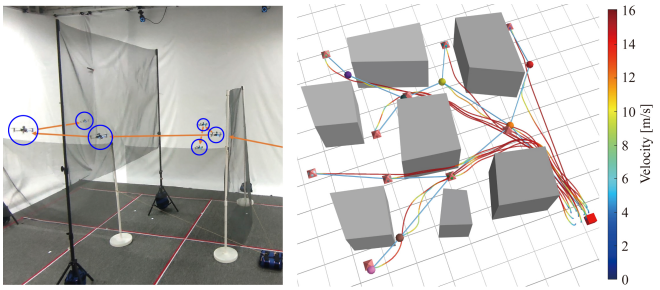

Fig. 1. **Left:** Experiment of 6 quadrotors cooperatively searching two targets. **Right:** Simulation of 15 agents (colored spheres) are deployed from the ground station (red cubic) to reach 10 targets (pink diamonds). The optimized network topology (blue lines) are kept during motion at all time.

The authors are with [1]the Department of Mechanics and Engineering Science, College of Engineering, Peking University, Beijing 100871, China; and [2]National University of Defense Technology, Hunan 410073, China. This work was supported by the National Natural Science Foundation of China (NSFC) under grants 62203017, T2121002, U2241214; 2030-Key Project under Grant 2020AAA0108200. Corresponding author: Meng Guo, meng.guo@pku.edu.cn.

certificates are proposed in [10], [20] for both constraints of connectivity maintenance and collision avoidance. Despite of their intuitiveness, these methods often lack theoretical guarantee on the safety when encountering control and state constraints. Furthermore, the LOS restriction is particularly difficult to incorporate due to its non-convexity. A geometric approach is proposed in [13] based on the Loyd algorithm, which drives the agents towards the intersection of neighboring communication zones. It however neglects the dynamic constraints such as limited velocity and acceleration. Optimization-based methods such as [21] directly formulate these constraints as mixed integer linear programs (MILP), of which the complexity increases drastically with the number of agents and obstacles, yielding them impractical for real-time applications.

B. Our Method

To tackle these limitations, this work proposes an integrated framework of network formulation and motion coordination for deploying UAV fleets in obstacle-cluttered environments. It generates an embedded network topology as spanning trees via the minimum-edge RRT* algorithm, subject to the LOS restrictions. During online execution, a collaborative motion coordination strategy is proposed based on the distributed model predictive control (MPC) that maintains the designed network topology and generates collision-free trajectories. In particular, the LOS constraints as well as the requirement of collision avoidance are explicitly formulated as linear constraints by restricting the UAVs in constructed convex polyhedra. It has been shown that the MPC can be solved online efficiently, with a theoretical guarantee on its feasibility. Effectiveness of the proposed framework is validated in numerous simulations and hardware experiments. The main contributions are two-fold:

- A planning method is introduced to construct a spanning-tree-like communication topology with a minimum number of edges. Different from the common self-organizing topology like [8], [10], [13], the proposes algorithm of topology designing decreases the number of agents required for the task.
- The proposed motion control strategy based on distributed MPCs theoretically guarantees the network connectivity and collision-avoidance. Compared with MILP-based methods in [21] which handles LOS restrictions, our method reduces the planning time by 90% and exhibits online adaptation towards dynamic targets.

II. Problem Description

A. Robot Model

As shown in Fig. 1, consider a fleet of $N > 0$ UAVs operating within the 3D workspace $\mathcal{W} \subset \mathbb{R}^3$. Each agent follows the standard double-integrator model $\dot{x}^i(t) \triangleq [v^i(t), u^i(t)]$, where $x^i(t) \triangleq [p^i(t),\, v^i(t)]$ is the state of agent i; $p^i(t), v^i(t), u^i(t) \in \mathbb{R}^3$ are the position, velocity and control input of agent i at time $t \geqslant 0$, respectively, $\forall i \in \mathcal{N} \triangleq \{1, \cdots, N\}$. Moreover, the state and control input of each agent $i \in \mathcal{N}$ are restricted by:

$$\|\Theta_a u^i(t)\|_2 \leqslant a_{\max},\ \|\Theta_v v^i(t)\|_2 \leqslant v_{\max},$$

where Θ_v, Θ_a are positive-definite matrices, and $v_{\max}$, $a_{\max} > 0$ are the maximum velocity and acceleration, respectively. Note that the total number of agents N is *not* fixed, rather designed as part of this problem.

B. Communication Constraints

Any pair of agents $i, j \in \mathcal{N}$ can direct communicate as *neighbors* if both the following two conditions hold: (i) their relative distance is within the communication range $d_c > 0$, i.e., $\|p^i(t) - p^j(t)\|_2 \leqslant d_c$; (ii) the line-of-sight (LOS) between the agents cannot be obstructed by obstacles, i.e., $\mathbf{Line}(p^i(t),\, p^j(t)) \cap \mathcal{O} = \varnothing$, where $\mathbf{Line}(p^i(t),\, p^j(t))$ is the line segment from position $p^i(t)$ to position $p^j(t)$; and $\mathcal{O} \subset \mathcal{W}$ is the volume occupied by a set of convex-shaped static obstacles, each of which is defined as the convex hull of a set of known vertices. Lastly, there is a ground station at position $p_g \in \mathcal{W}$, which needs to communicate with any agent as described above. With a slight abuse of notation, the ground station is denoted by agent 0 and $\widetilde{\mathcal{N}} \triangleq \{0, 1, \cdots, N\}$.

Consequently, an undirected and time-varying communication network $G(t) \triangleq (\widetilde{\mathcal{N}},\, E(t))$ can be constructed given the neighboring relation above, where $E(t) \subset \widetilde{\mathcal{N}} \times \widetilde{\mathcal{N}}$ for time $t > 0$. In other words, $(i,\, j) \in E(t)$ if the position of agents i and j at time t satisfies the two conditions above. The network $G(t)$ is called *connected* at time t if there exists a path between any two nodes in $G(t)$, i.e., $G(t) \in \mathcal{C}$, where $\mathcal{C}$ denotes the set of all connected graphs

C. Collision Avoidance

Each agent $i \in \mathcal{N}$ occupies a spherical region with radius $r_a > 0$, denoted by $\mathcal{B}^i(t) \triangleq \{p \in \mathcal{W} \,|\, \|p - p^i(t)\|_2 \leqslant r_a\}$. To avoid collisions, the spherical regions of any two agents $i, j \in \mathcal{N}$ must satisfy the condition $\mathcal{B}^i(t) \cap \mathcal{B}^j(t) = \varnothing$. Furthermore, each agent $i \in \mathcal{N}$ must avoid colliding with static obstacles at time $t > 0$, meaning that the condition $\mathcal{B}^i(t) \cap \mathcal{O} = \varnothing$ must be satisfied.

D. Problem Statement

Lastly, there are $M > 0$ target positions within the workspace $\mathcal{W}$, denoted by $\mathcal{P}_{\text{tg}} \triangleq \{p_{\text{tg},m},\, m \in \mathcal{M}\} \subset \mathcal{W} \backslash \mathcal{O}$, where $\mathcal{M} \triangleq \{1, \cdots, M\}$. Each target position is associated with potential tasks to be performed by the agents. Each agent $i \in \mathcal{N}$ starts from its initial position p_0^i, which is collision-free and located in close proximity to the ground station. Additionally, the initial communication network $G(0)$ is assumed to be connected. Thus, the problem is formulated as a constrained optimization problem:

$$\begin{aligned}
& \mathbf{min}_{\{u^i(t),\, T\}}\ N \\
\mathbf{s.t.}\ & \dot{x}^i(t) = [v^i(t), u^i(t)],\ \forall i,\ \forall t; \\
& \|\Theta_a u^i(t)\|_2 \leqslant a_{\max},\ \|\Theta_v v^i(t)\|_2 \leqslant v_{\max},\ \forall i,\ \forall t; \\
& \mathcal{B}^i(t) \cap \mathcal{B}^j(t) = \varnothing,\ \mathcal{B}^i(t) \cap \mathcal{O} = \varnothing,\ \forall i,\ \forall j,\ \forall t; \\
& p^{i_m}(T) = p_{\text{tg},m},\ \forall m;\ G(t) \in \mathcal{C},\ \forall t;
\end{aligned} \tag{1}$$

where $t \in [0, T]$; $T > 0$ is the termination time when all target positions are reached; $i_m \in \mathcal{N}$ is the agent that reaches target $m \in \mathcal{M}$; the objective is to minimize the total number of agents required for the mission; and the constraints are the aforementioned dynamic model, communication connectivity, and safety guarantees.

III. Proposed Solution

The proposed solution consists of two layers. First, the communication network is formulated given the workspace and target locations to minimize the team size. Second, a collaborative control strategy is designed to drive the agents towards the target positions while maintaining the desired topology and avoiding collisions.

A. Design of the Communication Network

1) Minimum Edge RRT:* To begin with, a modification of the informed RRT*, as introduced in [22], is proposed for the design of the communication network.. The proposed method, referred to as `MiniEdgeRRT`*, aims to find a set of paths from a starting point to a set of target points, minimizing the number of edges along each path. Specifically, the following three modifications are implemented.

Goal Region: The goal region is defined as the union of vertices surrounding the target points, allowing for the concurrent identification of paths toward multiple targets.

Cost: The cost between neighboring vertices in the tree is $\mathbf{c}(\hat{\nu}, \nu) \triangleq \|\hat{\nu} - \nu\|_2 + \kappa$, where $\hat{\nu}$ represents the parent node of ν and $\kappa > 0$ is a large penalty added to the cost of each edge. The total cost of a path Γ is the accumulated cost of all edges alone the path, denoted by $\texttt{Cost}(\Gamma)$. Thus, paths with fewer edges are always preferred.

Sample: New samples ν_{new} are generated within a distance d_c of an existing sample, i.e., $\|\nu_{\text{new}} - \nu_{\text{nearest}}\| \leqslant d_c$, ensuring that any two neighboring vertices satisfy the communication-range constraint.

Remark 1. Based on the analysis in [22], it can be shown that a path with fewer edges can be found for each target point in $\mathcal{P}_{\text{tg}}$ given the starting vertex. ■

2) Network Formulation: The paths planned by `MiniEdgeRRT`* are initially treated as independent paths to the targets, which can be further improved by increasing the number of *shared* edges. First of all, a spanning-tree topology is defined as follows.

Definition 1. An *embedded* spanning tree is defined as a set of 3-tuples: $\mathcal{T} \triangleq \{\nu_0, \nu_1, \cdots\}$, where each vertex $\nu_i \triangleq (i, p_i, I_i)$, with index $i \in \mathbb{Z}$, position $p_i \in \mathcal{W} \backslash \mathcal{O}$ and parent index $I_i \in \mathbb{Z}$ of vertex ν_i. Note that $\nu_0 \triangleq (0, p_0, \varnothing)$ is the root, corresponding to the ground station. ■

As summarized in Alg. 1, an iterative algorithm is proposed to incrementally construct the spanning tree by greedily adding target points based on their minimum-cost path. Specifically, the spanning tree $\mathcal{T}$ is initialized with the newly-added vertices $\mathcal{V}_{\text{new}}$, the set of indices $\mathcal{I}$ associated with the target points, and the current-best path Γ_c^i for each

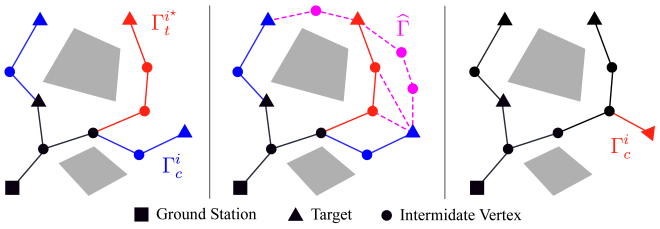

Fig. 2. Illustration of the iterative process in Alg. 1. **Left**: The target with the shortest path $\Gamma_t^{i^\star}$ (in red) is added to the tree, while the remaining targets are shown with their current-best paths Γ_c^i (in blue). **Middle**: Minimum-edge paths from other targets to the newly added points $\mathcal{V}_{\text{new}}$ are generated (in pink). **Right**: The updated topology after a new branch (in red) is added.

Algorithm 1: `ComNet()`

Input : p_0, $\mathcal{P}_{\text{tg}}$, $\mathcal{O}$
Output: $\{\Gamma^i\}$

1 $\mathcal{T} \leftarrow \{\nu_0\}$; $\mathcal{V}_{\text{new}} \leftarrow \{\nu_0\}$; $\mathcal{I} \leftarrow \mathcal{M}$; $\Gamma_c^i \leftarrow \varnothing$, $\forall i \in \mathcal{I}$;
2 **while** $\mathcal{I} \neq \varnothing$ **do**
3 **for** i **in** $\mathcal{I}$ parallel **do**
4 $\widehat{\Gamma} \leftarrow \texttt{MiniEdgeRRT}^\star(p_{\text{tg},i}, \mathcal{V}_{\text{new}}, \mathcal{O})$;
5 $\Gamma^\star \leftarrow \text{argmin}_{\Gamma \in \widehat{\Gamma}} \{\texttt{Cost}(\Gamma)\}$;
6 $\Gamma_c^i \leftarrow \texttt{Compare}(\Gamma^\star, \Gamma_c^i)$;
7 $i^\star \leftarrow \text{argmin}_{i \in \mathcal{I}} \{\texttt{Cost}(\Gamma_c^i)\}$;
8 $\mathcal{I} \leftarrow \mathcal{I} \backslash i^\star$;
9 $\mathcal{V}_{\text{new}} \leftarrow$ Extract vertices from $\Gamma_t^{i^\star}$;
10 $\mathcal{T} \leftarrow \mathcal{T} \cup \mathcal{V}_{\text{new}}$;
11 Compute the final Γ^i, $\forall i \in \mathcal{I}$ within $\mathcal{T}$;

target $i \in \mathcal{I}$, as shown in Line 1. During each iteration, the set of paths $\widehat{\Gamma}$ from each target $i \in \mathcal{I}$ to every vertices in $\mathcal{V}_{\text{new}}$ is derived via the proposed `MiniEdgeRRT`* in Line 4. The path with the minimum cost $\Gamma^\star$ is selected from the $\widehat{\Gamma}$ in Line 5. Subsequently, the current-best path Γ_c^i for target i is updated by comparing it with $\Gamma^\star$ in Line 6. The index of the target point with the minimum cost $i^\star \triangleq \text{argmin}_i \{\Gamma_t^{i^\star}\}$ is determined in Line 7. Then, the associated target $i^\star$ is removed from the set of potential targets $\mathcal{I}$, and the nodes extracted from $\Gamma_t^{i^\star}$ are added to the spanning tree as $\mathcal{V}_{\text{new}}$ in Lines 9 and 10, with their preceding nodes as parents. This process is repeated until the set $\mathcal{I}$ is empty, indicating that all targets have been connected to the tree $\mathcal{T}$. As illustrated in Fig. 2, the number of edges in this topology is significantly fewer than in a standard minimum spanning tree.

The computational complexity of Alg. 1 is similar to that of the informed RRT* in [22], i.e., $n_{\text{sample}} \log(n_{\text{sample}})$ where n_{sample} is the number of sampled vertices. In our implementation, the planning time for each call to the subroutine `MiniEdgeRRT`* is limited by a given duration $t_d > 0$. The computations for each target is performed in parallel, so the overall complexity is bounded by $\mathcal{O}\big(M n_{\text{sample}} \log(n_{\text{sample}})\big)$. Note that the proposed algorithm does not generate the optimal spanning tree, as this requires an exhaustive search over all possible target point sequences.

Given the spanning tree $\mathcal{T}$ derived above, the agents are divided into two groups $\mathcal{N} \triangleq \mathcal{N}_s \cup \mathcal{N}_c$. The agents in $\mathcal{N}_s$ are deployed at the target points as *searchers* for task execution,

while the agents in $\mathcal{N}_c$ are assigned to the intermediate vertices as *connectors*. Thus, the total number of agents required to form the tree $\mathcal{T}$ is given by $N = |\mathcal{T}| - 1$ with $N_s = M$ searchers and $N_c = N - M$ connectors. Furthermore, the reference path for each searcher $i \in \mathcal{N}_s$ is denoted by $\Gamma^i \triangleq \{p_g, \cdots, p_{\text{tg},n_i}\}$, representing a sequence of waypoints along the path in $\mathcal{T}$ from the root to the assigned target p_{tg,n_i}. In contrast, the connectors do not have reference paths and serve solely as intermediate relays.

Remark 2. Related works propose self-organizing typologies such as line graphs in [8] or spanning trees in [10], [13]. Although they are applied to dynamic tasks in various environments, they do not consider the number of agents required as part of the optimization criteria. ■

B. Collaborative Trajectory Planning via Distributed MPC

Given the desired communication topology $\mathcal{T}$, this section introduce a distributed control strategy to maintain this topology $\mathcal{T}$ during collaborative motion, while ensuring collision avoidance. The approach reformulates the problem with the distributed MPC framework [23], where the geometric and dynamic constraints are approximated and expressed as linear or quadratic inequalities.

To begin with, denote by $h > 0$ the sampling time and $K \in \mathbb{Z}^+$ the planning horizon. The planned state and control input of agent i at time $t + kh$ are denoted by $x_k^i(t) \triangleq [p_k^i(t), v_k^i(t)]$ and $u_k^i(t)$, respectively, $\forall k \in \mathcal{K} \triangleq \{1, \cdots, K\}$. Furthermore, the planned trajectory of agent i at the time t is defined as $\mathcal{P}^i \triangleq \{p_1^i, p_2^i, \cdots, p_K^i\}$. For brevity, in the sequel, the index "(t)" can be omitted if no ambiguity causes. Moreover, the so-called *predetermined* trajectory is denoted by $\overline{\mathcal{P}}^i \triangleq \{\overline{p}_1^i, \overline{p}_2^i, \cdots, \overline{p}_K^i\}$, where $\overline{p}_k^i(t) \triangleq p_{k+1}^i(t-h)$, $\forall k \in \{1, 2, \cdots, K-1\}$ and $\overline{p}_K^i(t) \triangleq p_K^i(t-h)$. Note that the planned trajectory $\mathcal{P}^i$ should comply with the double integrator model, i.e.,

$$x_k^i \triangleq \mathbf{A} x_{k-1}^i + \mathbf{B} u_{k-1}^i, \forall k \in \mathcal{K}, \tag{2}$$

where $\mathbf{A} \triangleq \begin{bmatrix} \mathbf{I}_3 & h\mathbf{I}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix}$, $\mathbf{B} \triangleq \begin{bmatrix} \frac{h^2}{2}\mathbf{I}_3 \\ h\mathbf{I}_3 \end{bmatrix}$; and the constraints for velocities and inputs remain the same as follows:

$$\|\Theta_a u_{k-1}^i\|_2 \leqslant a_{\max}, \ \|\Theta_v v_k^i\|_2 \leqslant v_{\max}, \forall k \in \mathcal{K}. \tag{3}$$

1) Collision Avoidance: Collision avoidance among the agents can be reformulated via the modified buffered Voronoi cell (MBVC) as proposed in our previous work [24], i.e.,

$$a_k^{ij\,\mathrm{T}} p_k^i \geqslant b_k^{ij}, \ \forall j \neq i, \ \forall k \in \mathcal{K}, \tag{4}$$

where the coefficients a_k^{ij} and b_k^{ij} are given by:

$$a_k^{ij} \triangleq \frac{\overline{p}_k^i - \overline{p}_k^j}{\|\overline{p}_k^i - \overline{p}_k^j\|_2}, \quad b_k^{ij} \triangleq a_k^{ij\,\mathrm{T}} \frac{\overline{p}_k^i + \overline{p}_k^j}{2} + \frac{r'_{\min}}{2},$$

and $r'_{\min} \triangleq \sqrt{4{r_a}^2 + h^2 v_{\max}^2}$. On the other hand, the collision avoidance between agent $i \in \mathcal{N}$ and all obstacles is realized by restricting its planned trajectory in a safe corridor formed by convex polyhedra. This corridor serves as the boundary

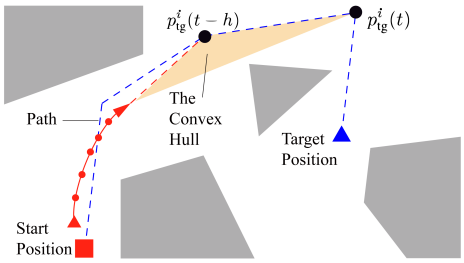

Fig. 3. Illustration of how the intermediate target point $p_{\text{tg}}^i(t)$ is determined for a searcher agent $i \in \mathcal{N}_s$ given its path Γ^i.

that separates the planned positions p_k^i and the inflated obstacles given by $\widehat{\mathcal{O}}(r_a) \triangleq \{p \in \mathcal{W} \mid \|p - p_o\|_2 \leqslant r_a, \, p_o \in \mathcal{O}\}$. Specifically, the constraint over the planned trajectory is stated as the following linear inequality:

$$a_k^{i,o\,\mathrm{T}} p_k^i \geqslant b_k^{i,o}, \ \forall k \in \mathcal{K}, \tag{5}$$

where $a_k^{i,o}$ and $b_k^{i,o}$ represent the linear boundary for agent i at step k, w.r.t. obstacle $o \in \mathcal{I}_o$; and $\mathcal{I}_o$ is the collection of obstacle indices. The detailed derivation is omitted here due to limited space and refer the readers to [25].

Moreover, as illustrated in Fig. 3, an intermediate target point denoted by p_{tg}^i is computed dynamically for each searcher $i \in \mathcal{N}_s$. Initially, $p_{\text{tg}}^i = p_g$. For time $t > 0$, $p_{\text{tg}}^i(t)$ is updated online as the closest vertex in its path Γ^i that

$$\textbf{Conv}\left(\left\{ \Gamma^i \left(p_{\text{tg}}^i(t-h), p_{\text{tg}}^i(t) \right), \, \overline{p}_K^i(t) \right\} \right) \cap \widehat{\mathcal{O}}(r_a) = \varnothing,$$

where $\Gamma^i(p_1, p_2)$ denotes the sequence of nodes on Γ^i from position p_1 to p_2; and $\textbf{Conv}(P) \triangleq \{\sum_{j=1}^J \theta_j p_j \mid \sum_j \theta_j = 1, \, p_j \in P, \, \theta_j \geqslant 0, \, j = 1, \cdots, J\}$ is the convex hull formed by points within the set P; $p_{\text{tg}}^i(t-h)$ is the intermediate target at time $t-h$. For connector $i \in \mathcal{N}_c$, its intermediate target point is determined by $\overline{p}_K^i$. Then, the position of the predetermined trajectory at the $(K+1)$-th step is chosen as $\overline{p}_{K+1}^i \triangleq p_{\text{tg}}^i$.

2) Connectivity Maintenance: Maintaining the desired topology during motion is the most challenging constraint to deal with. For each pair of neighboring agents in $\mathcal{T}$, this constraint is decomposed into two sub-constraints: (i) the relative distance between them should be less than d_c; (ii) the LOS connecting them should not be obstructed by obstacles.

The first sub-constraint is reformulated as a condition requiring each agent stays within a sphere, of which the center depends on its relative distance to the neighbor. Namely, the position of agent i at step k is constrained by:

$$\left\| p_k^i - c_k^{ij} \right\|_2 \leqslant d_c/2, \ \forall j \in \mathcal{N}^i, \ \forall k \in \mathcal{K}, \tag{6}$$

where $\mathcal{N}^i \subseteq \mathcal{N}$ is the set of parent and child agents of agent i within the topology $\mathcal{T}$; and the center $c_k^{ij} \in \mathcal{W}$ is determined based on the predetermined trajectory of agents i and j by considering three different cases:

(i) If $\|\overline{p}_k^i - \overline{p}_k^j\|_2 > d_w$ with $d_w > 0$ being a chosen distance slightly less than d_c, it indicates that agents i and j are far away and $c_k^{ij} = (\overline{p}_k^i + \overline{p}_k^j)/2$;

(ii) If $\|p - c^i_{k,\text{center}}\| < d_w \ \forall p \in \{\overline{p}^i_k, \overline{p}^j_k, \overline{p}^i_{k+1}, \overline{p}^j_{k+1}\}$, it indicates that agents i and j are in close proximity and $c^{ij}_k = c^i_{k,\text{center}}$ with $c^i_{k,\text{center}} \triangleq (\overline{p}^i_k + \overline{p}^j_k + \overline{p}^i_{k+1} + \overline{p}^j_{k+1})/4$;

(iii) Otherwise, they are in medium distance and $c^{ij}_k = \eta^{ij}(\overline{p}^i_k + \overline{p}^j_k)/2 + (1-\eta^{ij})c^i_{k,\text{center}}$, where $\eta^{ij} \in [0,\, 1]$ is solved by the following optimization problem:

$$\begin{aligned} &\eta^{ij} = \min_{0 \leqslant \eta \leqslant 1} \ \eta, \\ \textbf{s.t.} \ &\|\overline{p}^i_k - \overline{c}^{ij}_k\|_2 \leqslant d_w/2, \ \|\overline{p}^j_k - \overline{c}^{ij}_k\|_2 \leqslant d_w/2, \\ &\overline{c}^{ij}_k = \eta(\overline{p}^i_k + \overline{p}^j_k)/2 + (1-\eta)c^i_{k,\text{center}}, \end{aligned} \tag{7}$$

which can be solved by the standard algorithm of bisection method [26] with the search interval $[0, 1]$.

Secondly, for the LOS sub-constraint, linear constraints define a polyhedral safe zone for p^i_k and p^j_k. Before presenting these constraints, two intermediate points are computed:

$$\overline{p}^i_{k,\text{next}} \triangleq \overline{p}^i_k + \xi^\star(\overline{p}^i_{k+1} - \overline{p}^i_k); \ \overline{p}^j_{k,\text{next}} \triangleq \overline{p}^j_k + \xi^\star(\overline{p}^j_{k+1} - \overline{p}^j_k),$$

where $\xi^\star$ is the minimum $\xi \in [0,\, 1]$ such that:

$$\begin{aligned} \textbf{Conv}\big(\{\overline{p}^i_k,\, \overline{p}^j_k,\, \xi\overline{p}^i_{k+1} + (1-\xi)\overline{p}^i_k,\, \xi\overline{p}^j_{k+1} + (1-\xi)\overline{p}^j_k\}\big) \\ \cap \ \widehat{\mathcal{O}}(d_m) = \varnothing, \end{aligned}$$

where $d_m > 0$ is a small margin. Similar to (7), the optimal $\xi^\star$ can be computed by the bisection method. Given $\overline{p}^i_{k,\text{next}}$ and $\overline{p}^j_{k,\text{next}}$, the linear constraints that separate the free space $\textbf{Conv}(\mathcal{P}^{\text{free}})$ for $\mathcal{P}^{\text{free}} \triangleq \{\overline{p}^i_k, \overline{p}^j_k, \overline{p}^i_{k,\text{next}}, \overline{p}^j_{k,\text{next}}\}$ and obstacle $o \in \mathcal{I}_o$ are obtained as follows:

$$\begin{aligned} &\max_{\{\widehat{a}^{ij}_k, \widehat{b}^{ij}_k, \delta\}} \ \delta, \\ \textbf{s.t.} \ &\widehat{a}^{ij,o\mathrm{T}}_k \ p \geqslant \delta + \widehat{b}^{ij,o}_k, \ \forall p \in \mathcal{P}^{\text{free}}; \\ &\widehat{a}^{ij,o\mathrm{T}}_k p \leqslant \widehat{b}^{ij,o}_k, \ \forall p \in \mathcal{P}^o; \\ &\|\widehat{a}^{ij,o}_k\|_2 = 1; \end{aligned} \tag{8}$$

where $\widehat{a}^{ij,o}_k \in \mathbb{R}^3$, $\widehat{b}^{ij,o}_k \in \mathbb{R}$ and $\delta \in \mathbb{R}$ are the optimization variables; $\mathcal{P}^o$ are the vertices of obstacle $o \in \mathcal{I}_o$ after being inflated by d_m. Since $\textbf{Conv}(\mathcal{P}^{\text{free}}) \cap \widehat{\mathcal{O}}(d_m) = \varnothing$, it follows from the separating hyperplane theorem that $\delta > 0$. Thus, the optimization in (8) can be solved as a quadratic program similar to [25]. Given these coefficients $\widehat{a}^{ij,o}_k$ and $\widehat{b}^{ij,o}_k$, the LOS constraint for agents i and j is restated as:

$$\widehat{a}^{ij,o\mathrm{T}}_k p^i_k \geqslant \widehat{b}^{ij,o}_k, \ \forall j \in \mathcal{N}^i, \ \forall k \in \mathcal{K}, \tag{9}$$

which is the separating hyperplane for obstacle $o \in \mathcal{I}_o$.

C. Overall Algorithm

The objective of each searcher $i \in \mathcal{N}_s$ is to minimize the balanced cost, which is the distance to the target and velocity:

$$C^i \triangleq \frac{1}{2} Q_K \|p^i_K - p^i_{\text{tg}}\|^2_2 + \frac{1}{2} \sum_{k=1}^{K-1} Q_k \|p^i_{k+1} - p^i_k\|^2_2, \tag{10}$$

where $Q_k > 0$ are given positive definite parameters, $\forall k \in \mathcal{K}$. Meanwhile, each connector $i \in \mathcal{N}_c$ minimizes its distance to an intermediate point $\widetilde{p}^i_k$ determined by its neighbors, i.e.,

$$C^i \triangleq \frac{1}{2} \sum_{k=1}^{K} \|p^i_k - \widetilde{p}^i_k\|^2_2; \quad \text{and} \quad \widetilde{p}^i_k \triangleq \sum_{j \in \mathcal{N}^i} \alpha_j \, \overline{p}^j_k, \tag{11}$$

where $\alpha_j \geqslant 0$ are the weighting constants with $\sum_{j \in \mathcal{N}^i} \alpha_j \triangleq 1$, which are often chosen to bias the child agents. Thus, the overall trajectory optimization problem for agent $i \in \mathcal{N}$ is summarized as follows:

$$\begin{aligned} &\min_{\{\mathbf{u}^i, \mathbf{x}^i\}} \ C^i \\ \textbf{s.t.} \ &x^i_k = \mathbf{A} x^i_{k-1} + \mathbf{B} u^i_{k-1}, \\ &\|\Theta_a u^i_{k-1}\|_2 \leqslant a_{\max}, \ \|\Theta_v v^i_k\|_2 \leqslant v_{\max}, \forall k \in \mathcal{K}, \\ &a^{ij\,\mathrm{T}}_k p^i_k \geqslant b^{ij}_k, \ a^{i,o\,\mathrm{T}}_k p^i_k \geqslant b^{i,o}_k, \ \forall j \neq i, \ \forall k \in \mathcal{K}, \\ &\left\| p^i_k - c^{ij}_k \right\|_2 \leqslant d_c/2, \ \forall j \in \mathcal{N}^i, \ \forall k \in \mathcal{K}, \\ &\widehat{a}^{ij,o\mathrm{T}}_k p^i_k \geqslant \widehat{b}^{ij,o}_k, \ \forall j \in \mathcal{N}^i, \ \forall k \in \mathcal{K}, \\ &v^i_K = \mathbf{0}_3, \end{aligned} \tag{12}$$

where $\mathbf{u}^i$ and $\mathbf{x}^i$ are the stacked vectors of u^i_{k-1} and x^i_k, $\forall k \in \mathcal{K}$; the last constraint is enforced to avoid an aggressive terminal state. It is worth noting that the optimization (12) is a quadratically constrained quadratic program (QCQP) as it only has a quadratic objective function in addition to linear and quadratic constraints. Thus, it can be fast resolved by off-the-shelf convex optimization solvers, e.g. `CVXOPT` [27]. Moreover, if a solution can not be found, the predetermined trajectory $\overline{\mathcal{P}}$ is chosen as the fallback solution.

The complete planning algorithm is summarized in Alg. 2. The design of communication topology is performed via Alg. 1. Afterwards, the predetermined trajectories of all agents are initialized. In the main loop of execution, the agents exchange their predetermined trajectories via communication in Line 5. At each time step, the connectivity constraints including $\psi^{ij} \triangleq \{c^{ij}_k, \widehat{a}^{ij,o}_k, \widehat{b}^{ij,o}_k\}$ are calculated in Line 6 and exchanged among neighboring agents in Line 7. In combination with the constraints for collision avoidance, the overall constraints for (12) are stored as ζ^i in Line 8. Then, the optimization (12) is formulated and solved locally by each agent $i \in \mathcal{N}$ to obtain its planned trajectory in Line 9. It should be mentioned that the above procedure is applied to all agents including the searchers and connectors.

Algorithm 2: Overall Algorithm

Input : p_g, $\mathcal{P}_{\text{tg}}$, $\mathcal{O}$

1 $\Gamma^i \leftarrow \texttt{ComNet}(p_g, \mathcal{P}^i_{\text{tg}}, \mathcal{O})$;

2 $\overline{\mathcal{P}}^i(t_0) = \{p^i_0, \cdots, p^i_0\}$;

3 **while** *task not accomplished* **do**

4 **for** $i \in \mathcal{N}$ concurrently **do**

5 $\overline{\mathcal{P}}^j(t) \leftarrow$ Receive via communication;

6 $\psi^{i,j} \leftarrow$ Derive constraints (6) and (9);

7 $\zeta^i \leftarrow \{\psi^{i,j}, \forall j \in \mathcal{N}^i\}$;

8 $\zeta^i \leftarrow$ Add constraints (4) and (5);

9 $\mathcal{P}^i(t) \leftarrow$ Solve optimization (12) based on ζ^i;

10 $t \leftarrow t + h$;

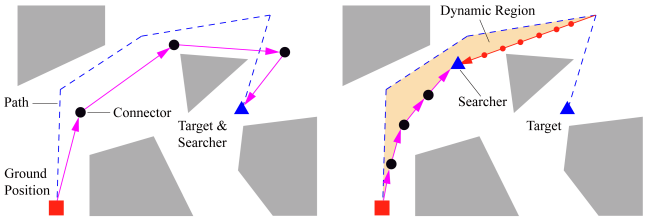

Fig. 4. **Left**: the chain (in solid line) from the ground station to a searcher has the same topological structure as the given path (in dashed line). **Right**: the area in orange shows the dynamic region.

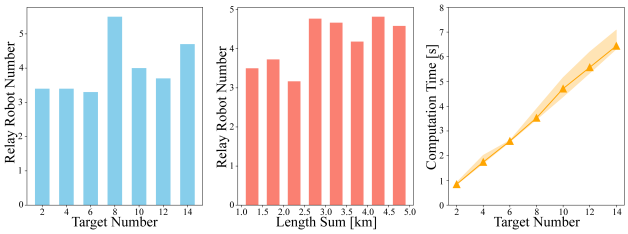

Fig. 5. The communication topology via Alg. 1 given different numbers of targets, including the number of relay agents required (**Left**), the total length of the spanning tree (**Middle**), and the computation time (**Right**).

D. Property Analysis

First, it can be shown that the proposed scheme ensures the multi-agent system remains both safe and connected at all time steps, as stated in the following theorem.

Theorem 1. *If the agents are initially collision-free and connected, the system remains collision-free and the communication graph $G(t)$ remains connected at all times.*

Proof. First, it can be shown that the optimization in (12) is recursively feasible. At the initial time t_0, the predetermined trajectory $\overline{\mathcal{P}}^i(t_0) = \{p^i(t_0), \cdots, p^i(t_0)\}$ satisfies all constraints in (12) as the agents are initially collision-free and the network is connected. Then, the optimization at time t remains feasible since the constraints (6) and (9) are satisfied at t if they are satisfied at $t-h$, as shown in Sec. III-B.2.

Second, given a feasible solution at $t-h$ with control inputs $u^i_{k-1}(t-h)$ and state variables $x^i_k(t-h)$ for $k \in \mathcal{K}$, the planned trajectories $p^i_k(t) = \overline{p}^i_k(t)$ are also feasible. For the constraint in (6), it is clear that $\overline{p}^i_k(t)$ lies within the constrained circle. For the constraint in (9), $\overline{p}^i_k(t)$ satisfies $\widehat{a}^{ij\mathrm{T}}_k \overline{p}^i_k(t) \geqslant \delta + \widehat{b}^{ij}_k$, confirming its feasibility. The feasibility of other constraints has already been proven in [25].

Since both the initial and successive optimizations are feasible, the planned trajectories $\mathcal{P}^i$ derived from the optimization (12) satisfy all constraints for all agents $i \in \mathcal{N}$, which concludes the proof. □

Remark 3. The communication network $G(t)$ is time-varying, with edges being added or removed as the agents move dynamically. However, the communication topology $\mathcal{T}$ derived from Alg. 1 is always contained in $G(t)$, meaning that the connectivity of $\mathcal{T}$ is guaranteed at all time. ■

Another property concerns the evolution of this embedded tree in the workspace. To begin with, the concept of equivalent topologies is defined as follows, similar to [28].

Definition 2. Two polylines are *topologically equivalent* if they share the same start and end points and can be transformed into each other without crossing any obstacle.■

Theorem 2. *The polyline formed by the positions of agents from the ground station to a searcher is topologically equivalent to the path in the embedded tree $\mathcal{T}$.*

Proof. (Sketch) Consider the dynamic region enclosed by the polyline, the searcher's trajectory, the line connecting p^i_K and p^i_{tg}, and the path in the tree $\mathcal{T}$, as shown in Fig. 4. Initially, the region is empty with agents near the ground station. During motion, the polyline and trajectory are collision-free. The line segment from p^i_K to p^i_{tg} does not intersect obstacles, as both points are within a convex polyhedron due to constraint (5). For replanning, only the tractive point changes, and the loop formed by $p_{\text{tg}}(t-h)$, $\overline{p}^i_K$, $p_{\text{tg}}(t)$, and Γ^i does not intersect obstacles. Thus, the dynamic region remains obstacle-free, confirming the topological equivalence. □

The above theorem shows that the polylines formed by all agents remain consistent with the embedded tree $\mathcal{T}$ at all times. In other words, both searchers and connectors are deployed according to the designed tree topology.

IV. Numerical Experiments

This section presents the numerical simulations and hardware experiments to validate the proposed scheme. The method is implemented in Python3 using `CVXOPT` [27] for optimization. All tests run on a computer with Intel Core i9 @3.2GHz of 10 Cores and 32GB memory. The source code is available at https://github.com/CYDXYYJ/IMPC-MD.

A. Numerical Simulations

To mimic ariel vehicles, the maximum velocity and acceleration of agents are set to 15m/s and 3m/s^2, respectively. The replanning interval for MPC is set to the sampling time $h = 0.5\text{s}$. The safety radius of agent is set to $r_a = 2\text{m}$. The communication range and the margin of safe zone are chosen as $d_c = 150\text{m}$ and $d_m = 3\text{m}$, respectively. The considered 3D workspace has a cubic volume from $(0\text{m}, 0\text{m}, 0\text{m})$ to $(500\text{m}, 500\text{m}, 100\text{m})$ with cluttered and convex-shaped obstacles, as shown in Fig. 1.

1) Design of Communication Network: The proposed Alg. 1 is evaluated with different numbers of target points within the obstacle-free space. The algorithm computes the desired communication network by `MiniEdgeRRT`*. For each set of target points, the algorithm is run 15 times. The results, including the number of relay nodes, total length of the spanning tree, and computation time, are summarized in Fig. 5. It is observed that the number of relay agents does not increase linearly with the number of targets, primarily due to the "edge sharing" mechanism in Alg. 1. A similar pattern is observed for the total length of the spanning tree. Finally, the computation time increases linearly with the number of targets, as expected from the analysis in Sec. III-A.

The proposed approach is compared with three baselines: (i) Minimum Spanning Tree (MST), (ii) Visibility Weighted Minimum Spanning Tree (VWMST) in [13], which selects

TABLE I

COMPARISON WITH BASELINES (AVG. OVER 10 RUNS)

Metric	N_r				N_h			
Targets	2	4	6	8	2	4	6	8
MST	7.0	8.4	8.7	9.9	10.7	27.7	38.2	61.6
VWMST	7.1	8.9	9.7	10.8	8.5	19.1	26.3	37.5
DST	7.0	8.2	8.5	9.5	9.1	23.2	33.3	45.3
Ours	6.6	7.7	7.7	8.6	9.6	24.1	32.0	49.5

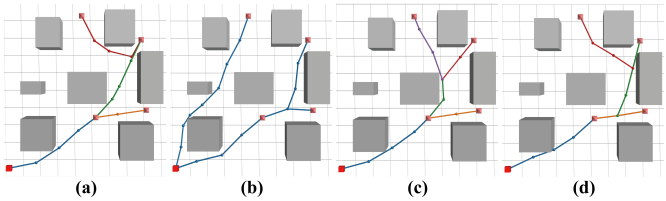

Fig. 6. Comparison of the resulting communication topology for 4 target points, between **(a)** the MST-based method **(b)** the self-organizing method VWMST, **(c)** the distance-based spanning tree DST and **(d)** our method. In addition to 4 searchers, 10, 13, 9, 8 connectors as relay agents are required by these methods, respectively.

the result with the fewest agents, (iii) Distance-based Spanning Tree (DST), which greedily adds targets based on distance. All methods are tested 10 times, comparing the average number of relay agents N_r and hops from the ground station to targets N_h. As shown in Table I, the our method requires only 8 relay agents for 4 targets on average. Compared to VWMST, it uses less than 15% of the relay agents for 4, 6, and 8 targets. VWMST, being decentralized, often leads searchers to take direct paths to targets, resulting in fewer hops but not shared edges. In comparison with MST and DST, our method requires slightly fewer relay agents, while MST has notably more hops.

2) Collaborative Motion Planning: The communication warning distance d_w is set to 142m, and the margin of LOS is set to 3m. The weighting parameters α_j in (11) are set to 3 for child neighbors and 1 for parent neighbors. The resulting trajectories under 7 targets within 46s are shown in Fig. 1 and 7, in addition to the relative distances between any two agents and the minimum distance between their LOS to the obstacles. The results show that the distance between any LOS and obstacles is always greater than $d_m = 3$m, and the distance between neighboring agents stays below $d_c = 150$m. Besides, the distance between any two agents stay positive at all times. The average speed of all 7 searchers reaches 80% of the maximum speed, ensuring efficient execution while maintaining safety and connectivity. As shown in Fig. 8, the average computation time per MPC iteration is 35ms across 100 iterations, with 17.3ms for constraints derivation in (8), and 17.7ms for solving the optimization in (12). The former however exhibits a larger variance across iterations due to the varying proximal obstacle environment.

For comparison, the method in [21] is implemented via MILP. The resulting trajectories in different workspaces are shown in Fig. 9. For a cubic obstacle of width 100m, the method in [21] takes over 857ms to find a trajectory of length 850m. In comparison, our method takes only 65ms and results in a trajectory of 765m. For a larger cubic obstacle of

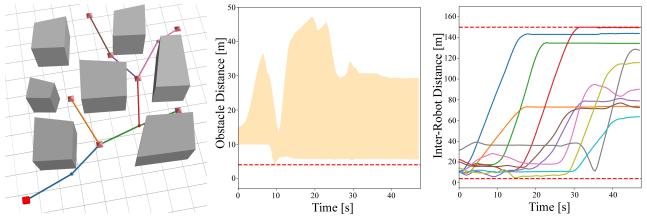

Fig. 7. Illustration of the communication topology (**Left**); the relative distance between the LOS of neighboring agents and the closet obstacle (**Middle**); and the relative distance between interconnected agents (**Right**).

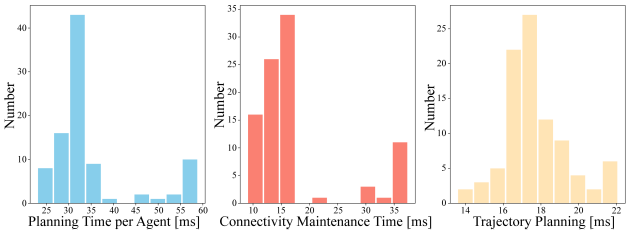

Fig. 8. Distribution of the computation time of each iteration across agents (**Left**), which consists of: (i) the time to derive the constraints in (8) (**Middle**); and (ii) the time to solve the final optimization in (12) (**Right**).

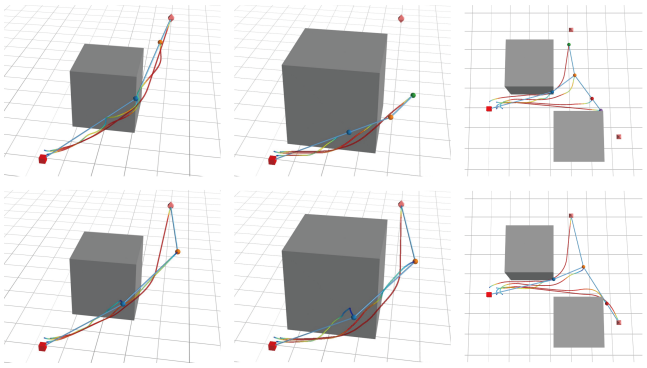

Fig. 9. Comparisons of the resulting trajectories by the method in [21] (**Top**) and our method (**Bottom**) for different obstacle environments.

width 150m, the method in [21] fails to complete the task, while our method finishes within 30.0s. In another scenario with two obstacles requiring more relay points, the method in [21] fails to generate a feasible solution due to long computation times (over 10s per replanning) and obstruction. Our method, however, completes the task in 23.5s with a trajectory length of 1291m, demonstrating its robustness in various workspaces.

B. Hardware Experiment

Hardware experiments are conducted using Crazyswarm, with motion tracked by OptiTrack and data transmitted to a workstation via Motive. Due to the down-wash effect, each quadrotor is enclosed by a safety ellipsoid with a diameter of 0.24m in the xy plane and 0.6m along the z axis. The maximum velocity and acceleration are set to 1.0m/s and 1.0m/s^2, with a sampling time $h = 0.2s$ and a horizon length $K = 8$. The workspace includes known 3D walls. Alg. 2 runs on a central computer with multi-processing, where each agent operates in an independent process.

In first experiment, two targets are placed between the walls. The communication topology consists of 4 connectors

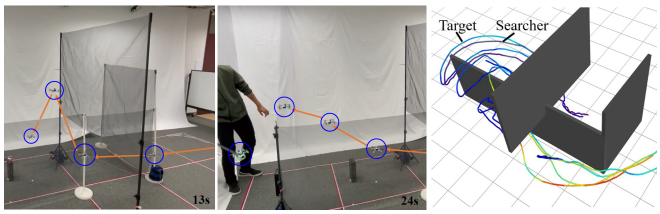

Fig. 10. Illustration of how the network and the agent trajectories adapt online when the target is dynamically moving.

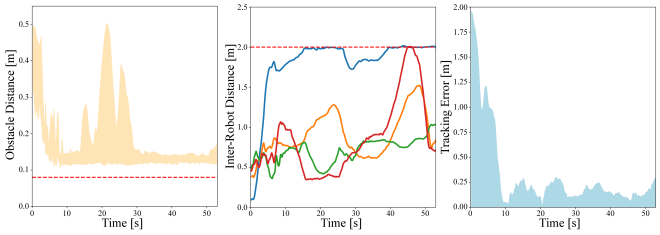

Fig. 11. Numerical results from the experiment study: the minimum distance between the LOS of any neighboring agents and the obstacles (**Left**); the distance between any pair of neighboring agents (**Middle**); and the tracking error between the target and the assigned searcher (**Right**).

and 2 searchers in an "F"-shaped configuration, as shown in Fig. 1. The entire mission takes around 10s, with an average velocity of $0.4\mathrm{m/s}$ for all targets, while ensuring collision avoidance and connectivity. In the second experiment, the target is *moved manually* as shown in Fig. 10. requiring the searcher to track it. The whole fleet adapts to the movement, maintaining the motion constraints, as shown in Fig. 11. The tracking error stays below $0.25\mathrm{m}$ at all times, demonstrating the system's robustness during online trajectory replanning.

V. Conclusion

This work presents a multi-UAV deployment method in obstacle-cluttered environments, ensuring LOS connectivity and collision avoidance. Different from most existing methods, the team size is designed, and the distributed MPC guarantees feasibility and safety. Future work will explore broader applications, such as exploration and surveillance.

References

[1] S.-J. Chung, A. A. Paranjape, P. Dames, S. Shen, and V. Kumar, "A survey on aerial swarm robotics," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 837–855, 2018.

[2] Q. Quan, R. Fu, M. Li, D. Wei, Y. Gao, and K.-Y. Cai, "Practical distributed control for vtol uavs to pass a virtual tube," *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 2, pp. 342–353, 2022.

[3] Z. Hu and X. Jin, "Formation control for an uav team with environment-aware dynamic constraints," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 1, pp. 1465–1480, 2024.

[4] Z. Tian, Y. Zhang, J. Wei, and M. Guo, "ihero: Interactive human-oriented exploration and supervision under scarce communication," in *Robotics: Science and systems (RSS)*, 2024.

[5] D. Yin, X. Yang, H. Yu, S. Chen, and C. Wang, "An air-to-ground relay communication planning method for uavs swarm applications," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 4, pp. 2983–2997, 2023.

[6] Y. Zhang, Z. Tian, J. Wei, and M. Guo, "Flykites: Human-centric interactive exploration and assistance under limited communication," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

[7] T. Schouwenaars, E. Feron, and J. How, "Multi-vehicle path planning for non-line of sight communication," in *American Control Conference (ACC)*, 2006, pp. 5757–5762.

[8] 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.

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

[10] 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)*, 2020, pp. 9845–9851.

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

[12] J. Esposito and T. Dunbar, "Maintaining wireless connectivity constraints for swarms in the presence of obstacles," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2006, pp. 946–951.

[13] M. Boldrer, P. Bevilacqua, L. Palopoli, and D. Fontanelli, "Graph connectivity control of a mobile robot network with mixed dynamic multi-tasks," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1934–1941, 2021.

[14] M. Guo and M. M. Zavlanos, "Multirobot data gathering under buffer constraints and intermittent communication," *IEEE transactions on robotics*, vol. 34, no. 4, pp. 1082–1097, 2018.

[15] L. Clark, J. A. Edlund, M. S. Net, T. S. Vaquero, and A. akbar Aghamohammadi, "Propem-l: Radio propagation environment modeling and learning for communication-aware multi-robot exploration," *arXiv preprint arXiv:2205.01267*, 2022.

[16] Z. Feng, C. Sun, and G. Hu, "Robust connectivity preserving rendezvous of multi-robot systems under unknown dynamics and disturbances," in *IEEE Conference on Decision and Control (CDC)*, 2015, pp. 4266–4271.

[17] P. R. 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.

[18] 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.

[19] M. Guo, M. M. Zavlanos, and D. V. Dimarogonas, "Controlling the relative agent motion in multi-agent formation stabilization," *IEEE Transactions on Automatic Control*, vol. 59, no. 3, pp. 820–826, 2013.

[20] 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)*, 2016, pp. 2659–2664.

[21] A. Caregnato-Neto, M. R. O. A. Maximo, and R. J. M. Afonso, "Resilient robust connectivity for multiagent systems with line of sight using mixed-integer programming," *Journal of Control, Automation and Electrical Systems*, vol. 33, no. 1, pp. 129–140, Feb 2022.

[22] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Informed rrt*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 2997–3004.

[23] G. Franzè, W. Lucia, and A. Venturino, "A distributed model predictive control strategy for constrained multi-vehicle systems moving in unknown environments," *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 2, pp. 343–352, 2021.

[24] Y. Chen, M. Guo, and Z. Li, "Deadlock resolution and recursive feasibility in mpc-based multi-robot trajectory generation," *IEEE Transactions on Automatic Control*, 2024.

[25] Y. Chen, C. Wang, M. Guo, and Z. Li, "Multi-robot trajectory planning with feasibility guarantee and deadlock resolution: An obstacle-dense environment," *IEEE Robotics and Automation Letters*, vol. 8, no. 4, pp. 2197–2204, 2023.

[26] R. L. Burden, J. D. Faires, and A. M. Burden, *Numerical analysis*. Cengage learning, 2015.

[27] A. Martin, D. Joachim, and V. Lieven, "Cvxopt," Website, http://cvxopt.org/.

[28] F. Gao, L. Wang, B. Zhou, X. Zhou, J. Pan, and S. Shen, "Teach-repeat-replan: A complete and robust system for aggressive flight in complex environments," *IEEE Transactions on Robotics*, vol. 36, no. 5, pp. 1526–1545, 2020.