Communication-Topology-preserving Motion Planning: Enabling Static Routing in UAV Networks

ZIYAO HUANG, Southeast University, China and City University of Hong Kong, Hong Kong
WEIWEI WU, CHENCHEN FU, XIANG LIU, and FENG SHAN, Southeast University, China
JIANPING WANG, City University of Hong Kong, Hong Kong
XUEYONG XU, Northern Information Control Research Academy Group Co., Ltd, China

Unmanned Aerial Vehicle (UAV) swarm offers extended coverage and is a vital solution for many applications. A key issue in UAV swarm control is to cover all targets while maintaining connectivity among UAVs, referred to as a multi-target coverage problem. With existing dynamic routing protocols, the flying ad hoc network suffers outdated and incorrect route information due to frequent topology changes. This might lead to failures of time-critical tasks. One mitigation solution is to keep the physical topology unchanged, thus maintaining a fixed communication topology and enabling static routing. However, keeping physical topology unchanged may sacrifice the coverage. In this article, we propose to maintain a fixed communication topology among UAVs, which allows certain changes in physical topology, so that to maximize the coverage. We develop a distributed motion planning algorithm for the online multi-target coverage problem with the constraint of keeping communication topology intact. As the communication topology needs to be timely updated when UAVs leave or arrive at the swarm, we further design a topology-management protocol. Experimental results from the ns-3 simulator show that under our algorithms, UAV swarms of different sizes achieve significantly improved delay and loss ratio, efficient coverage, and rapid topology update.

CCS Concepts: • **Networks** → **Mobile ad hoc networks**; • **Theory of computation** → *Design and analysis of algorithms*; • **Computer systems organization** → *Sensor networks;*

Additional Key Words and Phrases: UAV networks, motion planning, coverage, routing protocol, topology management

The work is supported in part by the National Key Research and Development Program of China under Grant No. 2019YFB2102200; the Science Technology and Innovation Committee of Shenzhen Municipality under project JCYJ20170818095109386; National Natural Science Foundation of China under Grants No. 61972086, No. 62072101, No. 62132009, No. 62232004, and No. 62102082; the Jiangsu Natural Science Foundation of China under Grant No. BK20210203; the Key Laboratory of Computer Network and Information Integration (Southeast University), Aeronautical Science Foundation of China under Grant No. 2017ZC69011; the National Science Foundation under Grant No. CNS-1730325; and the Hong Kong Research Grant Council under GRF 11216618.

Authors' addresses: Z. Huang, Southeast University, China and City University of Hong Kong, Hong Kong; e-mail: seuhuangziyao@outlook.com; W. Wu, C. Fu (Corresponding author), X. Liu, and F. Shan, Southeast University, China; e-mails: weiweiwu@seu.edu.cn, chenchenfu@seu.edu.cn, xiangliu@seu.edu.cn, shanfeng@seu.edu.cn; J. Wang, City University of Hong Kong, Hong Kong; e-mail: jianwang@cityu.edu.hk; X. Xu, Northern Information Control Research Academy Group Co., Ltd, China; e-mail: xxyyeah@163.com.

1550-4859/2023/12-ART24 $15.00
https://doi.org/10.1145/3631530

ACM Reference format:
Ziyao Huang, Weiwei Wu, Chenchen Fu, Xiang Liu, Feng Shan, Jianping Wang, and Xueyong Xu. 2023. Communication-Topology-preserving Motion Planning: Enabling Static Routing in UAV Networks. *ACM Trans. Sensor Netw.* 20, 1, Article 24 (December 2023), 39 pages.
https://doi.org/10.1145/3631530

1 INTRODUCTION

Unmanned Aerial Vehicle (UAV) swarm has been widely used in applications such as surveillance [13, 64], three-dimensional (3D) mapping [50, 59], disaster management [18, 20], and so on, where collaborative efforts among UAVs are needed. Existing studies [27, 34, 55, 66] have investigated various issues in UAV swarms, such as motion planning, connectivity, coverage, deployment, and so on. UAVs in the swarm are normally modeled as mobile agents [9, 58], and the tasks can be formulated as to visit a set of target points sequentially, referred to as the multi-target coverage problem.

To enable coordination among UAVs, most existing studies [5, 28, 32, 38, 42, 48, 51, 62] take the global connectivity constraint into consideration, i.e., during movement, a communication path should always exist between any two UAVs. However, compared with other mobile networks like vehicular ad hoc networks and mobile ad hoc networks, one unique challenge in the UAV network is that it suffers high frequency of topology change [24]. This problem is mainly caused by (1) the high mobility of UAVs and (2) dynamic UAV leave and arrival of the swarm. Next, we will respectively address these two issues.

First, as demonstrated in Figure 1(a) and (b), due to the high mobility of UAVs, the network topology of the swarm changes dramatically over time. To practically transfer data via or among such UAVs, communication paths have to be dynamically established or maintained through *dynamic routing*. More concretely, in dynamic routing algorithms, a packet would be transferred (1) either immediately via a periodically maintained communication path (in a proactive routing protocol such as OLSR [8]) (2) or later until the needed communication path has been found and established (in a reactive routing protocol such as AODV [41]). The frequent topology changes in proactive routing protocols can result in outdated communication paths being used for packet forwarding. These paths may no longer be efficient or even viable, leading to a higher end-to-end delay and packet loss ratio. Although in reactive routing protocols, the routes are found in an on-demand manner before the packets are sent out, the convergence time of the routing algorithm can be long, leading to extra end-to-end delay and also outdated route information. Overall, dynamic routing protocols suffer from long delays and packet loss ratios led by outdated route information, and this can cause failures of time-critical tasks for UAV swarms.

To mitigate the problem of frequent topology change led by UAV mobility, one simple solution is to maintain a fixed physical topology of UAV swarm and use static routing instead of dynamic routing, i.e., both of the physical topology and communication/network topology keep intact during the journey. However, maintaining a fixed physical topology may sacrifice the performance of coverage as UAVs in a swarm have less flexibility to cover the targets. In this article, we propose to maintain a fixed communication topology where the physical topology of UAVs can be changed in motion planning so that more targets can be covered. Figures 1(c) and (d) illustrate the idea of maintaining a virtually fixed communication topology when UAVs move.

We anticipate that this concept will significantly enhance the performance of the UAV network, particularly in terms of minimizing packet loss and reducing delays. This is due to the fact that the UAVs responsible for communication will always have viable routes, and the quality of their corresponding links will be high as the distance between them will be constrained. Preliminary

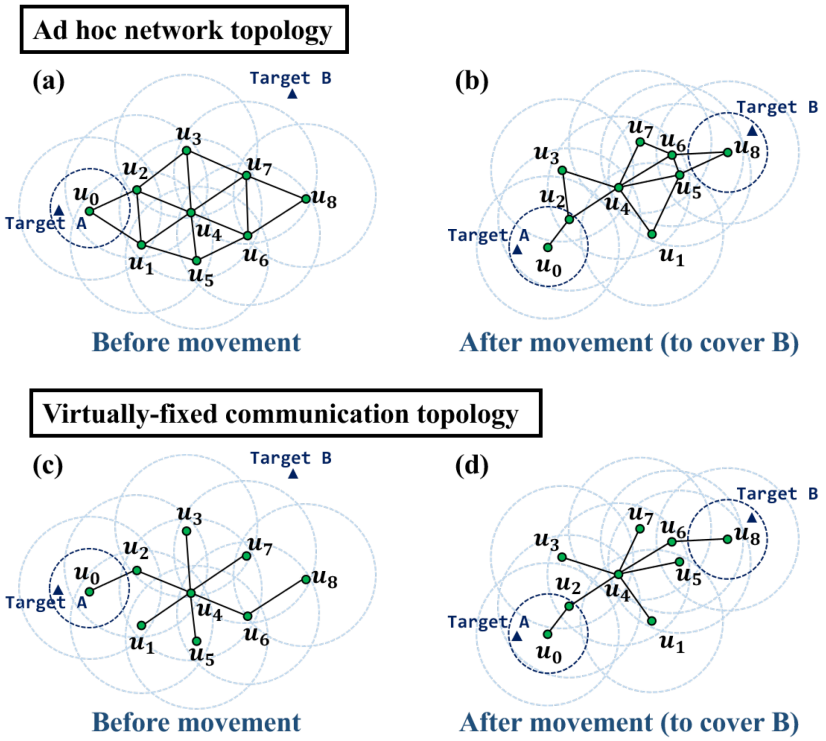


Fig. 1. Example that moving UAVs could communicate via a fixed communication topology. The communication range and sensing range of UAVs are represented as light black circles and dark black circles, respectively. (a) Ad hoc network topology that contains all links between any two communicable UAVs before movement. (b) The ad hoc network topology changes after movement. (c) Pre-configured communication topology that contains certain links between those communicable UAVs before movement. (d) The communication topology is maintained fixed after movement.

experimental results also validate that compared with maintaining a fixed communication topology during the movement of UAVs, OLSR and AODV do suffer from a high ratio of routing entries with long communication distances and a long average distance for the routes that are actually used for communication.

Although the idea is validated to be viable, it requires a well-designed motion planning strategy to enable the virtually fixed communication topology, without affecting the coverage performance of UAV swarm. Therefore, we consider the objective of maximizing the number of covered targets when UAVs move and formulate the problem as a **Topology-Preserving Multi-Target Coverage (TPMTC)** problem. More specifically, we assume that each interested/target point to be visited arrives dynamically and sequentially. For each newly appeared target, we design a distributed motion plan of UAV swarms to visit the new target and meanwhile cover as many existing targets as possible. The main challenges of TPMTC problem can be summarized as follows:

— The objective is the effective targeting of online objects by a swarm of UAVs. To achieve this, the motion planning algorithm must be capable of adapting to real-time changes in target locations and dynamically transforming their formation to cover as many targets, including the newly arrived ones, as possible.

— Each UAV is expected to move in a distributed and cooperative manner. The movement should cooperatively maintain the fixed communication topology feasible while achieving high coverage.

Second, as mentioned earlier, developing the idea of maintaining a fixed communication topology still faces another challenge, i.e., the dynamic UAV leave and arrival. For example, to deal with battery shortage while conducting long-lasting missions, it is a common paradigm in swarm management solutions to let UAVs provisionally leave the swarm and recharge or directly be replaced by other fully charged UAVs [12, 15, 16]. In this case, the maintained communication topology can be broken as a UAV leaves and connections with newly arrived UAVs are needed to be established. To mitigate this problem, we also design a topology-management protocol for the swarm to adapt to the UAV leave and arrival, which enables the swarm to maintain the communication topology under more sophisticated management operations.

In the literature, we are not the first to be aware of both communication and coverage for mobile agents. For example, Kantaros et al. [19] investigated the problem of optimizing area coverage while guaranteeing reliable communication among robots, Rahmanpour et al. [44] focused on visiting targets while ensuring data transmission rate. Fernando, Senanayake, and Swany [14] investigated the communication-aware control problem for covering a set of user equipment using a UAV swarm. All such works follow the framework of dynamic routing, which suffers from outdated route information. Our contributions can be summarized as follows:

— To address the problem of frequently varying network topology in UAV networks while avoiding unnecessarily weakening the coverage ability of swarm, we propose to maintain a virtually fixed communication topology among UAVs, i.e., ensure the maximum physical distance of UAVs pairs specified in a given graph is below the communication range. Thus, static routing can be enabled to preserve high-quality links and avoid outdated route information.
— We design a distributed motion planning algorithm for the UAV swarm to cooperatively cover the targets online. The algorithm is theoretically proven to guarantee that the UAV swarm converges to cover the targets while preserving the given communication topology during the movement phase. To boost coverage efficiency, we also design a parallel-coverage mechanism to cover multiple online-appearing targets simultaneously. Evaluation results on the ns-3 network simulator demonstrate that compared to dynamic routing protocols OLSR and AODV, the average end-to-end delay and average packet loss ratio are reduced by at least 42.856% and 51.91%, respectively.
— A topology-management protocol is proposed to further maintain the communication topology in more complex swarm management paradigms. With this protocol, the newly arrived member can easily start cooperating with others, and as a UAV leaves the swarm, the broken communication topology can be healed Results on a real postdisaster analysis dataset demonstrate a smaller cumulative moving distance of UAVs and recovery delay for the topology recovery process when UAVs leave the swarms in various sizes as compared to the baseline.
— Moreover, the developed topology-management protocol enables the UAV swarm to reconfigure the communication topology adaptively. With this feature, the **Topology-Preserving Motion Planning (TPMP)** algorithm can thoroughly exploit the flexibility of UAV swarms. The results on the real dataset show that the coverage performance under the TPMP algorithm with reconfiguration can even outperform the fully centralized baseline on both online and offline scenarios, while preserving the superior network performance using the static routing protocol.

The remainder of this article is organized as follows. The relevant studies are summarized in Section 2. Both the Topology-Preserving Multi-Target problem and the Topology Management

problem are formulated in Section 3. The algorithms proposed for the two problems are respectively introduced in Sections 4 and 5. The theoretical analysis of the proposed algorithms is presented in Section 6. The coverage quality, communication quality, and topology recovery quality of our algorithms are then evaluated in Section 7. Finally, we conclude the work in Section 8.

2 RELATED WORK

This article proposes the idea of preserving communication topology to enable static routing. We investigate the online multi-target coverage problem to address the tradeoff between coverage and communication among UAVs. Then a topology management protocol is developed to deal with dynamic UAV leaves and arrivals. Therefore, we respectively review the relevant studies on both connectivity-aware coverage and topology management of UAVs.

2.1 Connectivity-aware Coverage

Over the years, various coverage problems have been studied in different contexts [4, 31, 61]. In consideration of communication among agents for data transferring or agent coordination, most works [6, 52, 60] require a communication path always exists between any two agents, i.e., the global connectivity must be maintained. For example, in mobile sensor networks, researchers have been working on the Connected Target Coverage problem [11, 25, 36, 57, 60, 63] in past decades, which requires a group of connected sensors to cover multiple targets while optimizing metrics like energy cost, the number of used sensors, or moving distance. In multi-agent coordination, Reference [6] studies the problem of path planning for connected agents to visit points in a graph, and more problems of multi-agent pathfinding and conflict-based search under the connectivity constraint are summarized in the thesis [43]. In robotics, References [21, 38, 52] respectively investigate the problem of multi-target coverage, robust area coverage, and topology formation under the global connectivity constraint among robots. In these works, only global connectivity is considered, while the communication quality has not been discussed. Actually, maintaining global connectivity only guarantees the feasibility of communications. The communication quality could still be worse due to frequent topology changes.

In the past few years, a number of studies have taken the communication quality into consideration in decision makings on motion control for a swarm of agents [19, 26, 29, 30, 35, 44, 45, 49, 68]. In terms of the optimization objective, they either investigate the quality of coverage defined in different ways [19, 29, 35, 44, 45] or study non-coverage or general objectives [26, 30, 49, 68]. In those works, much more complex communication models than ours are used to evaluate the quality of communications between agents, and then the corresponding communication-aware strategies are proposed to improve the communication quality. However, those strategies are tightly related to the models they defined and still work under the framework of dynamic routing.

2.2 Topology Management

The network topology of a swarm of UAVs is jointly determined by the physical positions of the UAVs, which dictate whether a communication link exists between any two UAVs, and the routing entries of the UAVs, which govern the use of communication links when forwarding data packets. Prior research on managing the topology of a UAV swarm can be categorized into two groups: those that relocate UAVs [1, 7, 10, 22, 39] and those that update routing entries, known as routing protocols.

Routing protocol research, as detailed in References [24, 47], focuses on updating routing entries and does not involve UAV movement. Conversely, research on topology management that relocates UAVs includes several relevant studies. For example, Reference [7] uses a deep reinforcement learning algorithm to program the trajectory of each UAV, enabling them to create a varying topology

over time to maximize net benefit. Reference [10] introduces a topology management algorithm that initializes and modifies the positions of UAVs to maintain desired connectivity. To accomplish sparsely distributed missions with UAVs, Reference [22] develops a topology management algorithm for each UAV that serves as a relay, dynamically adjusting positions to maintain global connectivity and maximize overall objectives. Similarly, the topology management algorithm in Reference [39] adopts the virtual force model to adjust the positions of relay UAVs. Reference [1] proposes cooperation between a DRL-based router and topology management protocols that control UAV positions, but their approach is under the dynamic routing framework, which cannot support topology-preserving UAV swarms.

Most existing studies on topology management of UAV swarms update positions and routing entries of UAVs separately and within the dynamic routing framework. In contrast, our proposed topology management approach can easily update both positions and routing entries of UAVs, since it operates under the topology-preserving framework, where the communication topology simultaneously determines both elements.

2.3 Summary

In summary, to the best of the authors' knowledge, no prior works have considered the concept of preserving communication topology while covering multi-targets online with flexible physical topology, which fundamentally improves the network performance and achieves high coverage.

3 PRELIMINARY

3.1 Motivating Scenario

We consider real-time aerial monitoring systems, which are widely used in various scenarios like search and rescue [2], road detection [17], and disaster monitoring [53]. In such scenarios, targets (interesting points) appear over time and need to be observed/monitored via UAVs. Large amounts of real-time data generated are transferred over the UAV network, and hence the UAV swarm can preserve a fixed communication topology to boost the network efficiency as validated earlier. Ideally, to collect enough data for analysis, all targets should be covered for the time period from the moment that it appears to the moment that the UAV swarm completes its journey. However, given the limited number of UAVs and the dynamics of target arrivals, it is not possible to cover all targets at the same time. As a compromise solution, upon the arrival of a newly appeared target, the UAV swarm needs to cover the new target (for fairness and diversity of collected data) and try its best to keep as many existing targets being observed/covered as possible (for collecting larger amount of data).

3.2 TPMTC Problem

Now we formulate the problem above as a TPMTC problem that preserves a fixed communication topology while covering new appearing targets. Let U denote the UAV swarm. Let $p_{u,t}$ denote the position of UAV $u \in U$ at time t. Denote status profile at time t for the swarm U to be $P_{U,t} = \{p_{u,t} \in \mathbb{R}^2 | \forall u \in U\}$. Then $P_{U,0}$ contains initial positions of swarm U. Let $d_t(u_i, u_j)$ denote the Euclidean distance between UAVs u_i and u_j at time t. Also, we slightly abuse $d(p_i, p_j)$ to denote the distance between any two points p_i and p_j. A point of interest appears online with an arbitrary position and needs to be covered at least once by a UAV.[1] Let T denote the set of all targets. We

[1]The information of the target can be offered by another part of the swarm application on top of the motion planning module (our focus), e.g., a ground station or an object searching and finding module, which depends on the specific scenario and is beyond the scope of this article. We only assume that the information is revealed to one of the UAVs in the swarm.

use τ_i to denote the ith target, p_τ to denote the position of any target $\tau \in T$, and $d_t(\tau, \cdot)$ to denote the distance between a target τ and another target, UAV, or point at time t.

Coverage Model: A target τ is covered by UAV u at time t if the distance between them is not greater than the sensing range r_s, i.e., $d_t(\tau, u) \leq r_s$. We use $\eta_{u,t}$ to denote the set of targets covered by UAV u at time t.

Communication Topology: Certain pairs of pre-configured communication links between UAVs enable static routing among mobile UAVs. Formally, we denote the communication topology for U by $\widetilde{E} = \{(u_i, u_j) | u_i, u_j \in U\}$, and the graph $G(U, \widetilde{E})$ need to be acyclic, i.e., only one path exists between two UAVs over the communication topology. We use $\widetilde{N}_u = \{u' | (u, u') \in \widetilde{E}\}$ to denote u's neighbors in $\widetilde{E}$ that are called its $\widetilde{E}$-neighbors. The communication topology is exogenously given, and it is *feasible* at time t if each pair of UAVs in it can communicate with each other at time t, i.e., $d_t(u_i, u_j) \leq r_c, \forall (u_i, u_j) \in \widetilde{E}$, where r_c is the communication range. We aim to maintain the communication topology feasible during the movement of the UAV swarm when performing a coverage task.

Motion Plan: We use a four-tuple (t, v, u, p) to describe the movement of UAV, which indicates that UAV u moves to another position p with speed v at time t. The UAV swarm responds to the appearance of a newly appeared target by a motion plan. In general, we denote a motion plan by $M_{U,P_{U,0}} = \{(t, v, u, p) | t, v \in \mathbb{R}^+, u \in U, p \in \mathbb{R}^2\}$. It describes each UAV's motion at any time and the change of formation. Besides, we define the ending time of the plan as $t_M = \max\{t + d(p_{u,t}, p)/v \big| (t, v, u, p) \in M\}$.

We will write $P_{U,t}$, $\widetilde{E}_U$, $M_{U,P_{U,0}}$ as P_t, $\widetilde{E}$, M for short, respectively, if no ambiguity arises.

Next, we introduce the problem formulation. To keep communication topology feasible during UAV movement, the motion plan for UAVs should be topology preserving. We say that a motion plan M is topology preserving for UAVs U if and only if the pre-configured communication topology is maintained feasible during the whole movement. That is,

$$d_t(u_i, u_j) \leq r_c, \forall (u_i, u_j) \in \widetilde{E}, \forall t \in [0, t_M]. \tag{1}$$

Assume that t_i^a is the time when the ith target τ_i appears. The UAV swarm should change its formation to cover each newly appeared target, i.e.,

$$\exists t \in [t_i^a, t_M], u \in U,\ d_t(\tau_i, u) \leq r_s, \forall \tau_i \in T. \tag{2}$$

Meanwhile, the UAV swarm should keep covering as many existing targets as possible. We measure the coverage ability of a motion plan as follows. As targets arrive online sequentially, we divide the covering process into phases. In phase i, the coverage/visit requirement of the target τ_i from the UAV monitoring system arrives, and the UAV swarm moves to cover the target and meanwhile keeps covering as many existing targets as possible. Let n_i be the number of existing targets that are still kept observed/covered at the end of phase i. Then our objective is to maximize the average number of covered targets over phases, which naturally measures the **Average Coverage (AC)** of a motion plan. Therefore, the TPMTC problem can be defined as follows.

Definition 3.1 (TPMTC Problem). Given communication topology $\widetilde{E}$, UAV set U, and the initial status P_0, determine a motion plan M on the arrival of a target to maximize the average coverage $\sum_{i=1}^{|T|} n_i/|T|$ subject to constraints (1) and (2).

The notations are summarized in Table 1.

Since targets appear online, another might appear while UAVs are trying to cover one target. To boost the coverage efficiency or reduce the waiting time of targets, we also hope that the method that solves the TPMTC problem can deal with multiple targets simultaneously if possible. Thus, a parallel-coverage mechanism is also needed to reinforce the solution of TPMTC.

Table 1. Notations

Notation	Description
U	UAV set
T	Target set
τ_i	ith appeared target
$p_{u,t}$	UAV u's position at time t
P_t	Set of each UAV's position at time t
$d_t(\cdot,\cdot)$	Distance between any two points at time t
$\eta_{u,t}$	Set of targets covered by u at time t
N_t	Set of covered targets at time t
$n(t)$	The number of covered targets at time t
$\widetilde{E}$	The communication topology of U
$\widetilde{N}_u$	UAV u's neighbors in $\widetilde{E}$
M	Motion plan for U
t_M	The end time of M
t_i^a	The time of τ_i's appearance

3.3 Problem Hardness

To demonstrate the hardness of our problem, we define the SLK-TPMTC problem where UAVs are deployed to cover all targets. Here the "SLK" refers to "slack," indicating this problem is relatively easier than the original problem, since it deals with offline scenarios. In the SLK-TPMTC problem, all targets in T arrive initially, and moreover, only the final coverage, i.e., the number of covered targets at the final phase, is expected to be maximized instead of the average coverage.

Theorem 1. *SLK-TPMTC is NP-hard.*

Proof. We show that the SLK-TPMTC problem is NP-hard. The main idea of the proof is to reduce the **"Covering by Two Balls" (CTB)** problem [33], an NP-hard problem, to the decision version of the SLK-TPMTC problem.

The CTB problem is described as follows: Given a set of points $S = \{s_1, \ldots, s_m\}$, find two balls B_1, B_2 with the radius of 1 that cover all points at the same time or determine no such balls exist. The decision version of the SLK-TPMTC problem that is called K-SLK-TPMTC problem is to decide whether at least K targets can be covered by UAVs at the same time in SLK-TPMTC problem.

The reduction from the CTB problem to K-SLK-TPMTC problem can be conducted as follows. For any instance of a CTB problem, view each point in S as a target to be covered by UAVs. We set two UAVs $U = \{u_1, u_2\}$ and the parameters to be $\widetilde{E} = \{(u_1, u_2)\}$, $r_s = 1$, $r_c = 2 + \max\{d(s_i, s_j) | \forall s_i, s_j \in S\}$, $K = |S|$. It is obvious that the reduction can be done within polynomial time.

Next, we will demonstrate that $|S|$ points can be covered in the CTB problem if and only if K targets can be covered in the constructed instance of K-SLK-TPMTC problem.

First, if only one point exists in the CTB problem, then obviously the only point in the CTB problem, and the only target in K-SLK-TPMTC problem can be covered.

Then we consider $K = |S| \geq 2$. On the one hand, if in the CTB problem, all $|S|$ points can be covered, then obviously there exists a feasible solution where the distance between these two balls is no greater than $2 + \max\{d(s_i, s_j) | \forall s_i, s_j \in S\}$. Thus, we just make u_1, u_2 positioned as B_1, B_2 does. This implies a feasible solution in the constructed instance to cover K targets. If $|S|$ points cannot be covered simultaneously, then obviously any two UAVs cannot cover all K targets in the constructed instance even without considering communication range constraints. That is, there exists no feasible solution covering K targets in the constructed instance of K-SLK-TPMTC problem.

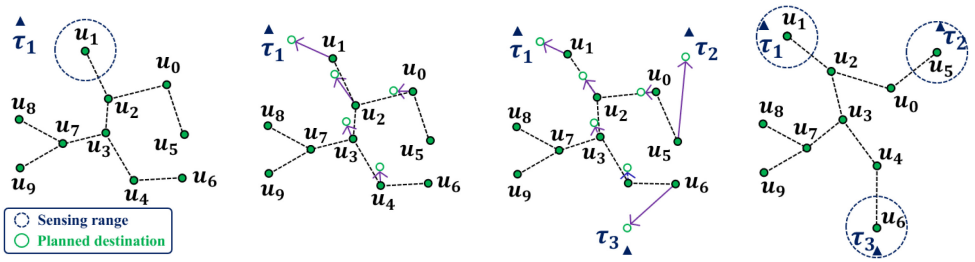


Fig. 2. Overview of the TPMP method. (a) Target τ_1 appears by running the KMA algorithm, and u_1 is elected to be in charge of covering τ_1. (b) To cooperatively enable u_1 to cover τ_1, UAVs run the TP3 algorithm to plan their destinations. (c) As UAVs are moving, another two targets τ_2 and τ_3 appear. Then UAVs adopt the Parallel-Coverage mechanism to try to cover all of them and re-calculate their destinations. (d) By running the ULMC algorithm, UAVs move simultaneously and arrive at their destinations. After that, all targets are covered.

On the other hand, in the constructed instance of the K-SLK-TPMTC problem, if all K targets can be covered, then we just make two balls B_1, B_2 positioned as u_1, u_2 does. This implies a feasible solution covering all $|S|$ points in the CTB problem. Moreover, if K points cannot be covered simultaneously, then obviously there exist no feasible solutions covering $|S|$ points in the CTB problem.

Therefore, $|S|$ points can be covered in the CTB problem if and only if K targets can be covered in the constructed instance of the K-SLK-TPMTC problem. □

3.4 Topology-management Problem

As mentioned earlier, the UAV swarm might face dynamic UAV arrival and leave, e.g., UAV in the swarm provisionally leave the swarm and return to recharge its battery. When a UAV leaves or arrives, to continue moving while preserving a fixed communication topology, the swarm needs to manage its communication topology judiciously. Especially when a UAV leaves the swarm, the preserved communication topology might break, and, thus, motion planning for remaining UAVs is invoked to adjust their positions and rebuild a new communication topology. Formally, based on the preceding models, we formalize the TM problem as follows.

Definition 3.2 (TM Problem). Given communication topology $\widetilde{E}$, UAV set U, a leaving or arriving UAV $u \in U$, and current status P_t, distributedly determine a new communication topology $\widetilde{E}'$ for new UAV set $U \setminus \{u\}$ or $U \cup \{u\}$ and a motion plan M such that $\widetilde{E}'$ is feasible after conducting M.

4 TOPOLOGY-PRESERVING MOTION PLANNING

In this section, a distributed method named TPMP is presented to solve the online TPMTC problem. The method contains three algorithms to cover each newly appeared target and a parallel-coverage mechanism to enhance the parallel processing capability of the UAV swarm. An example of TPMP can be found in Figure 2.

4.1 Overview

The online TPMTC problem could be divided into three sub-problems as follows: (1) *who to cover*: Which UAV is in charge of covering the newly appeared target?; (2) *where to go*: What are the final destinations of UAVs to cooperatively enable the selected UAV to cover the target while

preserving the communication topology?; and (3) *how to move*: How do the UAVs move from their current positions to their final destinations while preserving the communication topology during movement?

The proposed TPMP works as follows. When a new target τ_{new} appears, first the **Keep Most Anchors (KMA)** algorithm is called to elect a UAV named the root UAV u_{root} to be in charge of covering τ_{new}. Second, the **Topology-Preserving Pre-Planning (TP3)** algorithm is called to calculate the final destinations P_U^d where τ_{new} is covered by u_{root} and the UAV swarm covers as many existing targets as possible. Third, the **Uniform Linear Motion Control (ULMC)** algorithm is called to drive UAVs simultaneously to P_U^d from their initial positions P_U^i while preserving the communication topology during the movement.

Moreover, since the targets appear online, while UAVs are conducting an existing motion plan to cover a target τ_{now}, other targets might appear midway. UAVs might be able to cover τ_{now} and some of the midway targets simultaneously. Thus to enhance the parallel processing capability of the UAV swarm, a parallel-coverage mechanism is proposed.

Next, we would elaborate on the three algorithms and a mechanism. For the convenience of description, we would ignore the notation of communication topology $\widetilde{E}$ in the input of the algorithms, which is just given at the beginning. Additionally, in the following sections, a set of UAV positions would be denoted as $P_U^{\cdot} = \{p_u^{\cdot} | \forall u \in U\}$, where the superscript $\cdot$ is identification text (P_U is still the set of current UAV positions as aforementioned). For example, "d" represents the set of destinations, and "i" is used for the set of initial positions.

4.2 Who to Cover: KMA Algorithm

In this section, we present the KMA algorithm to elect a UAV named the root of UAV to cover the newly appeared target τ_{new}.

The main idea of KMA is to find a UAV that can cover τ_{new} while trying not to disturb UAVs that are covering target(s). More specifically, for a UAV u_a that is covering at least one target, to preserve its coverage for the target(s), we expect that it could remain stationary, and, thus, it is called an anchor. However, as other UAVs might move cooperatively to cover τ_{new}, to preserve the communication topology, an anchor may have to move. Therefore, we try to elect the UAV that could cover τ_{new} while the number of disturbed anchors is minimized.

To determine whether an anchor u_a would be disturbed by a UAV u to cover τ_{new}, the UAVs simply bid the Euclidean distance between τ_{new} and u_a with the theoretical maximum distance between u and u_a after broadcasting their number of covering targets.[2] That is, if $d_t(\tau_{\text{new}}, p_{u_a}^i) \leq r_s + r_c \cdot h(p_u^i, p_{u_a}^i)$, then it is approximated that the u_a would not be disturbed by u to cover τ_{new}, where $h(u, u_a)$ is the hop count between u and u_a.

4.3 Where to Go: TP3 Algorithm

In this section, we propose a distributed algorithm named TP3 to find the UAV destinations such that (1) the newly appeared target τ_{new} is covered by the root UAV u_{root} and (2) and the communication topology is preserved.

The main process of the TP3 algorithm is to initialize the destination set P_U^d to the initial position set P_U^i and iteratively update the destinations for needed UAVs. For the sake of description, both in this section and the proof section, we will use "move"-related term to indicate the search and update process for a UAV u's destination p_u^d. For example, "u tries to move closer to target τ" means u tries to search for a position nearer to τ and update p_u^d to the position if it is successfully found.

[2]This process could be implemented by adopting existing distributed election algorithms like References [3, 23].

ALGORITHM 1: TP3 Algorithm

Input: Set of initial UAV positions P_U^i, Target τ_{new}, Root UAV u_{root}
Output: Set of UAV destinations P_U^d

$P_U^d \leftarrow P_U^i$, $t \leftarrow$ current time;
Execute TPUB($u_{\text{root}}, \tau_{\text{new}}, \tau_{\text{new}}$);
Function TPUB(u, τ, u_p)**:**
  $t \leftarrow$current time;
  if u *is* u_{root} *and* $d_t(\tau_{new}, p_u^d) \leq r_s$ **then**
    return
  end if
  $p_{\text{N}} \leftarrow \text{NPTN}(\widetilde{N}_u, \eta_{u,t}, \tau)$;
  if $d_t(\tau_{new}, p_N) - d_t(\tau_{new}, p_u^d) < 0$ **then**
    $p_u^d \leftarrow p_{\text{N}}$;
    if u *is* u_{root} **then**
      TPUB(u_{root},τ_{new}, τ_{new});
    else
      reply to u's parent u_p;
    end if
  else
    $b \leftarrow$ bottleneck(u, τ);
    if $b \in \eta_{u,t}$ **then**
      stop covering b;
    else
      TPUB(b, u_p, u);
      Wait till b replies;
    end if
    TPUB(u, τ, u_p);
  end if
End Function
return P_U^d

Moreover, under such a narrative, the "current position" for a UAV mentioned later would be its latest destination instead of its real position.

As no cyclic path exists in the communication topology, a single tree rooted at the selected UAV u_{root} would be built by conducting Breadth-First-Search. The main idea of the TP3 algorithm is to let UAVs at the top level first try to move closer to their targets, either to cover τ_{new} (for u_{root}) or to help its parent move closer to the target (for non-root UAVs). As the connections to $\widetilde{E}$-neighbors need to be maintained, a UAV might not be able to move closer to its target, and then it would ask its children for help. In other words, UAVs conduct a top-down process to require help from their children if necessary, and children would move and reply to its parent in a bottom-up manner.

Obviously, the process of TP3 described above can be implemented in a distributed manner. For ease of description, we describe TP3 in Algorithm 1 in detail in a centralized manner. Specifically, we use a general function TPUB(u, τ, u_p), whose name indicates Topology-Preserving UAV Behaviour, to describe in the TP3 algorithm how a UAV u moves closer to the point τ given by another UAV u_p called its parent. On the appearance of a new target τ_{new}, we take the UAV assigned to τ_{new} as the root UAV, whose parent is defined as τ_{new} for convenience of describing the algorithm. The root UAV is denoted by u_{root}, and others are called *non-root UAVs*. UAV u_{root} tries

ALGORITHM 2: Functions used in TP3

Function NPTN($\widetilde{N}_u, \eta_u, \tau$)**:**
  $t \leftarrow$current time, $p \leftarrow 0$;
  $I_{\text{cirle}}, I_{\text{line}}, i.sources \leftarrow \emptyset$;
  $\widetilde{N}_{\text{circle}} \leftarrow$ All neighbor and coverd target circles;
  if τ *is within all circles in* $\widetilde{N}_{circle}$ **then**
    return τ
  end if
  Calculate set $I_{\text{circle}}, I_{\text{line}}$ for u;
  $I \leftarrow \{i | i \in I_{\text{circle}} \cup I_{\text{line}}, d_t(i, nc.center) \leq nc.radius, \forall nc \in \widetilde{N}_{\text{circle}}\}$;
  $p \leftarrow \underset{i \in I}{\text{argmin}}\, d_t(i, \tau)$;
  Save the sources of p in set $p.sources$;
  return p;
End Function
Function bottleneck(p_n, τ)**:**
  /* $un(\cdot)$ *indicates vector unitization.* */
  $b \leftarrow \underset{b \in p_n.sources}{\text{argmin}} \; un(\tau.\vec{p}os - p_n.\vec{p}os) \cdot un(b.\vec{p}os - p_n.\vec{p}os)$;
  return b;
End Function

to move closer to cover the target τ_{new}. However, to maintain the feasibility of the communication topology, it is possible that u_{root} cannot directly move to target τ_{new} without other non-root UAVs' movement. Thus, we design a method called *NPTN* that can find the position nearest to a target τ that a UAV u could move to without disturbing neighboring UAVs and covered targets. If the point found by NPTN is not closer to its target compared to its current position, then the root UAV currently falls in an unmovable status. To tackle this case, we further propose the *bottleneck* function to find the "bottleneck" UAV u_b among u_{root}'s children, i.e., its $\widetilde{E}$-neighbors. Then an unmovable UAV will require the "bottleneck" UAV to move and help it get out of unmovable status. The "bottleneck" UAV will either respond to the request immediately or after iteratively executing TPUB(·) to ask for other UAVs' assistance. The root UAV then can reconsider its situation and tries to move closer to the new target. The process repeats until the target τ_{new} is covered by the root UAV. Figure 3 gives an example of the distributed motion plan.

Then we introduce more details on the function NPTN and bottleneck. Before that, we define two kinds of circles.

Definition 4.1 (Neighbor Circle). For UAV $u, u_{nb} \in U$, if u_{nb} is an $\widetilde{E}$-neighbor of u, then the circle centered at UAV u_{nb}'s current position with radius r_c is called u's neighbor circle.

Definition 4.2 (Target Circle). For a target $\tau \in T$, the circle centered at τ's current position with radius r_s is called a target circle.

Function NPTN works as follows. Given a UAV u, it takes $\widetilde{N}_u, \eta_{u,t}, \tau$ as the input, where $\widetilde{N}_u$ is the set of u's $\widetilde{E}$-neighbors, $\eta_{u,t}$ is the set of targets covered by u at time t, and τ is the destination that u needs to move to. First, it computes the movable area of u, which is the intersection area between all the neighbor circles of $\widetilde{N}_u$ and target circles of targets in $\eta_{u,t}$. If τ is in the movable area, then NPTN returns τ as the destination p_n. Otherwise, NPTN finds a point within the movable area that minimizes its distance to τ_{new}. Such kind of point can only be on the intersection points of

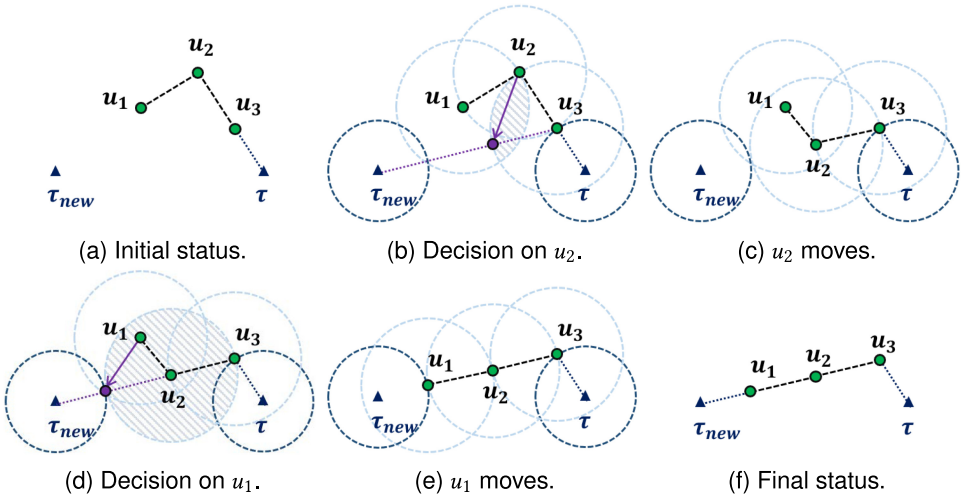


Fig. 3. Example of running TP3 algorithm to cover τ_{new}. (a) Initially target τ has been covered by u_3. The fixed communication topology $\widetilde{E}$ contains link (u_1, u_2) and (u_2, u_3). (b) u_1 is the root-UAV in charge of covering τ_{new}, while it is constrained by u_2 and cannot move closer to τ_{new}. Hence, u_1 selects u_2 as the bottleneck and asks it for assistance. Then u_2 finds the nearest pint p_n (to τ_{new}) within its movable area without disturbing u_1, u_3 (by calling NPTN($\widetilde{N}_{u_2}, \eta_{u_2}, \tau_{\text{new}}$)). (c) u_2 moves to the calculated point and give feedback to u_1. (d) u_1 again tries to find its nearest point to τ_{new}. (e) u_1 moves to the found point. (f) Finally, u_1 covers target τ_{new}.

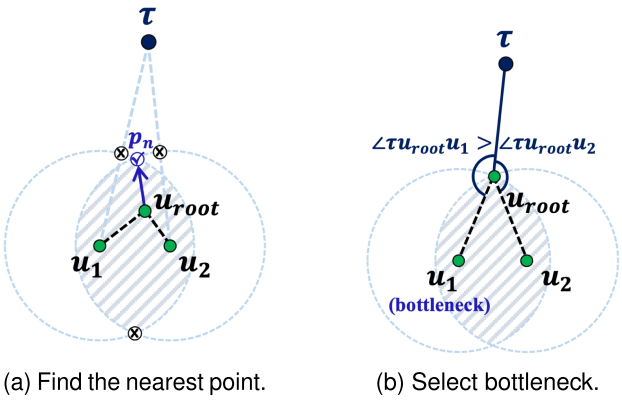


Fig. 4. Example of function NPTN and bottleneck, where shadow area is the movable area of u_{root}. (a) To cover τ, UAV u_{root} initially finds the point p_n nearest to τ it could reach without disturbing its children u_1 and u_2. (b) After moving to the found point, it becomes unmovable, and thus it founds u_1 as the bottleneck (that hinders it).

neighbor circles, denoted by I_{circle}, or on the intersection points between a neighbor circle $\odot(u', r_s)$ centered at u' with radius r_s where $u' \in \widetilde{N}_u$ and a line segment $\overline{\tau u'}$, denoted by I_{line}. Among these points, it just chooses the one within the movable area that is the closest to τ as the returned point p_n. Figure 4(a) shows an example of the computed nearest point. Note that we save the centers of neighbor circles and target circles that contain point p_n into a set *sources* for further usage.

When u is currently unmovable, it calls Function *bottleneck* to find the "bottleneck" UAV b for assistance, which is the one that achieves the largest angles generated by the sources of p_n and target τ. For example, as shown in Figure 4(b), u_1, u_2 are the sources of p_n. We compare the

ALGORITHM 3: ULMC

Input: Set of UAV initial position P_U^i, Set of UAV destinations P_U^d
Output: Motion plans M

$M \leftarrow \emptyset$;
/* UAVs bid to decide the moving time and departure time. */
$t^{\text{move}} \leftarrow \max_{u \in U} \frac{d(p_u^i, p_u^d)}{v_u^m}$, $t^{\text{dep}} \leftarrow \max_{u \in U} t_u^{\text{dep}}$;
foreach $u \in U$ **do**
  $M \leftarrow M \cup \{(t^{\text{dep}}, \frac{d(p_u^i, p_u^d)}{t^{\text{move}}}, u, p_u^d)\}$
end foreach
return M

magnitude of two angles $\angle \tau u_{\text{root}} u_1$ and $\angle \tau u_{\text{root}} u_2$, and choose u_1 as bottleneck. If a covered target is regarded as the bottleneck, i.e., $b \in \eta_{u,t}$, then u will stop covering it and reconsider its situation. Otherwise, if any of u's $\widetilde{E}$-neighbors is selected as the bottleneck, then u will ask the bottleneck UAV to move for assistance.

Actually, the bottleneck UAV is a non-root UAV and behaves nearly the same as the root UAV. While the root UAV wants to cover a target τ_{new}, a non-root UAV aims at being closer to the point given by its parent. Thus, we can also call NPTN to iteratively guide its movement.

Note that if any obstacle exists in the environment, then TP3 can still work by subtracting an "obstacle circle" while calculating a UAV's movable area, where the obstacle circle is centered at the position of the obstacle with the radius of the safe distance between a UAV and the obstacle. If subtracting such an obstacle circle makes the UAV fall into an unmovable situation, then the UAV can also request help from its children or stop covering certain target(s) as described earlier, which will help it move toward its target position while bypassing the obstacle.

4.4 How to Move: ULMC Algorithm

In this section, a distributed motion control algorithm named ULMC is proposed to simultaneously drive UAVs to a feasible destination set while preserving the communication topology.

The main idea of ULMC is to ask UAVs to move along a straight line to their destinations at constant speeds, and by carefully setting each UAV's speed, the communication topology could be preserved.

As presented in Algorithm 3, initially u calculates its own minimum moving time $\frac{d(p_u, p_u^d)}{v_u^m}$ according to the maximum speed v_u^m and determines its earliest departure time t_u^{dep}. Subsequently, UAVs bid their own moving times and departure times, and the bid with maximum one t^{move} and bid with the latest one t^{dep} would win.[3] Consequently, each UAV u sets its speed as $\frac{d(p_u, p_u^d)}{t^{\text{move}}}$ and moves simultaneously to destinations linearly in constant speed. Also, since the path and speed of each UAV are priorly determined before the actual movement, to avoid inter-UAV collision, UAVs can extra share their origination and destination while bidding. Then UAVs can adaptively adjust their flying height or speed to avoid collisions at the path intersections with others.

4.5 Parallel-coverage Mechanism

The algorithms above enable the UAV swarm to cover a newly appeared target while keeping as many existing targets covered as possible. In this section, a parallel-coverage mechanism is

[3]The bidding process could be implemented by adopting existing distributed election algorithms [3, 23].

ALGORITHM 4: Parallel-Coverage

Input: Current target τ_{now} to be covered, current destinations P_U^d

$\tau_U^d \leftarrow \{\tau_{\text{now}}\}$;
foreach *midway target* τ_{mid} **do**
  $u_{\text{root}} \leftarrow$KMA($P_U^d$, τ_{mid});
  $P_U^{\text{mid}} \leftarrow$TP3($P_U^d$, τ_{mid}, u_{root});
  $P_U^{\text{next}} \leftarrow P_U^{\text{mid}}$;
  foreach $\tau \in \tau_U^d$ **do**
    if $d(p_u^{next}, \tau) > r_c, \forall u \in U$ **then**
      $P_U^{\text{next}} \leftarrow P_U^d$;
      break;
    end if
  end foreach
  if $P_U^{next} = P_U^{mid}$ **then**
    $\tau_U^d \leftarrow \tau_U^d \cup \{\tau_{\text{mid}}\}$;
    $P_U^d \leftarrow P_U^{\text{next}}$;
    $M \leftarrow$ULMC(P_U, P_U^{d});
    Execute M;
  end if
end foreach

presented to further enhance the parallel processing capability of the UAV swarm, considering that new targets might appear midway while UAVs are moving to cover the currently appeared target τ_{now}. In the parallel-coverage mechanism, UAVs would try their best to cover the current target and the midway targets simultaneously rather than dealing with τ_{now} only, which would further enhance the parallel processing capability of the UAV swarm. The mechanism is illustrated in Algorithm 4.

Specifically, the parallel-coverage mechanism works as follows. When the first midway target appears while UAVs are moving to cover τ_{now}, first, the KMA and TP3 algorithms are called to get a new destination set P_U^{mid} for the midway target if τ_{now} is also being covered in P_U^{mid}, and UAVs would change their destinations to P_U^{mid} to cover τ_{now} and the midway target simultaneously.

Considering that multiple midway targets might occur, a target set τ_U^d is maintained to remember the targets UAVs currently plan to cover (including τ_{now} and some of the midway targets). If and only if the destination set for a midway target τ_{mid} enables all targets in the τ_U^d to be covered, then τ_{mid} would be added to τ_U^d and the destinations of UAVs would be changed. Finally, all targets added to τ_U^d could be covered simultaneously with the new destination set, where the parallel processing capability of the UAV swarm is further enhanced.

5 TOPOLOGY-MANAGEMENT PROTOCOL

Due to reasons like battery shortage, the UAV swarm occasionally has to face dynamical UAV leave and arrival in varying swarm management paradigms. To properly manage the maintained communication topology, in this section we propose the topology-management protocol to deal with both UAV leave and arrival. More concretely, this protocol works as follows.

— As a UAV u proactively leaves the swarm or passively fails, its $\widetilde{E}$-neighbors are informed or detect its failure, then they heal the broken communication topology and update their routing tables accordingly. This process is controlled by a mechanism named Topology-Recovery.

— As a UAV u arrives and decides to join the swarm, it establishes a link with any of the UAV[4] u' already in the swarm. As the communication link is built, u can know other members in the swarm and communicate with them via u'. Then it informs other members in the swarm so that others know u has joined the swarm and can communicate with it via u'.
— After arrivals or leaves of multiple UAVs, the design of the initial communication topology can be disrupted. Also, since the targets appear in an online manner, there can be a need to adapt the preserved communication topology. Specifically, when the reconfiguration is triggered at a UAV u, where the new communication topology to be preserved is given,[5] the UAV u determines the final position of each UAV in the swarm and the way the UAVs move to it according to its computing power. This process is controlled by a mechanism named Topology-Reconfiguration.

Since the situation that a UAV arrives is obviously easy to deal with, next we will elaborate on the Topology-Recovery mechanism that works when UAVs leave the swarm and the Topology-Reconfiguration mechanism that works when the UAV swarm needs to adapt the communication topology.

5.1 Topology-Recovery Mechanism

As a UAV leaves the swarm, since the communication topology is acyclic, a hole occurs, i.e., global connectivity over the communication topology is broken. To mitigate this problem, one solution is to use backup routing protocol [65] until another UAV or the left UAV itself joins the swarm and fills this hole. However, to achieve effective and reliable network performance brought by static routing, it is desirable to re-establish a new communication topology and continue using the static routing protocol. Thus, as presented in Algorithm 5, we propose a Topology-Recovery mechanism for remaining UAVs to repair the broken communication topology.[6] An example of the mechanism can be found at Figure 5.

First, the main idea of the mechanism is to ask the $\widetilde{E}$-neighbors of the left UAV u_l to approach its last position in the swarm. Therefore, before repairing the broken communication topology as u_l leaves, to cooperate to repair the communication topology without communication, their $\widetilde{E}$-neighbors should be capable of knowing each other. Thus, as u_l decides to leave the swarm, our mechanism asks u_l to tell each its $\widetilde{E}$-neighbor its current position, the IDs, and network addresses of all u_l's $\widetilde{E}$-neighbors.

Second, when knowing each other, those UAVs ($\widetilde{E}$-neighbors of u_l) are then required to gather toward u_l's last position $p_{u_\mathrm{l}}^{\mathrm{last}}$ so that they are closer enough to establish new links. To be specific, when u_l fails, the original swarm breaks into several separate UAV groups, each of which contains an $\widetilde{E}$-neighbor u_fn of u_l. Those UAV groups are inner-connected (and acyclic). Therefore, to make the swarm connected again, inter-group links must be established. The main idea of our mechanism is to make u_fn in each group to gradually move toward $p_{u_\mathrm{l}}^{\mathrm{last}}$ while other UAVs in this group assist u_l by TP3 algorithm to keep inner-group connectivity.

Third, as $\widetilde{E}$-neighbors of u_l are trying to distributedly gather toward $p_{u_\mathrm{l}}^{\mathrm{last}}$ and establish links with each other at the meantime, when should they stop and who do they need to build a link with? In our mechanism, those UAVs elect the one $u_\mathrm{fn}^{\mathrm{m}}$ who owns the maximum ID to replace u_l. More

[4]By "u establishes a link with u'" we mean that they both receive the broadcast message from each other, so they could know they are within each other's communication range and put (u, u') into the communication topology $\widetilde{E}$ (its own copy).
[5]Since there can be different motivations/objectives for the topology reconfiguration, we assume the trigger mechanism and methods of new communication topology have been installed for the UAV swarm to support general scenarios. Our protocol helps the UAV to move to achieve the desired communication topology.
[6]The recovery process can also be found in the demo https://www.dropbox.com/s/1cip2pn4yszss31/demo.mp4?dl=0

ALGORITHM 5: Topology-Recovery

Input: left UAV u_{l}, u_{l}'s last-sent position $p_{u_{\mathrm{l}}}^{\mathrm{last}}$, a small positive number $\epsilon < 1$

foreach $u_{fn} \in \widetilde{N}_{u_l}$ **do**
  $\widetilde{N}_{u_{\mathrm{fn}}} \leftarrow \widetilde{N}_{u_{\mathrm{fn}}} \setminus \{u_{\mathrm{l}}\}$;
  $u_{\mathrm{fn}}^{m} = \mathrm{argmax}_{u \in \widetilde{N}_{u_{\mathrm{l}}}} \mathrm{ID}(u)$;
  $\widetilde{N}_{u_{\mathrm{fn}}}^{\mathrm{new}} \leftarrow \emptyset$;
  if $u_{fn} = u_{fn}^{m}$ **then**
    $\widetilde{N}_{u_{\mathrm{fn}}}^{\mathrm{new}} \leftarrow \widetilde{N}_{u_{\mathrm{l}}} \setminus \{u_{\mathrm{fn}}\}$;
  else
    $\widetilde{N}_{u_{\mathrm{fn}}}^{\mathrm{new}} \leftarrow \{u_{\mathrm{fn}}\}$
  end if
  dir $\leftarrow un(p_{u_{\mathrm{fn}}} - p_{u_{\mathrm{l}}}^{\mathrm{last}})$;
  Start broadcasting message within own ID periodically;
  while *True* **do**
    $\widetilde{N}_{u_{\mathrm{fn}}}^{\mathrm{broad}} \leftarrow$IDs of UAVs contained in received broadcast messages;
    for $u \in \widetilde{N}_{u_l}^{new} \cap \widetilde{N}_{u_{fn}}^{broad}$ **do**
      $\widetilde{E} \leftarrow \widetilde{E} \cup \{(u, u_{\mathrm{fn}})\}$;
    end for
    $\widetilde{N}_{u_{\mathrm{fn}}}^{\mathrm{new}} \leftarrow \widetilde{N}_{u_{\mathrm{fn}}}^{\mathrm{new}} \setminus \widetilde{N}_{u_{\mathrm{fn}}}^{\mathrm{broad}}$;
    if $\widetilde{N}_{u_{fn}}^{new} = \emptyset$ **then**
      break;
    end if
    Move to $p_{u_{\mathrm{fn}}} + \epsilon \cdot$ dir with TPMP algorithm;
  end while
  End broadcasting and update routing paths;
end foreach

concretely, take $\mathrm{ID}(\cdot)$ to denote the ID of a UAV, $u_{\mathrm{fn}}^{m} = \mathrm{argmax}_{u \in \widetilde{N}_{u_{\mathrm{l}}}} \mathrm{ID}(u)$ can be elected without negotiation, since they all maintain a two-hop $\widetilde{E}$-neighbor set and know each other's ID. Then $u_{\mathrm{fn}}^{\mathrm{m}}$ needs to keep moving until it has established links with all others, i.e., UAVs in set $\widetilde{N}_{u_{\mathrm{l}}} \setminus \{u_{\mathrm{fn}}^{\mathrm{m}}\}$, and these UAVs can stop as long as they have established a link with $u_{\mathrm{fn}}^{\mathrm{m}}$.

Fourth, as a new communication topology has been built, to enable static routing over it, the communication/routing paths of UAVs might need to be updated. According to the variety of the communication topology, we propose that for the communication path between u_o and u_d:

— If the original path goes through $u_{\mathrm{fn}}^{\mathrm{m}}$ and u_{l} like $(u_o, \ldots, u_{\mathrm{fn}}^{\mathrm{m}}, u_{\mathrm{l}}, \ldots, u_d)$, then just remove u_{l} and save $(u_o, \ldots, u_{\mathrm{fn}}^{\mathrm{m}}, \ldots, u_d)$ as its new communication path.
— If the original path goes through $u_{\mathrm{fn}} \in \widetilde{N}_{u_{\mathrm{l}}} \setminus \{u_{\mathrm{fn}}^{\mathrm{m}}\}$ and u_{l} like $(u_o, \ldots, u_{\mathrm{fn}}, u_{\mathrm{l}}, \ldots, u_d)$, then just replace u_{l} with $u_{\mathrm{fn}}^{\mathrm{m}}$ and save $(u_o, \ldots, u_{\mathrm{fn}}, u_{\mathrm{fn}}^{\mathrm{m}}, \ldots, u_d)$ as its new communication path.
— Otherwise, the path does not need to change.

Naturally, from the update rules of the communication/routing paths, the following property of the recovery mechanism can be inferred.

Theorem 2. *When a UAV of the swarm fails, with our recovery mechanism, the hop count of the communication/routing path for each UAV pair does not increase.*

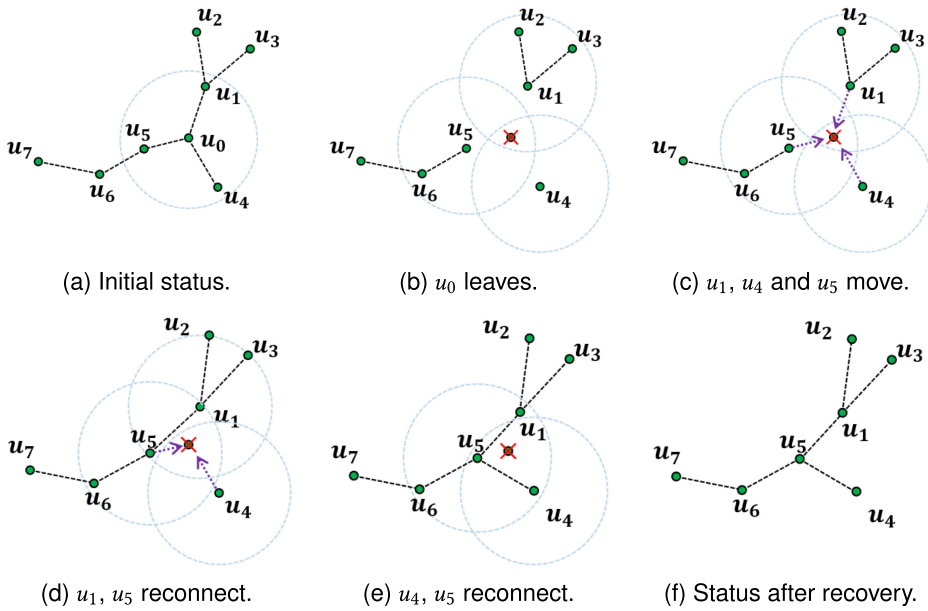


Fig. 5. Example of topology recovery mechanism. (a) Initially, the communication topology is feasible; u_1, u_4, and u_5 communicate via u_0. (b) u_0 fails or decides to leave, but u_1, u_4, and u_5 are not within each other's communication range. (c) u_1, u_4, and u_5 are informed by u_0 its leaving or detecting u_0's failure by periodically sent hello message, and then they gradually move toward the position that u_0 last reports and try to build communication links with (reconnect) each other. (d) As u_1 and u_5 find that they have stepped into each other's communication range by receiving periodically sent broadcast messages, they add each other's ID into their own $\widetilde{E}$-neighbor set and update the routing table accordingly. Since the ID of u_5 is larger, it keeps moving and tries to reconnect with u_4. (e) As u_5 and u_4 discover each other within their communication range, they establish the communication link between them and update accordingly. (f) The communication topology has been completely recovered, and the UAV swarm is ready for its undone tasks.

This theorem is simple but important. In our system model, the original communication topology is given by the user. This means that the communication topology might be well designed to satisfy certain QoS guarantees. Meanwhile, it is well known that hop count is a significant metric of network performance like end-to-end delay [67] and stability [56]. Therefore, our Topology-Recovery mechanism can not only repair the broken connectivity but also maintain some important network properties of the original swarm.

5.2 Topology-reconfiguration Mechanism

When the reconfiguration mechanism is triggered at a UAV u, the new communication topology to be preserved is given. Then the UAV u first requests the position of each UAV in the swarm. Subsequently, the UAV u checks whether the new communication topology is feasible for the current positions of UAVs or not. If the new communication topology is feasible, then it can be configured directly by broadcasting the new routing entries to each UAV. If the new communication topology is not feasible, then the UAVs need to move to new positions to enable it. Two mechanisms for the movement are provided as follows, where the UAV u can adaptively choose based on its available computation resources.

The first mechanism is simple and does not require much computation resources. To be concrete, the UAV u calculates the average UAV positions, denoted by $\overline{p}$, and broadcasts it to all UAVs

with the new communication topology. Subsequently, all UAVs move toward $\overline{p}$ by the ULMC algorithm. During this process, called the shrink process, all UAVs periodically broadcast HELLO messages and respond to HELLO messages sent by their neighbors in the new communication topology. Once each UAV has been established with its new neighbors, all UAVs stop moving and the shrink process ends. We will show in Theorem 4 that if initially and lastly the same communication topology is preserved, it is also preserved during the movement process. Therefore, we can easily conclude that during the shrink process, the UAV swarm is globally connected as the old communication topology is always preserved.

The second mechanism can lead to a shorter moving distance, but it requires more computation resources. To be concrete, we propose to use a solver to derive the new UAV positions where both the new and old communication topology are preserved, and the cumulative distance for the UAV to move from the current positions is minimized. Specifically, let p_u be the current position of the UAV u, $\tilde{E}$ be the communication topology, p'_u be the position of the UAV u to be solved, and $\tilde{E}'$ be the given new communication topology, the optimization problem can be formulated as follows:

$$\begin{aligned} &\min_{p'_u} \sum_{u \in U} \|p_u - p'_u\|_{L_1} \\ &\textit{subject to } \|p_{u_1} - p_{u_2}\|_{L_2} \leq r_c, \forall (u_1, u_2) \in \tilde{E} \cup \tilde{E}', \end{aligned} \tag{3}$$

where $\|\cdot\|_{L_1}$ and $\|\cdot\|_{L_2}$ are respectively L1-norm and L2-norm. Note that some solvers can deal with quadratic constraints directly, e.g., the Gurobi solver. Also, one can also slack the Euclidean distance constraints to linear constraints as Reference [38] does.

After obtaining the new positions of UAVs by the solver, the UAV u can broadcast the new positions to UAVs with the communication topology. Then the UAVs can follow the ULMC algorithm to move to their new positions, where the new communication topology is feasible to be configured. During this process, the UAV swarm is globally connected, since it also keeps the feasibility of the old communication topology.

Moreover, if the positions of UAVs under the new communication topology are also exogenously provided, then the swarm should follow the Topology-Reconfiguration mechanism to reconfigure the network before applying the ULMC algorithm to move to the given positions. This ensures that the global connectivity of the UAV network is maintained throughout the movement. If the UAVs were to move directly to the given positions, then the global connectivity of the UAV network might be compromised.

6 THEORETICAL ANALYSIS

In this section, we prove the theoretical guarantees of the proposed solution. We will first show that the communication topology is preserved during movement under the proposed motion planning, i.e., proper distance between each pair of UAVs in the communication topology $\widetilde{E}$ is maintained. Then we will further prove its convergence, i.e., the newly appeared targets always get covered.

6.1 Feasibility of Communication Topology

As the final destinations of UAVs are calculated by the TP3 algorithm and the movement from initial positions to the destinations is controlled by the ULMC algorithm, we will respectively prove that in both algorithms, the feasibility of communication topology is preserved.

Theorem 3. *The communication topology is feasible if UAVs are in the destinations planned by the TP3 algorithm.*

Proof. The destination of each UAV is calculated by the NPTN function, which is on or inside the movable area intersected by its neighbor circles. This implies that the communication topology

is preserved after every update for the UAV destination. As the update process ends, the UAV destinations are fixed, thus, the communication is feasible for UAVs in their destinations. □

Theorem 4. *Given each UAV's destination, assume that the communication topology is feasible for both UAVs' initial positions and destined positions. If UAVs run the ULMC algorithm, then the communication topology would not be broken during their movements.*

Proof. For any UAV pair $(u_a, u_b) \in \widetilde{E}$, assume that A_1, B_1 is respectively the initial position of u_a, u_b, A_2, B_2 the destined position of u_a, u_b, and A, B the position at any time slot during the movement. Then we could denote $\overrightarrow{AB}$ as the linear combination of $\overrightarrow{A_1B_1}$ and $\overrightarrow{A_2B_2}$, and by calculating the magnitude of $\overrightarrow{AB}$ we find that $|\overrightarrow{AB}| \leq \max\{|\overrightarrow{A_1B_1}|, |\overrightarrow{A_2B_2}|\}$.

Initially, since the communication topology is feasible when UAVs are in their initial positions and destined positions, we could get that

$$|\overrightarrow{A_1B_1}| \leq r_c, |\overrightarrow{A_2B_2}| \leq r_c. \tag{4}$$

Next, we prove that the connection between the UAV pair at any time would always be maintained, which indicates that the communication topology is maintained feasible during movement. This could be formalized as follows:

$$|\overrightarrow{AB}| \leq r_c. \tag{5}$$

Since u_a, u_b move straightly respectively from A_1, B_1 to A_2, B_2, by letting $k_a, k_b \in [1, +\infty)$, we could assume that

$$\begin{aligned} \overrightarrow{A_1A_2} &= k_a\overrightarrow{A_1A} \\ \overrightarrow{B_1B_2} &= k_b\overrightarrow{B_1B}. \end{aligned} \tag{6}$$

Meanwhile, u_a, u_b move in constant speeds while leaving A_1, B_1 at the same time and arriving A_2, B_2 at the same time, which indicates that

$$k_a = k_b := k, k \in [1, +\infty). \tag{7}$$

The Euclidean distance between u_a and u_b could be denoted as $|\overrightarrow{AB}|$, and the vector $\overrightarrow{AB}$ could be calculated by

$$\overrightarrow{AB} = \overrightarrow{AA_1} + \overrightarrow{A_1B_1} + \overrightarrow{B_1B} \tag{8}$$

and

$$\begin{aligned} \overrightarrow{AB} &= \overrightarrow{AA_2} + \overrightarrow{A_2B_2} + \overrightarrow{B_2B} \\ &= (\overrightarrow{AA_1} + \overrightarrow{A_1A_2}) + \overrightarrow{A_2B_2} + (\overrightarrow{B_2B_1} + \overrightarrow{B_1B}) \\ &\overset{(6)(7)}{=} (\overrightarrow{AA_1} + k\overrightarrow{A_1A}) - k\overrightarrow{B_1B} + (\overrightarrow{B_2B_1} + \overrightarrow{B_1B}). \end{aligned} \tag{9}$$

By adding Equations (8) and (9), $\overrightarrow{AB}$ could be denoted as

$$\begin{aligned} \overrightarrow{AB} &= (1 - \frac{k}{2})(\overrightarrow{B_1B} - \overrightarrow{A_1A}) + \frac{1}{2}(\overrightarrow{A_1B_1} + \overrightarrow{A_2B_2}) \\ &\overset{(6)(7)}{=} (\overrightarrow{B_1B_2} - \overrightarrow{A_1A_2}) + \frac{1}{2}(\overrightarrow{A_1B_1} + \overrightarrow{A_2B_2}) \\ &= (\overrightarrow{B_1A_1} + \overrightarrow{A_1A_2} + \overrightarrow{A_2B_2} - \overrightarrow{A_1A_2}) + \frac{1}{2}(\overrightarrow{A_1B_1} + \overrightarrow{A_2B_2}) \\ &= \frac{1}{k}\overrightarrow{A_2B_2} + (1 - \frac{1}{k})\overrightarrow{A_1B_1}. \end{aligned} \tag{10}$$

Thus the Euclidean $|\overrightarrow{AB}|$ distance between u_a and u_b could be denoted as

$$\begin{aligned}|\overrightarrow{AB}| &= \sqrt{|\overrightarrow{AB}|^2} = \sqrt{\overrightarrow{AB}\cdot\overrightarrow{AB}} \\ &\overset{(10)}{=} \sqrt{\frac{|\overrightarrow{A_2B_2}|^2}{k^2} + (1-\frac{1}{k})^2|\overrightarrow{A_1B_1}|^2 + (\frac{2}{k}-\frac{2}{k^2})\overrightarrow{A_1B_2}\cdot\overrightarrow{A_2B_2}}.\end{aligned} \tag{11}$$

Equation (11) indicates that $f:\frac{1}{k}\to|\overrightarrow{AB}|^2$ is a univariate quadratic function, and we could get that the coefficient of $(\frac{1}{k})^2$ is

$$\begin{aligned}&|\overrightarrow{A_1B_1}|^2+|\overrightarrow{A_2B_2}|^2-2\overrightarrow{A_1B_1}\cdot\overrightarrow{A_2B_2} \\ \geq&|\overrightarrow{A_1B_1}|^2+|\overrightarrow{A_2B_2}|^2-2|\overrightarrow{A_1B_1}||\overrightarrow{A_2B_2}| \\ =&(|\overrightarrow{A_1B_1}|-|\overrightarrow{A_2B_2}|)^2>0.\end{aligned} \tag{12}$$

It indicates that the parabola of $f(\frac{1}{k})$ opens upwards. Moreover, by Equation (7) we know that $\frac{1}{k}\in(0,1]$, therefore

$$\begin{aligned}|\overrightarrow{AB}| &= \sqrt{|\overrightarrow{AB}|^2} \\ &\leq \sqrt{\max\{f(\frac{1}{k}\to 0), f(\frac{1}{k}\to 1\})} \\ &= \sqrt{\max\{|\overrightarrow{A_1B_1}|^2, |\overrightarrow{A_2B_2}|^2\}} \\ &\overset{(4)}{\leq} \sqrt{\max\{r_c^2, r_c^2\}} = r_c.\end{aligned} \tag{13}$$

Hence for any UAV pair $(u_a,u_b)\in\widetilde{E}$, if they follow the ULMC method, then their Euclidean distance would not be greater than the communication range r_c at any time, i.e., the communication topology would always be feasible. □

6.2 Convergence of Coverage

As UAVs move in a distributed and cooperative manner in the motion plan, we need to show that with TPMP the UAV swarm can converge to cover targets. In TPMP, the destination set is calculated by TP3, and the movement is guided by ULMC. Obviously, ULMC converges to achieve the destined positions. Thus, to prove the convergence of coverage. Next, we propose that the TP3 algorithm converges to cover the newly appeared target, i.e., it would successfully return a destination set for the UAV swarm where the newly appeared target is covered.

To be specific, we will show that in the TP3 algorithm it will not cause a deadlock or infinitely wait for others' assistance, respectively. Specifically, we first prove that any UAV u would not be its bottleneck UAV's bottleneck and then show that the root UAV would keep moving closer to the new target until the target gets covered.

To begin with, we introduce a special line, called bottleneck boundary, that can help us figure out the relationship between an unmovable UAV u and its bottleneck (see Figures 8 and 9 for reference).

Definition 6.1 (Bottleneck Boundary). Assume that UAV u is unmovable and its target point is τ. The bottleneck boundary of u is the perpendicular of $\overline{\tau u}$ that goes across point u. We say that a point p_x is inside the bottleneck boundary of u *iff* $\overline{p_x u}$ has no intersection point with the boundary.

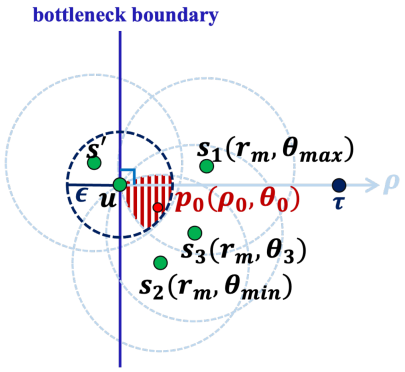


Fig. 6. Example when every source's center of an unmovable UAV u is inside the bottleneck boundary, and there might be a circle whose center $s' \in S_L$. Under this situation, we could always find a point p_0 that is reachable and closer to u's target point τ than u's current position.

Next, we will show an important property of the bottleneck boundary following the definition, i.e., the bottleneck must be on or outside the boundary. To formally prove this property, we will first show that the source centers of an unmovable UAV cannot be all inside the bottleneck boundary. Then, based on this property, we show the bottleneck would not be inside the bottleneck boundary.

First, we show that at least one source center lies outside the bottle boundary.

Lemma 6.1. *Assume that UAV u is unmovable, and we denote the set containing its sources' centers by $S_K = \{s_1, s_2, \ldots, s_k\}$. Then there exists a center s_i $(1 \le i \le k)$ that is outside the bottleneck boundary of u.*

Proof. We mix the centers of u's neighbor circles and centers of covered target circles in a set and divide them into two categories by whether it is a source's center. We denote the set of source centers by S_K, the set of centers that do not belong to any source by S_L, i.e., $S_L = \widetilde{N}_u \cup \eta_u \setminus S_K$.

We prove the lemma by contradiction. Specifically, assume that all source centers are inside the bottleneck boundary. Then, in such a situation, we can always find a point nearer to the target in u's movable area, i.e., area intersected by u's all neighbor circles and covered target circles even if $S_K \neq \emptyset$, which contradicts the assumption that u is unmovable.

It is easy to prove the lemma for the case $k = 1$. We mainly consider $k \ge 2$. In the following proof, we will try to find a reachable point p_n that is nearer to the target within the movable area. Here, instead of directly examining the original movable area, we will apply a transformation and consider a reduced movable area. As shown in Figure 6, we define the ray starting from u and going through τ as the polar axis and define the polar angle on $[-\pi, \pi]$. We apply a transformation as follows. Let r_m denote the minimum radius between the communication range and sensing range, i.e., $r_m = \min\{r_c, r_s\}$. We move the center of each source from (r_c, θ) or (r_s, θ) to (r_m, θ), and we change their radiuses to r_m. Note that the movable area is reduced in such a transformation, but this does not affect the search for a reachable point.

Now, we try to find such kind of reachable point p_n. We denote the position of $s_i \in S_K$ by (r_m, θ_i). Assume that all centers are inside the bottleneck boundary, i.e., $\theta_i \in (-\frac{\pi}{2}, \frac{\pi}{2})$ for $i \in K$. Since u is unmovable and the point it is currently on is intersected by $\odot(s_i, r_m)$, the equation of the circle is $\rho = 2r\cos(\theta_i - \theta)$. Let $\theta_{max} = \max_{i \in K} \theta_i$ and $\theta_{min} = \min_{i \in K} \theta_i$. Given the area intersected by $\odot((r_m, \theta_{max}), r_m)$ and $\odot((r_m, \theta_{min}), r_m)$, any point (ρ, θ) within it should follow

$$\frac{\rho}{2r_m} \le \min\{\cos(\theta_{min} - \theta), \cos(\theta_{max} - \theta)\}. \tag{14}$$

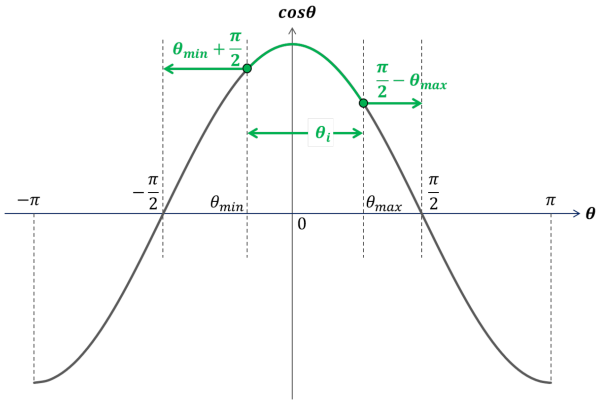


Fig. 7. Cosine curve in $[-\pi, \pi]$.

By $\frac{\rho}{2r_m} \geq 0$ and inequality (14), we have $\cos(\theta_{min} - \theta) > 0$ and $\cos(\theta_{max} - \theta) > 0$. As shown in Figure 7, according to the property of the cosine curve, we have

$$\theta \in (\theta_{max} - \frac{\pi}{2}, \theta_{min} + \frac{\pi}{2}) \tag{15}$$

and

$$\frac{\rho}{2r_m} \leq \cos(\theta_i - \theta), \forall i \in K. \tag{16}$$

The intuitive interpretation of formula (15) is that the point $(\theta_i, \cos(\theta_i))$ (which is initially on the curving segment stroked with green in Figure 7) cannot slide out of range $(-\frac{\pi}{2}, \frac{\pi}{2})$, and inequality (16) demonstrates that the lowest point of a curving segment intercepted from cosine curve within $[-\pi, \pi]$ must be one of its two endpoints. For any circle center $s_i \in S_K$ and any point $p'(\rho', \theta')$ inside the area described by Equation (14), we have the following:

$$\begin{aligned} d^2(s_i, p') - r^2 &= \left(\sqrt{r_m^2 + \rho'^2 - 2\rho' r_m \cos(\theta_i - \theta')}\right)^2 - r_m^2 \\ &= 2r_m\rho' \left(\frac{\rho'}{2r_m} - \cos(\theta_i - \theta')\right) \overset{\text{By (16)}}{\leq} 0, \end{aligned} \tag{17}$$

where the last inequality is due to inequality (16). In addition, since we have $\theta_{min}, \theta_{max} \in (-\frac{\pi}{2}, \frac{\pi}{2})$, we find that the area described by Equation (14) contains a subarea inside the bottleneck boundary, which can be denoted by

$$X_\theta := (\theta_{max} - \frac{\pi}{2}, \theta_{min} + \frac{\pi}{2}) \cap (-\frac{\pi}{2}, \frac{\pi}{2}) \neq \emptyset. \tag{18}$$

Hence, we can find a point $p_0(\rho_0, \theta_0)$ where $\theta_0 \in X_\theta$ and $\rho_0 = 2r_m \cdot \min\{\cos(\theta_{min} - \theta_0), \cos(\theta_{max} - \theta_0)\}$ such that p_0 is inside $\odot(s_i, r_m)$ for $\forall s_i \in S_K$. Moreover, p_0 is closer to τ than u, because $\theta_0 \in X_\theta \subset (-\frac{\pi}{2}, \frac{\pi}{2}), \rho_0 > 0$. That is, p_0 is inside the bottleneck boundary, while u is the pole on the bottleneck boundary.

If $S_L = \emptyset$, then we have already found a point p_0 where the UAV u could move to, and it is closer to τ than the point u is currently on. Thus, it contradicts the assumption that u is unmovable.

If $S_L \neq \emptyset$, then let $\epsilon = r_m - \max_{s' \in S_L} d(s', u)$. Since $S_L \subset \widetilde{N}_u \cup \eta_u$ and $S_L \cap S_K = \emptyset$, we have $d(s', u) < r_m$ for $\forall s' \in S_L$, which implies $\epsilon > 0$. It is obvious that a point inside $\odot(u, \epsilon)$ is also inside $\odot(s', r_m)$ for $\forall s' \in S_L$. Then we can find a point $p_0(\rho_0, \theta_0)$ where $\theta_0 \in X_\theta$ and

$$\rho_0 = \min\{\epsilon, 2r_m \cos(\theta_{min} - \theta_0), 2r_m \cos(\theta_{max} - \theta_0)\}.$$

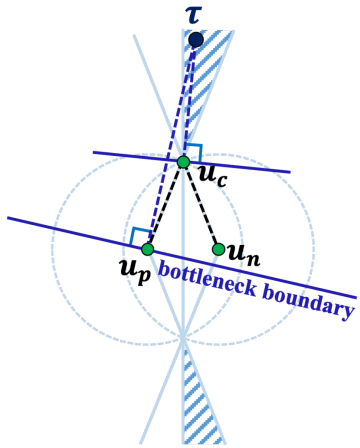


Fig. 8. Example when a UAV u_p is the bottleneck of its bottleneck child u_c. Under this situation, we will find that u_c is always inside its parent u_p's bottleneck boundary, which contradicts with Lemma 6.1.

Therefore, UAV u can move to such a point that is closer to its target τ, which also contradicts the assumption that u is unmovable. This completes the proof. □

Then we could conclude the property of the bottleneck boundary as follows.

Lemma 6.2. *The bottleneck must be on or outside the bottleneck boundary.*

Proof. Assume on the contrary that the bottleneck is inside the boundary. According to the selection rule of bottleneck, the bottleneck has the maximum polar angle among all the source centers. Thus, every source's center must also be inside the bottleneck boundary. This contradicts Lemma 6.1. Therefore, the lemma is proved. □

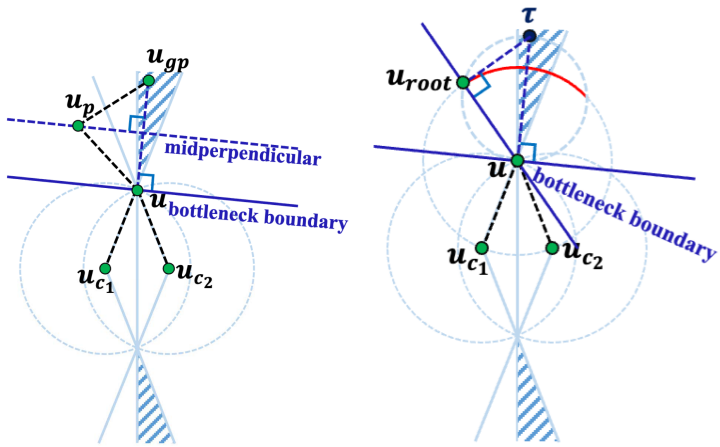
Lemma 6.2 indicates that an unmovable UAV's target and bottleneck stand on the opposite side of the bottleneck boundary. Based on this property, we can show that any two UAVs would not be each other's bottleneck during the distributed motion plan, verifying that TP3 will not cause deadlock.

Theorem 5. *Once the child UAV u_c has been chosen to be its parent u_p's bottleneck, u_p will not be chosen as u_c's bottleneck.*

Proof. Assume that u_p is the bottleneck of u_c. By Lemma 6.2, u_p must be in or outside the bottleneck boundary of u_c, i.e., $\angle \tau u_c u_p \in [\frac{\pi}{2}, \pi]$. Figure 8 shows an example. Applying law of cosines on this angle in $\Delta \tau u_c u_p$, we have

$$\begin{aligned} |\overline{\tau u_p}| &= \sqrt{|\overline{u_c u_p}|^2 + |\overline{\tau u_c}|^2 - 2|\overline{u_c u_p}||\overline{\tau u_c}| \cos \angle \tau u_c u_p} \\ &> |\overline{u_c u_p}| = r_m. \end{aligned} \tag{19}$$

If u_p is a non-root UAV, then the target point given to u_c must be u_p's parent. Since we have proved that $d(u_p, \tau) > r_m$, u_p must be the root UAV, and τ is the real target assigned to u_p. However, u_p cannot cover τ even if it has already been at the nearest point to τ; otherwise, it would not step into the process of selecting bottleneck. Hence, it chooses u_c as its bottleneck and asks it to be closer to τ. With the same deduction used in Equation (19), we could easily get $|\overline{\tau u_p}| > |\overline{\tau u_c}|$, which indicates that u_c must be inside or on the $\odot(\tau, |\overline{\tau u_p}|)$. Moreover, the bottleneck boundary of u_p



(a) When u_{gp} is u_p's parent. (b) When u_{gp} is u_p's target τ.

Fig. 9. Possible positions of non-root UAV u's parent u_p under different situations, when u is unmovable for covering target τ or moving closer to its grandparent u_{gp}. u_{c_1} and u_{c_2} are children of u, where u_{c_1} is chosen to be u's bottleneck. The position of u_p is the target point u wants to be closer to. (a) If u_{gp} is u_p's parent, then u_p must be on the midperpendicular of $\overline{u_{gp}u}$. (b) If u_{gp} is u_p's target τ, then u_p is the root UAV u_{root}, and it must be on the arc stroked in red.

could be viewed as the tangent at point u_p. Hence, τ and u_c lie on the same side of the bottleneck boundary, i.e., u_c is inside the bottleneck boundary of u_p. However, u_c is the bottleneck of u_p, it cannot be inside the bottleneck boundary by Lemma 6.1. This is a contradiction, and the theorem is proved. □

Moreover, for any unmovable UAV u, we can show that u's target and the target of u's bottleneck lie on the same side of u's bottleneck boundary.

Lemma 6.3. *Assume that UAV u is unmovable, and u_c is the bottleneck child it selects, then the target point of u_c must be inside the bottleneck boundary of its parent u.*

Proof. First, consider the case that u is the root UAV. In such a case, the target point of its child u_c is exactly the target τ assigned to u. By Definition 6.1, τ is obviously inside the bottleneck boundary of u.

Next, consider the case that u is a non-root UAV. Under this situation, the target point of u_c is u's parent UAV u_p, whose position is constrained by its child u and u's target point u_{gp}. Since u_{gp} might be u_p's parent (when u_p is a non-root UAV) or target (when u_p is the root UAV), we discuss it by case.

If u_{gp} is u_p's parent UAV, then $d(u_p, u_{gp}) = d(u_p, u) = r_m$, since u_p is the bottleneck of u_{gp} and u is the bottleneck of u_p, as shown in Figure 9(a). It indicates that u_p lies on the midperpendicular of $\overline{u_{gp}u}$. Obviously, the midperpendicular is parallel to the bottleneck boundary of u, and thus u_p is inside the bottleneck boundary of u.

If u_{gp} is u_p's target τ, then u_p is the root UAV u_{root}. Since u is u_{root}'s bottleneck, we know that u is on or outside the bottleneck boundary of u_{root} by Lemma 6.2. Thus as shown in Figure 9(b) u_{root} must lie on the minor arc stroked in red, which is intersected by $\odot(\frac{\tau+u}{2}, \frac{|\overline{\tau u}|}{2})$ and $\odot(u, r_m)$. Moreover, since the bottleneck boundary of u could be viewed as the tangent of $\odot(\frac{\tau+u}{2}, \frac{|\overline{\tau u}|}{2})$ at point u, the arc must be inside u's bottleneck boundary. Thus u_{root} must be inside u's bottleneck boundary.

Hence, the lemma is proved. □

This indicates that u's current bottleneck u_c will move toward the area inside u's bottleneck boundary. Then, according to Lemma 6.2, a bottleneck cannot be inside the bottleneck boundary, and thus u_c would become a non-bottleneck UAV, which would help u get out of the unmovable situation.

Based on the properties above, we can conclude that TP3 converges to cover targets in the motion plan.

Theorem 6. *In the motion plan TP3, each newly appeared target could be successfully covered by its assigned root UAV.*

Proof. We will prove in a bottom-up manner that every UAV could always be movable after several times of child's movements, and, finally, the root UAV will cover the new target.

First, consider a leaf UAV u that has no child. If u is unmovable, then, by Theorem 5, we know that u's parent cannot be its bottleneck. Then u's bottleneck could only be its covered target. In such a case, to respond to the newly appeared target, u would stop covering the bottleneck target. When it stops covering all its bottleneck targets, no bottleneck exists, and u will be movable.

Next, consider a non-leaf UAV u whose children are all leaf UAVs. If u is unmovable, then, similarly, we could get that u's bottleneck u_b can only be its child (a leaf UAV) by Theorem 5. Based on the deduction above, we know that u_b would keep moving closer to its target point. Moreover, by Lemma 6.3 we know u_b's target point is inside u's bottleneck boundary. Once u_b moves across the boundary, it would not be u's bottleneck by Lemma 6.2. Thus u will turn into movable status when all its children have been inside its bottleneck boundary by Lemma 6.1.

Last, the same inference could be applied from the bottom to up to the root UAV, and the root UAV can finally cover the new target. Thus, the theorem is proved. □

As mentioned, apparently the ULMC converges to drive UAVs to the destined positions. Therefore the convergence of coverage for the TPMP method is guaranteed, i.e.,

Theorem 7. *With the TPMP method, the UAV swarm converges to cover all appeared targets.*

Note that our method can be easily extended to 3D space by mapping our current geometry concepts to their 3D forms, e.g., mapping a target circle to a target sphere. Also, our preceding theoretical results still hold for the extended methods by the same mapping methods, e.g., mapping the bottleneck boundary to a bottleneck plane.

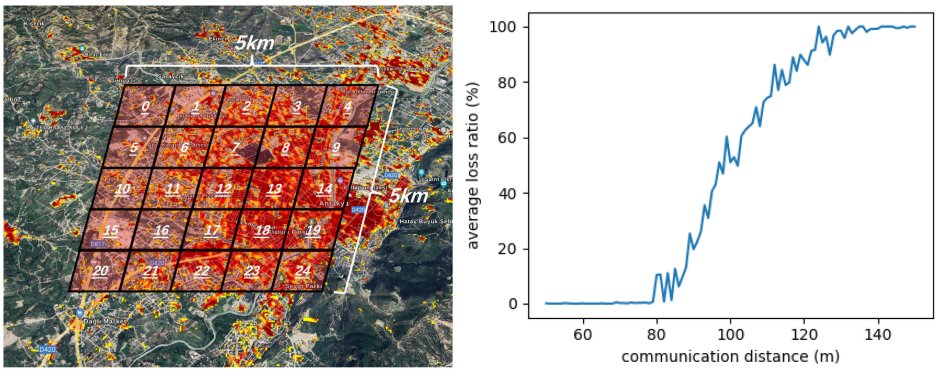
7 EXPERIMENTS

We conduct numerical experiments to evaluate our algorithms in both coverage and communications.

7.1 Environment Setup

We consider the scenes of disaster response, which are the typical scenarios of monitoring with UAV swarms. To provide more clarity, we refer to the analysis results released by Aria et al. (2019), which include the damage degrees of disaster areas for significant events since 2014, based on pre and post-disaster satellite images. For our research, we select four events, two of which are earthquakes, and the other two are fires. We manually choose 25 square areas, each having a length of 1 km. One of the events we pick is the Turkey earthquake of 2023, and the selected areas are displayed in Figure 10(a).

For each selected area, we crop the heatmap image within it, where the center position of each pixel is viewed as a candidate point. Next, we employ the k-means algorithm to partition the



(a) 25 areas selected on the map of damage degrees in Turkey after the earthquake.

(b) Average packet loss ratio at different communication distances.

Fig. 10. Simulation configurations: areas selected for generation of targets and average packet loss ratio at different communication distances for picking the communication radius.

candidate points into 50 clusters. The resulting centers of these clusters serve as our targets in the experiments.

The appearance order of targets, the initial positions and the communication topology of UAVs are randomly generated, but they are fixed for the same area for fair comparison among different online coverage algorithms. Note that the maximum degree of the randomly generated communication topology is set to 3. Also, the sensing radius r_s is set to 50 m. The size of the UAV swarm has been predetermined to be 25, 50, 80, and 100 to encompass both small and large scales of UAV swarms.

7.2 Network Configuration

The communication range r_c is essentially a marginal communication distance within which two UAVs can communicate reliably with a low packet loss ratio, which should be selected judiciously according to the quality of the channel in practice.

Therefore, we first need to introduce the model of the channel we used in our simulation. Specifically, the experiments of communications are conducted in a popular network simulator ns-3 [46]. In this simulator, we use IEEE 802.11b radio standard that operates in the 2.4-GHz frequency band for wireless communication. A log distance path loss propagation model with three distance fields and a Friis propagation model are chained to jointly address the effect of path loss and fading in wireless communications. The constant range propagation model with light speed is adopted to characterize the delay in the propagation between two UAVs. Meanwhile, the DSSS error rate model [40] is adopted to determine the bit error rate during transmission.

To find the proper communication range given the model of channel or experimental environments, the communication range can be determined by testing the packet loss ratio with respect to the communication distance between two UAVs and then locating a marginal distance with a low packet loss ratio. In our experiments, we make two UAVs with fixed positions that keep transmitting data packets of size 64 bytes at the constant rate of 3.5 kbps for 1,000 seconds from one to another. The average packet loss ratio under each communication distance within the range from 50 to 150 m with a step of 1 m are presented in Figure 10(b). Results show that the packet loss ratio is very low as the communication distance is below about 80 m. As a result, we set the communication range r_c of UAVs to 70 m, which ensures the links preserved by the TPM P algorithm to be reliable even with 10m of bias in positioning or drift in UAV movement.

7.3 Coverage Quality

We compare the following algorithms, respectively:

(1) *STBA* [37]: A deployment algorithm that determines the positions of a fixed number of UAVs such that the UAV network is globally connected to a sink node with a fixed location while maximizing the number of covered targets. To adapt this algorithm to our online scenario, the position of the newly appeared target at a round is set to the sink node's position, which ensures the algorithm satisfies that the newly appeared target gets covered.
(2) *TPMP-recfg*: This algorithm basically follows the TPMP algorithm. However, every time the UAV swarm has covered 10 more targets, it calls the STBA algorithm to calculate the UAV deployment for covering all appeared targets. Subsequently, it sends the minimum spanning tree of the communication graph formed by the STBA's UAV deployment to the Topology-Reconfiguration mechanism as the new communication topology. After reconfiguration, it asks the UAVs to move to the positions provided by the STBA algorithm. Then the UAV swarm continues to follow the TPMP algorithm to cover newly appeared targets under the constraint of the new communication topology.
(3) *TPMP-fixed*: Our solution for TPMTC, which assigns the target with KMA and then employs motion plan TP3.
(4) *RAND-fixed*: Uniformly sample a UAV to cover the newly appeared target and then employ motion plan TP3.
(5) *MC-fixed*: Solve the SLK-TPMTC with the Gurobi solver under the initially given communication topology.
(6) *EXP*: Coverage in expectation for static UAV swarm that is globally connected with minimum overlaps among UAVs. Specifically, we simply assume that any two UAVs are non-overlapping if they are not required to be connected following the communication topology, the distance between two UAVs for each edge in the communication topology is the maximum, i.e., r_c, and the probability that a target is covered by the swarm equals to the ratio of the whole sensing area of UAVs to the whole sampling area.
(7) *STAR* [38]: Solve the coverage-optimal deployment under the constraint of a star formation for the UAV swarm.

Note that STBA, TPMP-fixed, TPMP-recfg, and RAND algorithms are online algorithms that can cover each newly appeared target sequentially.

First, the AC of all online algorithms is demonstrated in Figure 11. The results indicate that both of our proposed algorithms, TPMP-fixed and TPMP-recfg, outperform the baseline algorithm RAND-fixed under all UAV swarm sizes. In fact, the AC of TPMP-fixed and TPMP-recfg exceeds that of RAND-fixed by 36.649% and 209.804%, respectively, demonstrating the efficacy of the KMA algorithm in terms of coverage. Notably, the AC of TPMP-recfg algorithm is 92.133% of that of the fully centralized baseline algorithm STBA, which is a relatively close approximation.

Second, we closely monitor the online coverage of targets as they appear online. Specifically, we keep a record of the number of targets covered in every round using all online algorithms. As shown in Figure 12, we noticed a decline in the number of covered targets in online algorithms as the UAV swarm size decreases, such as when $K = 25$. This decline could be attributed to the inherent limitations of the UAV swarm's coverage ability. However, as the UAV swarm size increases beyond 25, the TPMP-recfg algorithm exhibited a comparable coverage ability with the STBA algorithm in most rounds. Impressively, the TPMP-recfg algorithm even outperformed the STBA algorithm in achieving better coverage at a later stage when K is 50 or 80. Therefore, we conclude that our TPMP-recfg algorithm has the potential to thoroughly exploit the coverage ability of a UAV swarm, outperforming even fully centralized algorithms. It is worth noting that

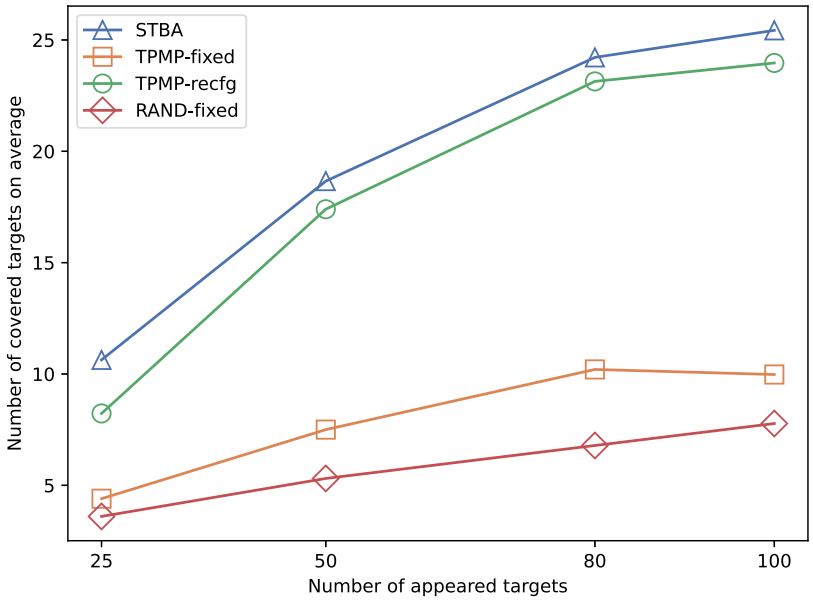


Fig. 11. Average coverage.

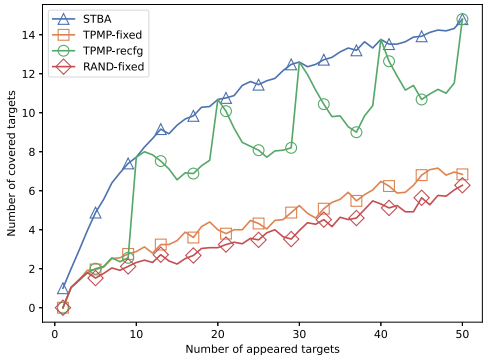
the TPMP-recfg algorithm did not surpass the STBA algorithm in terms of coverage when $K = 100$. This could be attributed to the fact that the STBA algorithm has already achieved the global optimum, covering all appeared targets. Hence, theoretically, there is no better solution that exists.

Third, we compare the average time of covering each newly appeared target, referred to as the coverage time, under each online algorithm. Specifically, we set the average flying speed of UAVs as 20 m/s, and the time for reconfigurations in STBA has not been counted in the coverage time. Results illustrated in Figure 13 show that (1) as swarm size increases, the average coverage time under all algorithms decreases, indicating a better coverage ability; (2) among all online algorithms, our TPMP-fixed and TPMP-recfg algorithm outperforms the other two algorithms STBA and RAND-fixed at every swarm size. (3) The coverage time averaged on all instances under the TPMP-fixed and TPMP-recfg is 5.326 and 6.374 seconds, which is reduced by at least 49.204% and 39.208% compared to two other baselines, respectively.

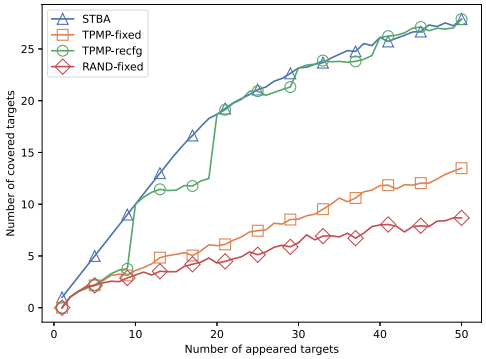
Fourth, we investigate the deployment performance of all algorithms. Specifically, for each online algorithm, we monitor the vector of UAV positions derived by the algorithm at each round and pick the one where most targets (including the unappeared ones for that round) get covered as the final deployment. The number of covered targets after the UAV deployment is referred to as the final coverage. As illustrated in Figure 14, our TPMP-recfg algorithm outperforms all algorithms other baseline algorithms, including the STBA algorithm, whose final coverage exceeds the STBA, EXP, MC-fixed, RAND-fixed, and STAR algorithms by 3.728%, 75.379%, 87.186%, 142.127%, and 145.518%. This again validates the potential of our proposed solution to thoroughly exploit the UAV swarm's flexibility.

7.4 Communication Quality

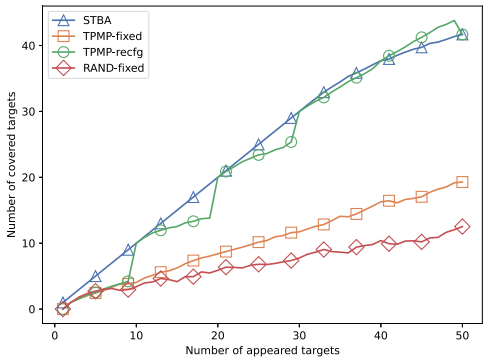
We further validate the improvement in communications based on motion plans generated by the STBA, TPMP-fixed, and TPMP-recfg algorithms and conducted by the ULMC algorithm, where the average speed of moving UAVs is set to be 20 m/s. The simulation time is set to 100 seconds. 50 experiments are conducted respectively for swarm sized 25, 50, 80, and 100. In each



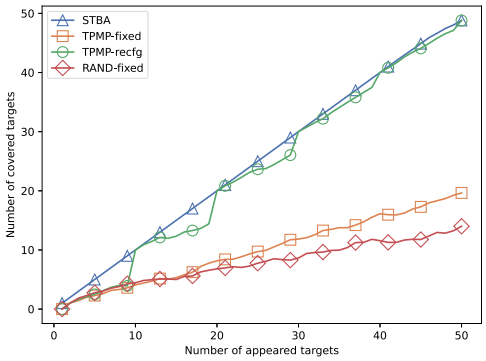
(a) $K = 25$.



(b) $K = 50$.



(c) $K = 80$.



(d) $K = 100$.

Fig. 12. Online coverage.

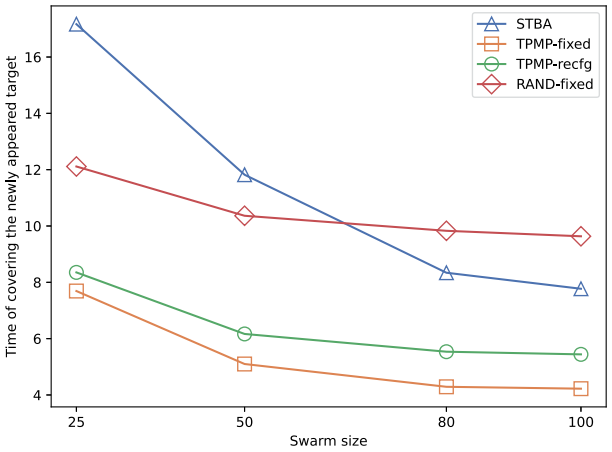


Fig. 13. Avearge time of covering each newly appeared target.

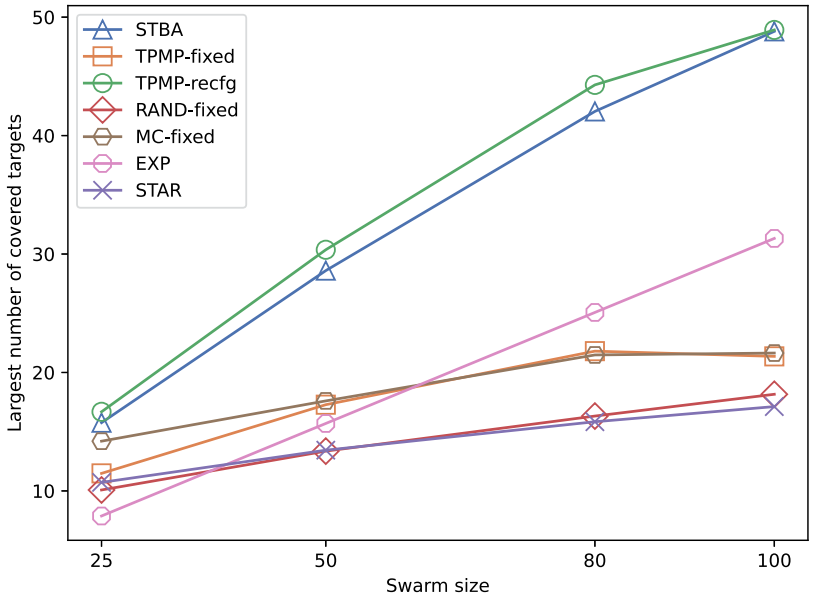
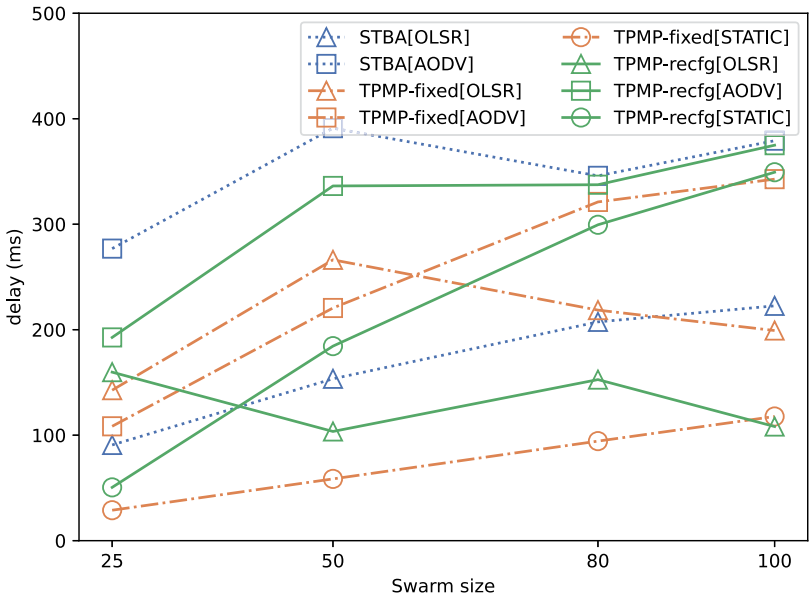


Fig. 14. Final coverage.

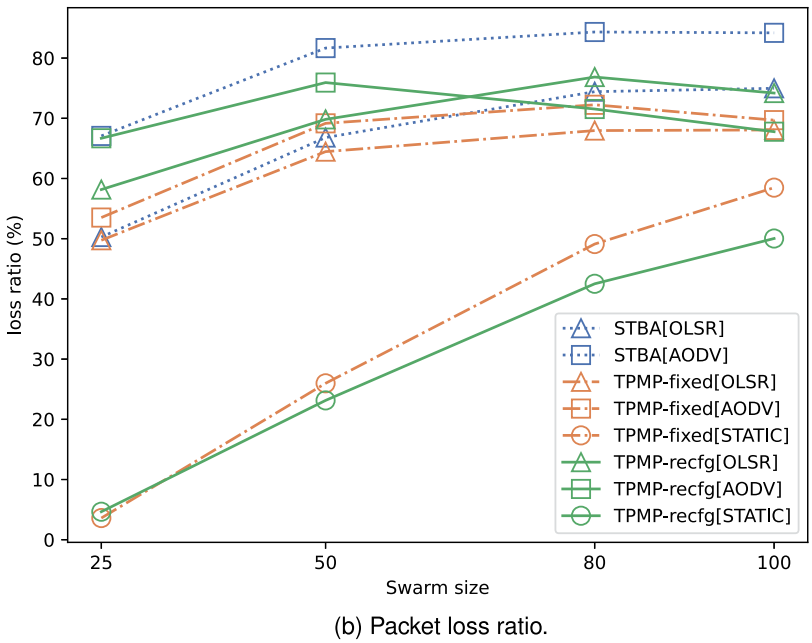
experiment, three instances are generated. Instances in the same experiment share exactly the same UAV motion plans and network settings (including the communication model as introduced in Section 7.2 and the network traffic). The only difference of them is that they respectively use static, OLSR [8], and AODV [41] routing protocols, except that no static routing protocol can be configured when conducting motion plans generated by the STBA algorithm. Specifically, as a fixed communication topology $\widetilde{E}$ is preserved during the movement, the static protocol is enabled by initially configuring each UAV's routing table according to the shortest paths in $\widetilde{E}$. OLSR is a standard table-driven/proactive routing protocol, where each node periodically maintains and updates the routing table. AODV is a classical on-demand/reactive routing protocol, where each node would search for a routing path only when communication needs arise. The Constant Bit Rate traffic is applied in the experiments. More concretely, the packets sized 64 bytes are generated at the rate of 3.5 kbps between 8, 16, 26, and 33 randomly picked UAV pairs when the swarm size is respectively 25, 50, 80, and 100. Moreover, to model the variable/emergency communication needs in real-time monitoring scenarios, the lifetime of each communication UAV pair is set to 5 seconds, i.e., the communication UAV pairs are re-selected per 5 seconds.

The network performance, including the end-to-end delay and the packet loss ratio, is shown in Figure 15. The labels in the figure indicate the algorithm utilized for generating the motion plan, accompanied by the routing protocol employed, enclosed within brackets. The results illustrate the following:

— Our proposed solutions, TPMP-recfg[STATIC] and TPMP-fixed[STATIC], exhibit superior performance compared to other baselines in terms of packet loss ratio at every swarm size. Specifically, on all instances, the packet loss ratio under the TPMP-recfg[STATIC] method and TPMP-fixed[STATIC] method is 30.0825% and 34.2925%, respectively. Compared to other solutions using a dynamic routing protocol, the TPMP-recfg[STATIC] method reduces packet loss ratio by at least 51.91% (TPMP-fixed[OLSR]) and up to 62.071% (STBA[AODV]).
— When the swarm size is relatively small (i.e., K = 25), both TPMP-recfg[STATIC] and TPMP-fixed[STATIC] outperform other baselines in terms of end-to-end delay. Specifically,



(a) End-to-end delay.



(b) Packet loss ratio.

Fig. 15. Network performance for UAV swarm sized 25, 50, 80, and 100 in different routing protocols.

the delay under the TPMP-recfg[STATIC] method and TPMP-fixed[STATIC] method is 50.528 and 28.874 ms, respectively. Compared to other solutions with a dynamic routing protocol, TPMP-fixed[STATIC] reduces delay by at least 68.177% (STBA[OLSR]) and up to 89.571% (STBA[AODV]). On instances from all swarm sizes, the delay under the

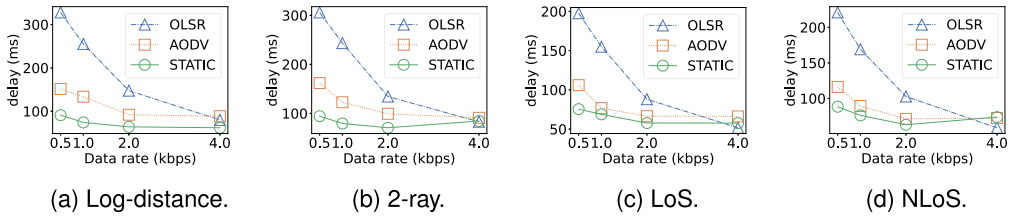


Fig. 16. End-to-end delay (ms) of the TPMP algorithm under different channel models, routing protocols, and data rates when UAV swarm size is 50.

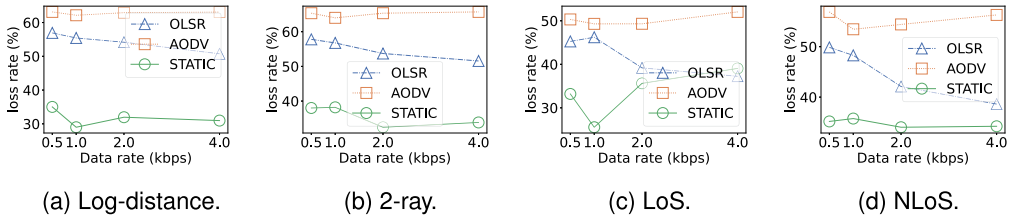


Fig. 17. Packet loss ratio (%) of the TPMP algorithm under different channel models, routing protocols, and data rates when UAV swarm size is 50.

TPMP-fixed[STATIC] method is 74.887 ms, which reduces delay under a dynamic routing protocol by at least 42.856% (TPMP-recfg[OLSR]) and up to 78.495% (STBA[AODV]).

The results obtained confirm that maintaining a communication topology during movement can enhance the end-to-end delay and the packet loss ratio. We also note that as the size of the UAV swarm increases, the TPMP-recfg[STATIC] approach does not exhibit better performance compared to other solutions that use dynamic routing protocols. This is likely due to the fact that the TPMP-recfg method achieves better coverage performance, leading to a larger physical distance between UAVs, which results in a larger number of forwarding times and propagation delay. To investigate this further, we monitored the hop count of each UAV pair in the static routing tables under the two methods. The results indicate that the average hop count under the TPMP-recfg motion plan is always the largest compared to that under the TPMP-fixed motion plan with the same routing protocol. This may explain the longer delay observed in TPMP-recfg[STATIC].

Subsequently, considering that the UAV swarm may work in different scenarios with various transmission conditions, we validated the network performance of our solution under different channel models. Specifically, we employed all three UAV-to-UAV channels as summarized in refyan2019comprehensive, which are log-distance, line-of-sight, and 2-ray models. Moreover, we also tested the performance under the non-line-of-sight channel to cover the cluttered environment with obstacles. Additionally, we investigated the impact of the data rate, which is set to 0.5, 1, 2, and 4 kbps, respectively. The UAV swarm size is fixed to 50. As illustrated in Figures 16 and 17, results demonstrate that (1) our solution outperforms dynamic routing solutions in terms of both delay and loss ratio at nearly all channel conditions and data rates; (2) averaged on all instances, the end-to-end delay of our solution is 73.800 ms, which is reduced by 54.950% and 26.408%, compared to the OLSR-based and AODV-based solutions, respectively; and (3) averaged on all instances, the packet loss ratio of our solution is 33.876%, which is reduced by 30.855% and 41.948%, compared to the OLSR-based and AODV-based solutions, respectively. This validates the robustness of our solution, which preserves a fixed communication topology, under different network conditions.

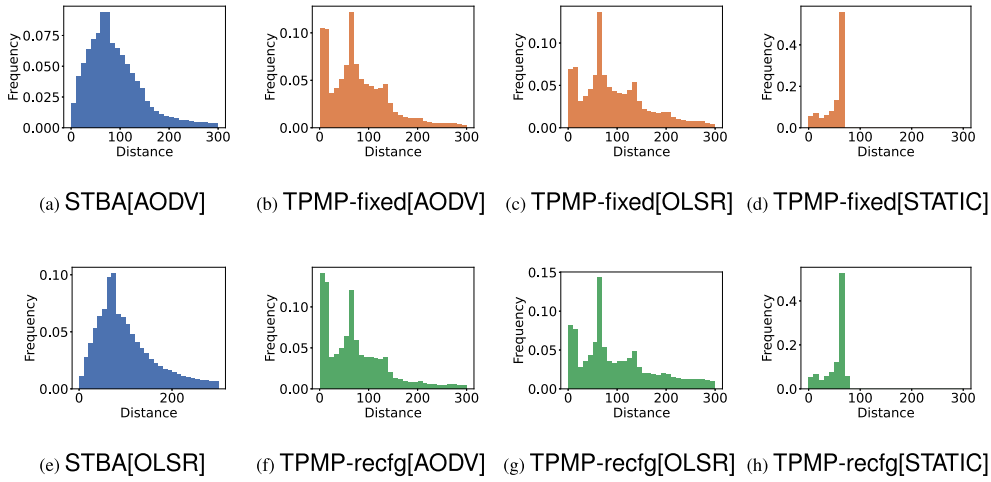


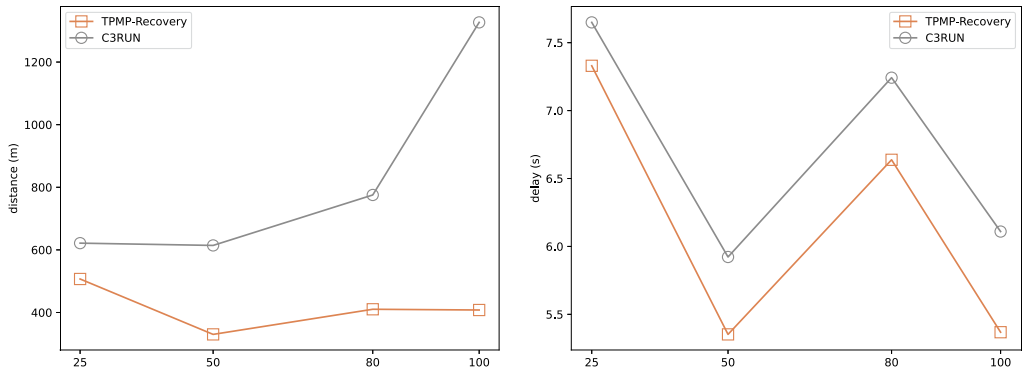
Fig. 18. The occurrence frequency of distance between UAVs in the routing table entries under different methods.

To investigate the reasons behind the improvement, we further analyzed the distance between all UAV pairs in the routing tables, recorded every 2 seconds under different methods. The occurrence frequency of the distances is illustrated in Figure 18. The results demonstrate the following:

— The distance of all UAV pairs used for communication under the TPMP-fixed[STATIC] and TPMP-recfg[STATIC] method is strictly less than 71 m. This is in contrast to the OLSR-based and AODV-based methods, where the maximum distance that packets can be forwarded between corresponding UAVs can be over 300 m. This indicates that if packets are forwarded between corresponding UAVs under these methods, then the loss ratio of the link will be extremely high.[7]
— Based on the testing results in Figure 10(b), the average packet loss ratio is approximately greater than 1%, 20%, and 50% as the distance between two UAVs that packets are forwarded between is greater than 80, 90, and 100 m, respectively. Accordingly, the percentage of neighboring UAVs in routing table entries that are greater than 80, 90, and 100 m under an OLSR-based method are at least 47.920% (TPMP-recfg[OLSR]), 44.001% (TPMP-fixed[OLSR]), and 39.740% (TPMP-fxied[OLSR]). The percentage under an AODV-based method are respectively 35.511% (TPMP-recfg[AODV]), 31.416% (TPMP-recfg[AODV]), and 27.612% (TPMP-recfg[AODV]).
— The average distance of neighboring UAVs in the table entries under the static, OLSR, and AODV policy are respectively 53.265, 80.456, and 98.672 m. This indicates that the average propagation delay between UAVs under the static routing protocol is the smallest in expectation.

Moreover, we monitored the routing table entries and tried to calculate the route between each two UAVs in a centralized manner during the simulation. Results demonstrate that when using dynamic routing protocols, there can be cyclic routes or no routes.[8] Specifically, following the

[7]Although the packets are forwarded with a low loss ratio, the overall packet loss ratio of STATIC-based reaches over 50% when the swarm size is 100. This may be due to network congestion as the traffic of the network increases. The loss ratio under OLSR-based and AODV-based methods is even worse.
[8]No routes indicate that for a given UAV pair, the next-hop address to the destination does not exist in some UAVs on the forwarding path starting from the source UAV.



(a) Cumulative moving distance. (b) Recovery delay after detecting the failure.

Fig. 19. Performance of topology recovery.

OLSR protocol, the ratio of cyclic routes is at least 3.593% (TPMP-fixed[OLSR]), and the ratio of no routes is at least 26.082% (TPMP-fixed[OLSR]). Following the AODV protocol, the ratio of cyclic routes is at least 1.658% (TPMP-recfg[AODV]), and the ratio of no routes is at least 0.044%. Note that since the AODV algorithm works in an on-demand manner, we only check UAV pairs that exist in at least one of the UAV routing tables. However, under the static routing protocols, routes for all UAV pairs exist and are correct. This further validates the efficiency of our solution.

7.5 Topology-Recovery

To assess the effectiveness of our proposed TPMP-Recovery mechanism, which is described in Section 5.1, we conduct a comparison with the C3RUN algorithm [54]. The C3RUN algorithm shares some similarities with our mechanism, in that it utilizes UAVs to detect failures by broadcasting HELLO/heartbeat messages. However, there are some important differences between the two approaches: First, in the C3RUN algorithm, all UAVs target the last reported position of the failed UAV, whereas in our mechanism, only the neighbors of the failed UAV are targeted. Second, in the C3RUN algorithm, the UAVs move sequentially toward their target positions to ensure global connectivity, whereas in our mechanism, the UAVs use the ULMC algorithm to simultaneously move to their destinations with connectivity guaranteed by Theorem 4. To ensure a fair comparison, we enhance the C3RUN algorithm by allowing the UAVs to move simultaneously toward their final positions, as calculated by the original algorithm.

We evaluate both mechanisms in terms of two metrics: the cumulative moving distance and the recovery delay after detecting a failure, which we will refer to as recovery delay for brevity.[9] As shown in Figure 19, our TPMP-Recovery mechanism outperforms the C3RUN algorithm in both metrics across all swarm sizes. Specifically, our mechanism reduces the cumulative moving distance and recovery delay by 50.409% and 8.301%, respectively, as compared to the C3RUN algorithm. Meanwhile, the cumulative moving distance for the C3RUN algorithm exhibits a rapidly increasing trend with increasing swarm size, whereas our mechanism does not, indicating better scalability. Finally, as swarm size increases, the recovery delay for the C3RUN algorithm can drop as the cumulative moving distance increases, while our mechanism achieves consistently lower values for both metrics. This suggests that our mechanism requires fewer UAVs to recover the topology as compared to the C3RUN algorithm.

[9]Note that recovery delay includes the delay for successfully detecting the failure. However, since both algorithms employ the same detection method, we have neglected this part of the delay.

7.6 Experiment Summary

The extensive experimental results demonstrate that (1) enabling static routing by preserving a communication topology can significantly improve the performance of the UAV network, including end-to-end delay and packet loss ratio, under different network conditions as the physical distances of the communication links are strictly bounded; (2) with the developed topology reconfiguration protocol, the UAV swarm can thoroughly exploit its coverage ability when its movement is constrained by a communication topology, the coverage performance of which for the online and offline (deployment) scenarios can even outperform the fully centralized baseline algorithm; and (3) the proposed topology recovery mechanism can rapidly repair the topology with less recovery time and energy consumption.

8 CONCLUSION

In this article, we first propose a distributed motion planning algorithm TPMP for the UAV swarm applications to maintain a virtually fixed communication topology to address the tradeoff between coverage and communication. Then we offer a topology management mechanism to (1) deal with the dynamic UAV leave and arrival that might break the current communication topology and (2) allow the UAVs to reconfigure the preserved communication topology adaptively. Subsequently, we prove the feasibility and convergence of the TPMP algorithm in theory and evaluate the proposed solutions based on datasets of real postdisaster scenarios. Results demonstrate that the TPMP algorithm and the topology reconfiguration protocol thoroughly exploit the flexibility of UAV swarms, the coverage performance of which can even outperform the fully centralized baseline algorithm, while achieving a much lower packet loss ratio than any other solutions using dynamic routing. Moreover, when the communication topology is fixed at all rounds, our solution also achieves the lowest end-to-end delay and outperforms solutions with dynamic routing protocols robustly under different network conditions. In addition, our topology recovery mechanism is validated to be rapid and energy efficient.

This article acknowledges the potential of a preserved communication topology to improve the coverage ability of UAV swarms while achieving superior network performance. Although the design of the communication topology is not extensively covered, future research should optimize the topology to balance coverage and network metrics and adjust the reconfiguration frequency. To enhance the scalability of the swarm, we aim to extend existing solutions to become more flexible and adaptive, such as incorporating hierarchy structures that utilize both dynamic and static routing protocols. These improvements will enable us to achieve better scalability while maintaining a robust and reliable communication network for the UAV swarm.

REFERENCES

[1] Muhammad Morshed Alam and Sangman Moh. 2022. Joint topology control and routing in a UAV swarm for crowd surveillance. *J. Netw. Comput. Appl.* 204 (2022), 103427. https://www.sciencedirect.com/science/article/pii/S1084804522000844

[2] Ebtehal Turki Alotaibi, Shahad Saleh Alqefari, and Anis Koubaa. 2019. LSAR: Multi-UAV collaboration for search and rescue missions. *IEEE Access* 7 (2019), 55817–55832.

[3] Baruch Awerbuch. 1987. Optimal distributed algorithms for minimum weight spanning tree, counting, leader election, and related problems. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing*. 230–240.

[4] Tauã M. Cabreira, Lisane B. Brisolara, and Paulo R. Ferreira Jr. 2019. Survey on coverage path planning with unmanned aerial vehicles. *Drones* 3, 1 (2019), 4.

[5] Christelle Caillouet, Frédéric Giroire, and Tahiry Razafindralambo. 2018. Optimization of mobile sensor coverage with UAVs. In *Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS'18)*. IEEE, 622–627.

[6] Tristan Charrier, Arthur Queffelec, Ocan Sankur, and François Schwarzentruber. 2019. Reachability and coverage planning for connected agents. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 1874–1876.

[7] Ziyan Chen, Nan Cheng, Zhisheng Yin, Jingchao He, and Ning Lu. 2022. Service-oriented topology reconfiguration of UAV networks with deep reinforcement learning. In *Proceedings of the 14th International Conference on Wireless Communications and Signal Processing (WCSP'22)*. IEEE, 753–758.

[8] Thomas Clausen, Philippe Jacquet, Cédric Adjih, Anis Laouiti, Pascale Minet, Paul Muhlethaler, Amir Qayyum, and Laurent Viennot. 2003. Optimized link state routing protocol (OLSR), Technical Report. IETF. 75 pages.

[9] Chen Dai, Kun Zhu, and Ekram Hossain. 2022. Multi-agent deep reinforcement learning for joint decoupled user association and trajectory design in full-duplex multi-UAV networks. *IEEE Trans. Mobile Comput.* (2022).

[10] Tulio Dapper e Silva, Carlos F. Emygdio de Melo, Pedro Cumino, Denis Rosario, Eduardo Cerqueira, and Edison Pignaton De Freitas. 2019. STFANET: SDN-based topology management for flying ad hoc network. *IEEE Access* 7 (2019), 173499–173514.

[11] Milan Erdelj, Tahiry Razafindralambo, and David Simplot-Ryl. 2012. Covering points of interest with mobile sensors. *IEEE Trans. Parallel Distrib. Syst.* 24, 1 (2012), 32–43.

[12] Milan Erdelj, Osamah Saif, Enrico Natalizio, and Isabelle Fantoni. 2019. UAVs that fly forever: Uninterrupted structural inspection through automatic UAV replacement. *Ad Hoc Netw.* 94 (2019), 101612.

[13] Jose Falcon, Mehrube Mehrubeoglu, and Pablo Rangel. 2022. Real-time image processing in a multi-UAV system for structural surveillance through IoT platform. In *Real-Time Image Processing and Deep Learning 2022*, Vol. 12102. SPIE, 139–151.

[14] Malintha Fernando, Ransalu Senanayake, and Martin Swany. 2022. CoCo games: Graphical game-theoretic swarm control for communication-aware coverage. *IEEE Robot. Autom. Lett.* 7, 3 (2022), 5966–5973.

[15] Andy Ham, Delchyne Similien, Stanley Baek, and George York. 2022. Unmanned aerial vehicles (UAVs): Persistent surveillance for a military scenario. In *Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS'22)*. IEEE, 1411–1417.

[16] Erez Hartuv, Noa Agmon, and Sarit Kraus. 2018. Scheduling spare drones for persistent task performance under energy constraints. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. 532–540.

[17] Ji Hong, Zhi Gao, Tiancan Mei, Yifan Li, and Chunyang Zhao. 2019. UAV-based traffic flow estimation and analysis. In *Proceedings of the IEEE 15th International Conference on Control and Automation (ICCA'19)*. IEEE, 812–817.

[18] Abhishek Joshi, Sarang Dhongdi, Mihir Dharmadhikari, Ojit Mehta, and K. R. Anupama. 2022. Enclosing and monitoring of disaster area boundary using multi-UAV network. *J. Ambient Intell. Human. Comput.* (2022), 1–19.

[19] Yiannis Kantaros and Michael M. Zavlanos. 2016. Distributed communication-aware coverage control by mobile sensor networks. *Automatica* 63 (2016), 209–220.

[20] Adil Khan, Jinling Zhang, Shabeer Ahmad, Saifullah Memon, Haroon Akhtar Qureshi, and Muhammad Ishfaq. 2022. Dynamic positioning and energy-efficient path planning for disaster scenarios in 5G-Assisted multi-UAV environments. *Electronics* 11, 14 (2022), 2197.

[21] Karthik Soma, Koresh Khateri, Mahdi Pourgholi, Mohsen Montazeri, Lorenzo Sabattini, and Giovanni Beltrame. 2023. A complete set of connectivity-aware local topology manipulation operations for robot swarms. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 5522–5529. DOI: https://doi.org/10.1109/ICRA48891.2023.10160312

[22] Do-Yup Kim and Jang-Won Lee. 2019. Joint mission assignment and topology management in the mission-critical FANET. *IEEE IoT J.* 7, 3 (2019), 2368–2385.

[23] Junguk L. Kim and Geneva G. Belford. 1996. A distributed election protocol for unreliable networks. *J. Parallel Distrib. Comput.* 35, 1 (1996), 35–42.

[24] Demeke Shumeye Lakew, Umar Sa'ad, Nhu-Ngoc Dao, Woongsoo Na, and Sungrae Cho. 2020. Routing in flying ad hoc networks: A comprehensive survey. *IEEE Commun. Surv. Tutor.* 22, 2 (2020), 1071–1120.

[25] Zhuofan Liao, Jianxin Wang, Shigeng Zhang, Jiannong Cao, and Geyong Min. 2014. Minimizing movement for target coverage and network connectivity in mobile sensor networks. *IEEE Trans. Parallel Distrib. Syst.* 26, 7 (2014), 1971–1983.

[26] Daniel Bonilla Licea, Moises Bonilla, Mounir Ghogho, Samson Lasaulce, and Vineeth S. Varma. 2019. Communication-aware energy efficient trajectory planning with limited channel knowledge. *IEEE Trans. Robot.* 36, 2 (2019), 431–442.

[27] Chao Liu and Zhongshan Zhang. 2022. Towards a robust FANET: Distributed node importance estimation-based connectivity maintenance for UAV swarms. *Ad Hoc Netw.* 125 (2022), 102734.

[28] Chao Liu, Zhongshan Zhang, and Qian Zeng. 2021. Distributed connectivity maintenance for Flying Ad-hoc Networks considering bridging links. *Phys. Commun.* 48 (2021), 101409.

[29] Zhiyu Liu, Bo Wu, Jin Dai, and Hai Lin. 2020. Distributed communication-aware motion planning for networked mobile robots under formal specifications. *IEEE Trans. Contr. Netw. Syst.* 7, 4 (2020), 1801–1811.

[30] Ruyu Luo, Hui Tian, and Wanli Ni. 2021. Communication-aware path design for indoor robots exploiting federated deep reinforcement learning. In *Proceedings of the IEEE 32nd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'21)*. IEEE, 1197–1202.

[31] Aastha Maheshwari and Narottam Chand. 2019. A survey on wireless sensor networks coverage problems. In *Proceedings of 2nd International Conference on Communication, Computing and Networking*. Springer, 153–164.
[32] Ajith Anil Meera, Marija Popović, Alexander Millane, and Roland Siegwart. 2019. Obstacle-aware adaptive informative path planning for uav-based target search. In *Proceedings of the International Conference on Robotics and Automation (ICRA'19)*. IEEE, 718–724.
[33] Nimrod Megiddo. 1990. On the complexity of some geometric problems in unbounded dimension. *J. Symbol. Comput.* 10, 3-4 (1990), 327–334.
[34] Syed Agha Hassnain Mohsan, Muhammad Asghar Khan, Fazal Noor, Insaf Ullah, and Mohammed H. Alsharif. 2022. Towards the unmanned aerial vehicles (UAVs): A comprehensive review. *Drones* 6, 6 (2022), 147.
[35] Arjun Muralidharan and Yasamin Mostofi. 2021. Communication-aware robotics: Exploiting motion for communication. *Annu. Rev. Contr. Robot. Auton. Syst.* 4, 1 (2021).
[36] Sajjad Nematzadeh, Mahsa Torkamanian-Afshar, Amir Seyyedabbasi, and Farzad Kiani. 2022. Maximizing coverage and maintaining connectivity in WSN and decentralized IoT: An efficient metaheuristic-based method for environment-aware node deployment. *Neural Comput. Appl.* (2022), 1–31.
[37] Tu N. Nguyen, Bing-Hong Liu, and Shih-Yuan Wang. 2019. On new approaches of maximum weighted target coverage and sensor connectivity: Hardness and approximation. *IEEE Trans. Netw. Sci. Eng.* 7, 3 (2019), 1736–1751.
[38] Jacopo Panerati, Luca Gianoli, Carlo Pinciroli, Abdo Shabah, Gabriela Nicolescu, and Giovanni Beltrame. 2018. From swarms to stars: Task coverage in robot swarms with connectivity constraints. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'18)*. IEEE, 7674–7681.
[39] Faezeh Pasandideh, Tulio Dapper e Silva, Antonio Arlis Santos da Silva, and Edison Pignaton de Freitas. 2022. Topology management for flying ad hoc networks based on particle swarm optimization and software-defined networking. *Wireless Netw.* (2022), 1–16.
[40] Guangyu Pei and Tom Henderson. 2009. Validation of ns-3 802.11 b PHY model. Retrieved from http://www.nsnam.org/~pei/80211b.pdf
[41] Charles Perkins, Elizabeth Belding-Royer, and Samir Das. 2003. RFC3561: Ad hoc on-demand Distance Vector (AODV) Routing, Technical Report. IETF. 35 pages.
[42] Chiara Piacentini, Sara Bernardini, and J Christopher Beck. 2019. Autonomous target search with multiple coordinated UAVs. *J. Artif. Intell. Res.* 65 (2019), 519–568.
[43] Arthur Queffelec. 2021. *Connected Multi-agent Path Finding: How Robots Get Wway with Texting and Driving*. Ph. D. Dissertation. Université Rennes 1.
[44] Shirin Rahmanpour and Reza Mahboobi Esfanjani. 2018. Planning of communication and motion strategies for networked mobile robots. In *Proceedings of the 9th Conference on Artificial Intelligence and Robotics and 2nd Asia-Pacific International Symposium*. IEEE, 46–53.
[45] Shirin Rahmanpour and Reza Mahboobi Esfanjani. 2019. Energy-aware planning of motion and communication strategies for networked mobile robots. *Inf. Sci.* 497 (2019), 149–164.
[46] George F. Riley and Thomas R. Henderson. 2010. The ns-3 network simulator. In *Modeling and Tools for Network Simulation*. Springer, 15–34.
[47] Arnau Rovira-Sugranes, Abolfazl Razi, Fatemeh Afghah, and Jacob Chakareski. 2022. A review of AI-enabled routing protocols for UAV networks: Trends, challenges, and future outlook. *Ad Hoc Netw.* 130 (2022), 102790.
[48] Sunitha Safavat and Danda B. Rawat. 2022. OptiML: An enhanced ML approach towards design of SDN based UAV networks. In *Proceedings of the IEEE International Conference on Communications*. IEEE, 1–6.
[49] Shreya Santra, Leonard B. Paet, Mickael Laine, Kazuya Yoshida, and Emanuel Staudinger. 2021. Experimental validation of deterministic radio propagation model developed for communication-aware path planning. In *Proceedings of the IEEE 17th International Conference on Automation Science and Engineering (CASE'21)*. IEEE, 1241–1246.
[50] Zhexiong Shang and Zhigang Shen. 2022. Flight planning for survey-grade 3D reconstruction of truss bridges. *Remote Sens.* 14, 13 (2022), 3200.
[51] Akshay Shetty, Timmy Hussain, and Grace Gao. 2021. Decentralized connectivity maintenance for multi-robot systems under motion and sensing uncertainties. In *Proceedings of the 34th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+'21)*. 2445–2458.
[52] Luca Siligardi, Jacopo Panerati, Marcel Kaufmann, Marco Minelli, Cinara Ghedini, Giovanni Beltrame, and Lorenzo Sabattini. 2019. Robust area coverage with connectivity maintenance. In *Proceedings of the International Conference on Robotics and Automation (ICRA'19)*. IEEE, 2202–2208.
[53] S. Sudhakar, V. Vijayakumar, C. Sathiya Kumar, V. Priya, Logesh Ravi, and V. Subramaniyaswamy. 2020. Unmanned aerial vehicle (UAV) based forest fire detection and monitoring for reducing false alarms in forest-fires. *Comput. Commun.* 149 (2020), 1–16.
[54] Wen Tian, Zhenzhen Jiao, Min Liu, Meng Zhang, and Dong Li. 2019. Cooperative communication based connectivity recovery for UAV networks. In *Proceedings of the ACM Turing Celebration Conference-China*. 1–5.

[55] Jiaxing Wang, Lin Bai, Jianrui Chen, and Jingjing Wang. 2023. Starling flocks-inspired resource allocation for ISAC-aided green Ad Hoc networks. *IEEE Trans. Green Commun. Netw.* 7, 1 (2023), 444–454.
[56] Liao Wenxing, Wu Muqing, Zhao Min, Li Peizhe, and Li Tianze. 2017. Hop count limitation analysis in wireless multi-hop networks. *Int. J. Distrib. Sens. Netw.* 13, 1 (2017), 1550147716683606.
[57] Shiguang Wu, Zhiqiang Pu, Tenghai Qiu, Jianqiang Yi, and Tianle Zhang. 2022. Deep reinforcement learning based multi-target coverage with connectivity guaranteed. *IEEE Trans. Industr. Inf.* (2022).
[58] Zhaoyue Xia, Jun Du, Jingjing Wang, Chunxiao Jiang, Yong Ren, Gang Li, and Zhu Han. 2021. Multi-agent reinforcement learning aided intelligent UAV swarm for target tracking. *IEEE Trans. Vehic. Technol.* 71, 1 (2021), 931–945.
[59] Hao Xu, Peize Liu, Xinyi Chen, and Shaojie Shen. 2022. D^2SLAM: Decentralized and distributed collaborative visual-inertial SLAM system for aerial swarm. arXiv:2211.01538. Retrieved from https://arxiv.org/abs/2211.01538
[60] Xianghua Xu, Zhixiang Dai, Anxing Shan, and Tao Gu. 2019. Connected target ϵ-probability coverage in WSNs with directional probabilistic sensors. *IEEE Syst. J.* (2019).
[61] Xiaoling Xu, Lixin Yang, Wei Meng, Qianqian Cai, and Minyue Fu. 2019. Multi-agent coverage search in unknown environments with obstacles: A survey. In *Proceedings of the Chinese Control Conference (CCC'19)*. IEEE, 2317–2322.
[62] Evşen Yanmaz. 2022. Positioning aerial relays to maintain connectivity during drone team missions. *Ad Hoc Netw.* 128 (2022), 102800.
[63] Jiguo Yu, Ying Chen, Liran Ma, Baogui Huang, and Xiuzhen Cheng. 2016. On connected target k-coverage in heterogeneous wireless sensor networks. *Sensors* 16, 1 (2016), 104.
[64] Won Joon Yun, Soohyun Park, Joongheon Kim, MyungJae Shin, Soyi Jung, David A. Mohaisen, and Jae-Hyun Kim. 2022. Cooperative multiagent deep reinforcement learning for reliable surveillance via autonomous multi-UAV control. *IEEE Trans. Industr. Inf.* 18, 10 (2022), 7086–7096.
[65] Fan Zhang and Gangqiang Yang. 2020. A stable backup routing protocol for wireless Ad Hoc networks. *Sensors* 20, 23 (2020), 6743.
[66] Ruilong Zhang, Qun Zong, Xiuyun Zhang, Liqian Dou, and Bailing Tian. 2022. Game of drones: Multi-UAV pursuit-evasion game with online motion planning by deep reinforcement learning. *IEEE Trans. Neural Netw. Learn. Syst.* (2022).
[67] Xin Ming Zhang, Yue Zhang, Fan Yan, and Athanasios V. Vasilakos. 2014. Interference-based topology control algorithm for delay-constrained mobile ad hoc networks. *IEEE Trans. Mobile Comput.* 14, 4 (2014), 742–754.
[68] Zixuan Zhang, Bo Zhang, and Yunlong Wu. 2022. Joint communication–motion planning in networked robotic systems. *Appl. Sci.* 12, 12 (2022), 6261.

Received 17 December 2022; revised 9 July 2023; accepted 14 October 2023