SNPSG Journal of Intelligent Unmanned Systems

2026, Vol. 1, No. 1, Article 123
DOI: https://doi.org/10.65218/jius.2026.123

RESEARCH ARTICLE

Simulating and Analyzing Coordination Approaches among Unmanned Air and Unmanned Ground Vehicles for Search And Rescue Scenarios

Tatiana Reimer[1,*], Bruno Olivieri[1,*], Markus Endler[1,*]

1 Department of Informatics, Pontifical Catholic University of Rio de Janeiro (PUC-Rio), Rio de Janeiro, Brazil

* Corresponding author(s): Tatiana Reimer, Bruno Olivieri and Markus Endler
Email(s): tbarata@inf.puc-rio.br, bolivieri@inf.puc-rio.br, endler@inf.puc-rio.br

Received: 26 March 2026 | Accepted: 19 April 2026 | Published Online: 4 July 2026

Abstract: When dealing with large-scale natural disasters such as floods, landslides, hurricanes, or heavy snowfalls, there are often many victims who are trapped in hard-to-reach places, making the time to locate and rescue them critical. In this context, deploying unmanned aerial vehicles (UAVs) alongside a swarm of unmanned ground vehicles (UGVs) has the potential to speed up the Search and Rescue (SAR) missions, as the collaboration between these agents can combine advantages from both of them. The idea is that any simple UAV equipped GPS and communication capabilities can locate besieged and isolated individuals, referred to as Points of interest (POIs), and when it comes across (flies over) a GPS-limited UGV, it guides the UGVs toward the POIs. In this work, we explore different approaches to Air-to-Ground (A2G) coordination in a number of distinct scenarios among unmanned vehicles, and through simulation compare their efficiency based of parameters and metrics.

Keywords: Mobility, Coordination Protocol, UAV, Swarms, Unmanned Ground Vehicles, Air-to-Ground communication

I. Introduction

In disaster scenarios, time is a critical and limited resource. The urgency to locate and rescue survivors is increased by the rapidly changing conditions of the affected area, and the speed of response by rescue teams can be greatly affected in a matter of seconds, making it critical to look for effective ways to carry out these missions. Therefore, Search And Rescue (SAR) missions are inherently complex and require seamless coordination between teams.

SAR operations have traditionally relied on human teams equipped with limited situational awareness, operating under severe time pressure and often in environments that are dangerous, unstable, or inaccessible. These conditions increase the risk to rescuers and can also affect the first hours of the mission, which are crucial [1].

Two factors play an important role in the overall outcome of SAR missions:

Urgency: In most SAR missions, missing persons are isolated and trapped in a life-threatening situation, where the victim has little time for survival. So, it becomes important for the rescue entourage to locate and assist isolated or trapped people at the earliest possible time.

Isolated or inaccessible regions: many SAR missions occur in isolated or inaccessible regions, e.g., flooded areas, points of landslides, snowy mountains, forests, etc., or where the transportation and communication infrastructure is broken, making it much more difficult, time- and energy-consuming to run the entire rescue operation.

Real world incidents highlight the urgency of improving the efficiency and autonomy of SAR operations. In many disaster scenarios, such as floods, landslides, or volcanic areas, human rescuers face serious environmental and logistical challenges that can delay the location and assistance of victims.

For example, in July 2025, a Brazilian tourist fell near a crater of the active volcano Mount Rinjani in Indonesia during a guided hike, in an area with extremely steep terrain that made immediate human intervention difficult. The search and rescue operation involved local emergency teams supported by helicopters, that conducted an aerial search to locate the victim. The actual rescue team consisted of volunteer climbers, that faced extreme conditions, such as fog, unstable terrain, and shifting sands. A small drone from one of the hikers in the group was only used to try to locate the victim, with no actual participation in the rescue. Due to the difficulties and the exclusive use of human teams, locating the victim took several days, which led to fatality.

In flood disasters, such as the flood in Rio Grande do Sul, Brazil, in May 2024, the environment can completely change, as roads can be submerged and landslides can block access routes and communication infrastructure can be damaged, making it more difficult to locate survivors. During this event, rescue operations were carried primarily by human teams, supported by helicopters and boats with volunteer groups to reach isolated areas. Drones were also deployed to identify victims trapped on rooftops and to map flooded neighborhoods, providing visual data. However, all these operations relied on manual coordination and communication between rescuers on helicopters, boat and drone operators.

Another example occurred in Colombia in May 2023, when a small plane crashed in the Amazon rainforest and four children survived the accident. The dense vegetation, the absence of roads and poor visibility from above turned the search into a complex mission that lasted 40 days. The rescue effort mobilized soldiers, indigenous trackers familiar with the forest, several helicopters that carried aerial searches and planes flying over the jungle fired flares to help search crews on the group at night. Although the children survived, their survival was attributed to their knowledge of the forest, not to the efficiency of the SAR mission.

These situations show how environmental conditions can

 DOI: https://doi.org/10.65218/jius.2026.123

transform a rescue mission into a time-consuming and resource-intensive task. They reinforce the need for autonomous and cooperative agents capable of operating without relying on human presence.

In such contexts, unmanned vehicles can be deployed to collect information, locate victims, and explore hazardous zones. UAVs are particularly effective in providing rapid aerial coverage and real time visual data, while UGVs can perform close range support to the victims and deliver essential supplies.

With the popularization of Unmanned Aerial Vehicles (UAVs) in areas such as surveillance and agriculture, due to their effective performance in monitoring large areas and operating challenging environments, they have also been used in SAR missions, significantly enhancing the efficiency and safety of rescue operations [2].

Other Unmanned Vehicles have also been used in SAR missions [3]. In particular, Unmanned Ground Vehicles (UGVs) are useful for SAR missions that navigate complex terrain, as they can maneuver through rough and hazardous environments and rescue victims.

Hence, SAR operations using both UAVs and UGVs offer multiple advantages that make them an invaluable asset in emergency situations [4]. Some prominent benefits are the following:

Speed of Response: Drones can reach a location much faster than ground vehicles, including UGV, thanks to their ability to fly over traffic and other obstructions. They are also capable of reaching inaccessible places, such as remote or hazardous areas.

Situational Awareness: By providing rapid, cheap access to aerial images/data of a large area, drones allow early responders to map the entire search zone and pinpoint possible places where persons might be trapped and waiting for help. UAVs can further provide real-time visual information and data, reducing the time that ground vehicles move to the locations of trapped people.

Detection and identification: UAVs can carry different sensors, including thermal cameras, which are widely used in search and rescue missions. These sensors can identify ground objects and humans by detecting their heat signatures, making them easy to spot, especially in the dark or dense areas.

Overview Images: Drones can also carry high-resolution cameras capable of capturing still images that provide valuable information to people on the ground, including to UGV that do the actual visitations to the searched and trapped people.

Rescue Announcement: Drones equipped with a loudspeaker or a siren can broadcast a message repeatedly, making the missing people aware that a search is underway. This feature can also facilitate direct communication with victims or ground crews.

Illumination: UAVs can carry spotlights to light up target areas, providing additional visibility during night operations.

Coordination of Ground Team: And last but not least, UAVs can spot the current location of each agent of the ground team, be it a human or a UGV and pass it directions of the most urgent or immediate location that should be visited next for the early response.

In addition to locating individuals, UAV-UGV coordination can also help in the collection of information that can assist SAR teams at the base station by providing statistical data such as how much time it takes to locate individuals and how much time each individual assistance takes. This data fusion can improve the efficiency of the mission.

There are numerous challenges to be explored in UAV-UGV coordination. The number of agents involved, the selection of coordinating agents, the available context information, and the mission objectives all influence the complexity of the problem at hand. In addition to that, there are limitations in communication ranges, heterogeneous sensing and mobility capabilities, and the need to exchange information without relying on a centralized infrastructure.

The most commonly cited type of work in this field is one in which UAVs operate under battery limitations and rely on UGVs that function as mobile charging stations. In these approaches, the problem is typically formulated as a Traveling Salesman Problem, where UAVs must plan routes that ensure they can periodically rendezvous with UGVs to recharge. Our work shares similarities with this class of problems, particularly with studies such as [5], in that agent trajectories are designed to enable interactions between heterogeneous agents in the environment.

However, the problem addressed in this dissertation can be interpreted as the dual of the Traveling Salesman Problem based problem, as in our case UAVs do not dynamically adjust their paths to explore the environment. Consequently, it is the UGVs that must adapt their behavior, navigating the environment to find the Points of Interest detected by the UAVs.

Using UAVs as coordinators offers significant advantages due to their aerial perspective and faster mobility, allowing them to quickly oversee large areas and relay information to ground units. However, UGVs play a crucial role in close-range navigation, as they operate directly in the terrain, allowing them to easily assist individuals.

Within this context, our framework, A2G-Coord, proposes a collection of algorithms that enable collaboration between UAVs and UGVs in SAR missions. The approach aims to investigate how information exchange between air and ground agents can optimize mission time under opportunistic communication constraints.

In this work, we address key constraints that impact UAV-UGV coordination in SAR missions:

- **GPS-limited:** only the UAVs have GPS, while the UGVs use compass navigation and only have anti-collision systems like Light Detection and Ranging (LIDAR) sensors.
- **Distributed:** UAVs and UGVs communicate and coordinate through message exchange. As a result, the system operates without a central clock.
- **Unknown environment:** the area is unknown to both the UGV and the UAV. In particular, the Points of Interest can be anywhere on the map.
- **Predefined Path planning for aerial coverage and mapping:** each UAV of the swarm has a predefined path planning.

This work is organized as follows: The study of the most recent works is presented in Section 2. The formulation of the problem at hand is described in Section 3. The proposed approach, called A2G-Coord, is detailed in Section 4. The simulation of the approach is described in Section 5. The illustrated results and discussion are shown in Section 6. Finally, the conclusions and next steps of this research are summarized in Section 7.

 DOI: https://doi.org/10.65218/jius.2026.123

II. Related Work

During the search for relevant work, we found works that focus more on UAV-UGV and UAV-UAV collaboration. As we focus on both, this section is divided into two subsections for each type of collaboration.

A. Overview of UAV-UGV collaboration

According to Ding *et al.* [6], UAV-UGV coordination systems can be classified based on the functional roles assumed by aerial and ground agents. The authors identify four primary roles (sensors, actuators, decision makers, and auxiliary facilities) and analyze the different coordination dynamics by alternating and combining these functional roles. Sensors are responsible for detecting events or changes in the environment and sending data to other components or vehicles; actuators are responsible for performing actions or activities; decision makers are responsible for making decisions for other components or vehicles; and auxiliary facilities are responsible for providing agents with services such as energy and communication.

When it comes to combining functional roles, it is important to consider the advantages and limitations of these agents. For example, UAVs excel as sensors and decision makers because they can be deployed to quickly find ground targets, and UGVs excel as actuators and auxiliary facilities because they are closer to points of interest and are less limited by the battery capacity.

The more recent survey by Munasinghe *et al.* [7] expands Ding *et al.* [6] by introducing new classifications of the collaboration between UAVs and UGVs. They explore the different applications of UAV-UGV when it comes to surveillance and monitoring, agriculture, and infrastructure inspection, as well as the limitations and challenges of Air-Ground collaboration towards complex coordination, communication latency and bandwidth constraints, energy and resource management, environment and terrain challenges, computational burden, communication network instability, and embedded hardware limitations.

Building upon Ding *et al.* [6], our work extends the study of UAV-UGV collaboration by focusing on scenarios where UAVs act both as sensors and decision makers, while UGVs function primarily as actuators. In this configuration, UAVs provide UGVs with environmental information and guidance, serving as "eyes in the sky", and command a swarm of UGVs.

Regarding limitations and challenges, and following the categorization proposed by Munasinghey *et al.* [7], our work deals with *Complex Coordination*, as UAVs are much faster than UGV in our scenarios, and *Communication Latency and Bandwidth Constraints*, as all communication between agents is opportunistic.

B. UAV-UGV Coordination approaches

From the papers shown in Ding *et al.* [6], in most of the papers shown, UAVs use aerial cameras to obtain environment information and send them to a swarm of UGVs.

Michael *et al.* [8] propose an decentralized approach for controlling a team of nonholonomic UGVs using aerial robots. They introduce an abstraction that represents the entire UGV swarm through a spanning ellipsoid, capturing their position, orientation and shape.

More recent works have continued to contribute to this area of study, exploring advanced coordination strategies and improving communication frameworks.

Cladera *et al.* [9], based on the work of Miller *et al.* [10], show a collaboration method between UGVs and UAVs in scenarios without communication infrastructure within unexplored areas. Given the limited number of UGVs and time constraints, the UAV is tasked with searching the area, generating a semantic map, and disseminating the data to the UGVs. The UGVs then opportunistically communicate with each other and use the information to locate points of interest.

In Miller *et al.*, the UAVs build a semantic map in real time and opportunistically pass them to the UGVs, allowing them to choose and deconflict their targets without any external intervention. Communication between agents is distributed and opportunistic and UGVs do not have access to GPS.

In Castro *et al.* [11], two UAVs explore the area and update the environment map of a partially mapped area, while one UGV is responsible for updating the map at ground level and having the ability to transport a UAV to its top, changing the UAV battery during an aerial mission.

Wei and Fang [12] use UAVs to search for bushfires and guide UGVs to rescue individuals. UAVs scan the area in a lawnmower pattern for bushfire detection. When they detect them, they alert close residences and generate paths for the UGVs to find victims.

Qin *et al.* [13] combine the agility of UAVs with the computational resources of UGVs to optimize mapping in GPS-denied environments. The UGV explores the environment using 2.5D SLAM to build a preliminary map, which is then refined by the UAV through 3D mapping. The cooperation between UAVs and UGVs is decentralized, that is, each agents maintains their own navigation strategy, building maps independently.

In our first article [14], we focused solely on UAV-UGV coordination. There, we presented initial simulation results that showed that in a UAV-UGV coordination scenario, load distribution is an effective strategy for addressing the problem. However, based on the experiments we conducted, we extended our approach to incorporate UAV-UAV communication.

C. UAV-UAV communication strategies

The use of a single UAV in SAR missions can greatly impact the overall result, as it increases limitations in terms of time efficiency, simultaneous actions, fault tolerance, and flexibility, and limits the mission to a single point of failure. Misra *et al.* [15] demonstrate that in a $n \times n$ grid, using only one UAV to search the area leads to a worst-case time complexity of $O(n^2)$, putting an excessive strain on the UAV's power resources and increasing the mission duration.

Similarly, a system with Multi-UAV that does not effectively collaborate can face the same limitations. The lack of collaboration between UAVs can lead to overlapping search areas, waste of resources such as energy, inefficient path planning, and increase mission time, as UAVs operate in isolation rather than as a coordinated swarm.

Javaid *et al.* [16] argue that effective UAV-UAV communication enable UAV swarms to overcome fundamental challenges, such as autonomous flights, collision avoidance, distributed processing, and joint operations.

Using Multi-UAVs that collaborate during the mission by exchanging information can effectively address these limitations, as they can distribute the computational workload among several agents. This approach allows UAVs to compensate for individ-

DOI: https://doi.org/10.65218/jius.2026.123

ual failures and ensures a more scalable solution as the search area grows [17].

The survey by Kumar *et al.* [18] systemizes the UAV swarm objectives into key functions: Maximal Cover Area, UAV Path Planning, Intra-Swarm Collision Avoidance, Obstacle Avoidance, Swarm Formation, Target Tracking, and Optimal Resource Allocation.

Communication plays a crucial role in enabling such coordination. According to Chen *et al.* [19], the interaction between UAVs can occur in two ways: U-T-U (UAV-to-UAV), which UAVs communicate with each other directly or indirectly, in multi-hop communication paths, and U-T-I (UAV-to-Infrastructure), where UAVs communicate with a ground station to obtain mission information. These two types of communication are combined through swarm communication architectures that can be classified into centralized, decentralized, single-group, multi-group, and multi-layer and ad hoc networks.

The authors argue that centralized architectures, which evolved from single-UAV systems, rely on infrastructure, being more vulnerable to single points of failure. In contrast, decentralized architectures allow UAVs to interact autonomously through U-T-U links.

In a decentralized architecture, many communication protocols can be used in a Multi-UAV scenario, like gossip, flooding, and routing. Flooding protocols can be useful to make sure that all UAVs receive the information, and it can also create a huge number of messages, increasing energy consumption. As UAVs have limited energy, a gossip-based protocol may be more useful as it reduces the number of transmissions. Routing protocols focus on maintaining efficient communication.

Lucchesi *et al.* [20] analyzed how distributed UAV swarms are affected by communication delay, message loss, and leader-speed in leader-follower formations. Their results showed that high speed operations combined with communications challenges present significant challenges. Therefore, in SAR missions, where high speed is key and communications are usually unreliable, ensuring robust UAV-UAV communication becomes a priority.

Ren *et al.* [21] propose a gossip-based broadcast protocol that aims to fix the problem of long propagation delay by improving forward probability and neighbors to enhance communication among UAVs in decentralized networks. To do this, they built a data structure that can record package reception of UAVs and formulated the gossip broadcasting as a partially observable Markov decision process. This approach allows each UAV to simultaneously decide its forwarding probability and neighbor selection, adapting dynamically to topological changes.

Wang *et al.* [22] propose a framework to minimize sensor data collection in a set of two-dimensional distributed sensors using collaborative UAVs. In their approach, UAVs can choose to fly or hover to collect data based on conditions, and each UAV communicates with each other to exchange cluster and trajectory information. In addition to that, UAVs share their remaining energy, flight state, and sensor information to ensure balanced task distribution and prevent overlapping paths. Their results show that the proposed framework using multiple UAVs shortened the total mission time.

Cavalcanti *et al.* [23] proposed a collaborative multi-UAV data fusion approach for SAR involving multiple moving targets in limited time. The proposed approach, called MUDE (Multiple UAVs in a Dynamic search Environment) aims to identify mobile targets with a predictible movement on the ground through the use of an UAV swarm. The UAVs have a predefined path until they detect a target. Following that, they transmit the target's coordinates to the other UAVs in the swarm. This information is used to establish social and individual paths and to determine search areas while eliminating overlapping waypoints.

Liu *et al.* [24] analyzed opportunistic data collection in wireless sensor networks (WSNs). UAVs that pass though the WSN periodically are used to collect the sensing data from the ground servers. As the energy capacity of the UAV is limited, they need to return to the headquarters for charging and updating information periodically.

Recently, the use of machine learning methods has been explored in coordination and communication mechanisms.

Garg and Jha [25] use a decentralized deep reinforcement learning algorithm, called dec-DQNC8, to integrate opportunistic communication and domain knowledge to optimize area coverage during floods. UAVs can only obtain information about the flooded area based on their observation of its local surroundings and the data provided by other UAVs (via opportunistic communication), eliminating the need for a central entity in coordinating the coverage task. The reward in the reinforcement learning algorithm is based on the information obtained by the UAV from the environment. The authors also used an energy model for UAVs, where they require energy to take off, hover, fly, and communicate, limiting the UAVs' actions during the mission.

Lamenza *et al.* [26] investigate collaborative and noncollaborative solutions in UAV swarms for sensor data collection, where sensors are randomly distributed in an area. They propose four different decentralized deep reinforcement learning approaches: Noncollaborative algorithm, Collaborative algorithm, Flexible sensor count algorithm, and Flexible agent count algorithm. The results show that collaborative strategies significantly reduce mission completion time compared to noncollaborative approaches, although they require longer training times due to increase in model complexity.

III. Problem Definition

This work finds its motivation in the problem of time-critical Search and Rescue (SAR) missions using swarms of autonomous agents, where the main goal of these missions is to locate targets, usually individuals, in danger in hazardous environments, and where there is a limited time frame to find and serve the victims, i.e. giving them initial assistance or bringing food/water.

An aerial agent is represented by an Unmanned Aerial Vehicle (UAV), and a ground agent is represented by an Unmanned Ground Vehicle (UGV). We assume that all agents have means of wireless communication with fixed range with other UAV/UGV agents. They communicate to share their partial knowledge of the mission so far, as well as pass directions or positions to other agents.

The environment is represented as a two-dimensional bounded map. Each Points of Interest (POI) represents a group of individuals in danger, and these targets remain stationary throughout the entire mission.

Aerial agents are used to locate targets in situations of forced immobility or life-threatening situations. This is because we assume that they can scan regions quickly, providing a broad overview of the area and identifying possible target locations using cameras, thermal imaging, and other sensors. However, aerial agents may not be capable of landing in the vicinity of the victims, nor delivering the life-saving supplies. On the other hand, ground agents can supposedly move across uneven, flooded, or unstable terrain to perform close-up searches and assist trapped or fragile individuals.

During such SAR missions, GPS signals may be compromised for teams on the ground while remaining accessible on the air. In such environments, UGVs cannot rely on GPS for accurate localization and instead depend on UAVs for directional guidance to reach POIs. Therefore, in this work, UAVs are the only agents with GPS capability.

The assumption that UGVs are GPS-limited is grounded in the operation realities of hostile or unstructured environments. In regions with very dense vegetation and many tree canopies, in rural/urban canyons, or subterranean search areas, the line of sight required for the satellite signals is often compromised at ground level, while wireless communication is still possible. Hence, UAVs flying at higher altitudes may serve as the system's primary navigation anchor.

We also assume that all communication between UAVs, UGVs and POIs happens opportunistically, as the distributed algorithm operates only through peer-ro-peer message exchange between agents. Whenever they come into mutual radio coverage, they exchange information that will help the mission goal.

Given these requirements and assumptions, the problem at hand is to design, implement, and evaluate heuristics and distributed algorithms for coordination among all UAV/UGV agents to minimize the time it takes for all POIs to be visited by at least one UGV. Or else, given a specific time frame (e.g. until the next rainfall), maximize the number of visited POI.

An overview of the problem can be seen in Figure 1.

The UAV-UGV coordination could be implemented using a centralized architecture, in which all agents periodically report their position and status to a control center responsible for computing the next actions for each agent. However, such an approach is often impractical in disaster scenarios, since the communication infrastructure may be unavailable or damaged, network connectivity may be intermittent, and centralized coordination introduces a single point of failure. Furthermore, the latency associated with transmitting global information to a central server may delay critical decisions.

For these reasons, this work adopts a fully decentralized coordination model, where each UAV and UGV makes decisions based only on locally available information and opportunistic communication with nearby agents.

A. Coverage Path Planning

Coverage Path Planning (CPP) is a motion planning strategy for autonomous robots designed to determine the most efficient path for one (or several) robots to cover every point in a defined region, optimizing factors like time, energy, and path length. Common techniques include cellular decomposition, grid-based, and spiral methods motion patterns [27].

In the scope of this work the search area is divided into 12 regions arranged on a grid, and each UAV is assigned to visit a number of these regions. These regions are assigned according to the number of UAVs in a mission. For example, if a mission has two UAVs, each will be assigned 6 regions; if a mission has 3 UAVs, each will be assigned four regions.

Within each assigned region, the UAV follows a boustrophedon path, in which the UAV performs a back-and-forth motion across the grid. This pattern is adequate for the total coverage of an area with minimal overlap [28].

The boustrophedon strategy was chosen because it provides complete coverage of rectangular regions while minimizing overlap between adjacent trajectories, it is computationally simple and deterministic, which is desirable for systems involving multiple agents, and this pattern allows an easy partitioning of the search space among several UAVs, enabling scalable exploration as the number of UAVs increases.

After moving through a region, the UAV flies back to the base station to recharge, as its battery capacity is limited and allows it to cover only a single region per flight.

An example of UAV Coverage Path Planning (CPP) is depicted in Figure 2. It represents two UAVs during a mission: the blue UAV has completed one assigned region, while the green UAV has completed two assigned regions.

B. Mathematical Formalization

The search area is represented by a two-dimensional Euclidean space $E \subset \mathbb{R}^2$, a square grid of size $n \times n$. Each grid cell in this grid may be empty or occupied by a Point of Interest (POI).

Let

$$P = \{p_1, \ldots, p_i\}$$

denote the set of all POIs,

$$A = \{a_1, \ldots, a_j\}$$

the set of UAVs, and

$$G = \{g_1, \ldots, g_k\}$$

the set of UGVs, where i, j and k are the number of POIs, UAVs, and UGVs, respectively.

The movement of the UAVs follows a trajectory based on the Coverage Path Planning described previously.

Each moving agent $u_n \in A \cup G$ occupies a position $pos(u_n, t) = (x_n, y_n)$ at time t. The communication between these agents occurs only when they are within a certain distance r_c, representing the communication range. Therefore, an exchange between two agents u_i and u_j happens if

$$d(u_i, u_j) \leq r_c$$

where $d(u_i, u_j)$ is the Euclidean distance between them.

When an UAV detects a POI, it stores the corresponding coordinates in a local buffer that maintains a list of known POIs. When an UAV encounters an UGV within range, it transfers this information to the UGV based on a strategy, allowing the UGV to navigate toward the POIs.

The mission time is represented by $t \in [0, T]$, where T denotes the total duration of the mission. Each moving agent has a position in the area that changes over time, represented as (x_t, y_t).

1) Objective Function

The main goal of the mission is to minimize the time required to find all POIs giving that each POI has to be visited at least

DOI: https://doi.org/10.65218/jius.2026.123

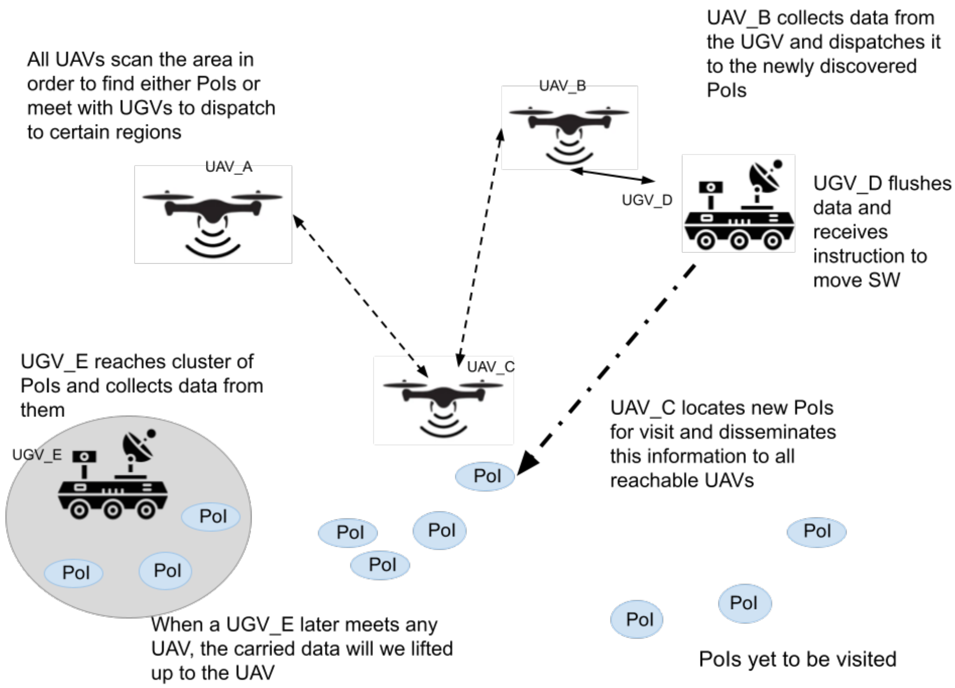


Fig. 1 Problem overview

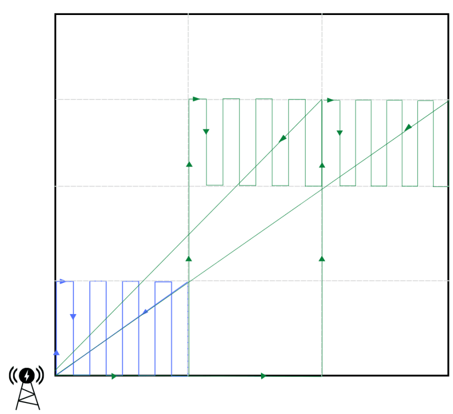


Fig. 2 CPP example showing two UAVs during a mission

once by an UGV. To formalize this metric, we define the moment of the first visit to each Point of Interest (POI).

The visit time $T(p_i)$ for a POI $p_i \in P$ is defined as the earliest time instant t at which at least one UGV $g \in G$ reaches the POI's position:

$$T(p_i) = \min\{t \in [0, T] \mid \exists g \in G, \text{pos}(g, t) = p_i\} \tag{1}$$

Based on this definition, the optimization problem addressed in this work can be divided into two sub-objects:

1. **Minimization of Total Time:** The goal is to minimize the elapsed time until the last POI is attended:

$$\min \left(\max_{p_i \in P} T(p_i) \right) \tag{2}$$

2. **Maximization of Coverage:** Given a fixed time horizon T_{max}, the objective is to maximize the number of POIs that have been visited in this time.

$$\max \sum_{p_i \in P} x_i, \quad \text{where } x_i = \begin{cases} 1, & \text{if } T(p_i) \leq T_{max} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

C. Evaluation of the mission success

Based on our metric of total mission time, we evaluated how the POI discoveries are distributed throughout the mission.

Let $T(P_i)$ denote the first visit time of POI $p_i \in P$, as defined in Equation (3-1), and let T_f denote the total mission time, defined as:

$$T_f = \max_{p_i \in P} T(p_i) \tag{4}$$

To compare discovery times across missions with different durations, we define the normalized discovery time of each POI as:

$$\tau(p_i) = \frac{T(p_i)}{T_f} \tag{5}$$

where $\tau(p_i) \in [0, 1]$ represents the relative moment at which a POI p_i is discovered during the mission.

The set of normalized discovery times captures the temporal distribution os POI discoveries throughout the mission and is defined as:

$$\mathcal{T} = \{\tau(p_1), \tau(p_2), \ldots, \tau(p_{|P|})\} \tag{6}$$

In an ideal mission, each POI should be visited exactly one by an UGV, so we evaluate redundancy to compare the algorithms.

DOI: https://doi.org/10.65218/jius.2026.123

We define redundancy as the number of unnecessary visits performed by UGVs to POIs that have already been visited.

Let $V(p_i)$ denote the total number of times a POI $p_i \in P$ is visited by any UGV during the mission. The redundancy associated with that POI is defined as the number of visits beyond the first one:

$$R(p_i) = \max(0, V(p_i) - 1) \tag{7}$$

The total redundancy of the mission is defined as the sum of redundant visits across all POIs:

$$R_{total} = \sum_{p_i \in P} R(p_i) \tag{8}$$

IV. Proposed Approach

To address the challenges of decentralized control of the defined problem, we introduce the algorithm family A2G-Coord. We divided the algorithms between Local Algorithms and Cooperative Algorithms.

Local algorithms refer to how UAVs pass information to UGVs, whereas Cooperative Algorithms refer to how UAVs exchange information with each other. By combining the two types of algorithms, we have Local-Cooperative Algorithms.

A. Local Algorithms

We define Local Algorithms as the set of coordination strategies that dictate how UAVs interact with UGVs. These algorithms operate under the assumption that each UAV makes decisions independently, based on its knowledge of the positions of the UGVs and POIs aiming to dispatch the UGVs without means of geolocalization toward some POIs.

Since UGVs do not have access to GPS, they rely on UAVs for their navigation guidance, receiving the coordinates of the next points to visit from an UAV. This happens when an air vehicle opportunistically meets with an UGV, which happens when they come into each other's communication range. Therefore, the efficiency of such cooperation is largely dependent on how well the swarm of UAVs distribute such "next PoI to visit" commands to the UGVs.

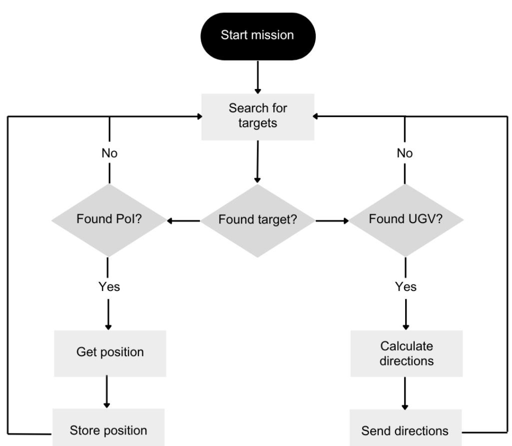


Fig. 3 UAV mission flow chart

We also consider that all UAVs are initially positioned in a single corner of a rectangular search area, where their recharging station is located. In all the simulations we used Boustrophedon Coverage Path[29], where the search area is segmented into cells and the algorithm is applied individually to each geographic section. After completing the navigation of a section, each UAV returns to its starting position to recharge before proceeding to the next section. We used only this Boustrophedon Coverage Path Planning since a deeper study and optimization of CPPs goes beyond the scope of this work.

Furthermore, when more than one UAV is used, the search area is divided into equal parts and one segment is assigned to each UAV. For example, if there are two UAVs, each one covers - and scans - half of the map; while if three UAVs are used, each one becomes responsible for traversing and scanning a third of the map, and so on.

During the mission, each UAV broadcasts messages at regular intervals to search for points of interest or UGVs. When a POI is located, it replies with a message containing its unique identification, POI-ID, and the UAV record its own position in the map at this precise moment. The UAV then records these coordinates in a buffer, known as *POI Buffer*. This buffer is capable of storing all the necessary information, eliminating any concern regarding its size from our solution.

When an UGV is detected, the UAV calculates the direction (angle and distance) from its own position towards the position of the next PoI in the *PoI Buffer*, to direct the UGV to move towards this POI. It sends a copy of *POI Buffer* in a message to the UGV. This process is shown in figure 3.

The UGVs begin their mission at the same location as the UAVs. They move towards the edge of the map in uniformly distributed directions, as shown in Figure 4.

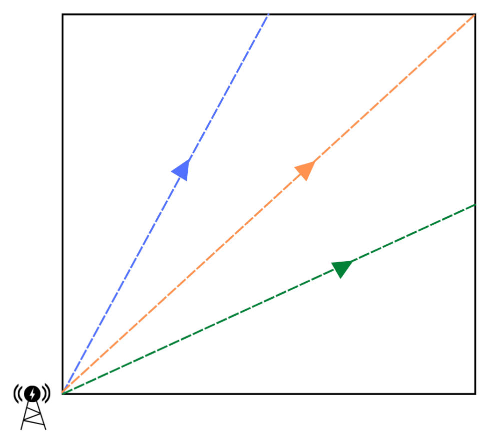


Fig. 4 Initial UGV trajectories for a mission with three UGVs

During this mission, UGVs can communicate with both UAVs and POIs, whenever they are mutually within their radio ranges. When communicating with a UAV, the UGV receives the list of detected POIs(*POI buffer*) that define the sequence of waypoints required to reach them. Based on this information, the UGV begins a new mission, following each direction in the buffer sequentially until all POIs have been visited and the buffer is emptied. Even if a UGV has not received a specific direction to a POI, it will still communicate with the POI if it happens to be along its path. This process is shown in figure 5.

POIs act as beacons, repeatedly broadcasting *"Hallo, I'm here"* messages every few seconds with their unique identification, POI-ID.

The following sections present four different Local Algorithms: Base Algorithm (BA), Greedy Algorithm (GA), Load

 DOI: https://doi.org/10.65218/jius.2026.123

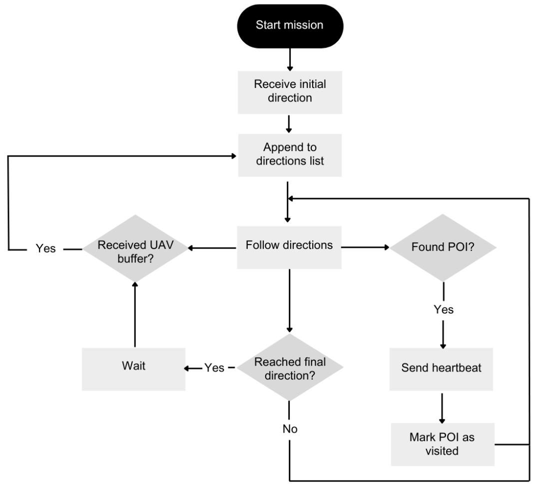


Fig. 5 UGV mission control flow chart

Balancing Algorithm (LBA) and Load Balancing Algorithm Plus (LBA+).

1) Base Algorithm (BA)

The Base Algorithm is the simplest and the most straightforward form of the Local Algorithms. It involves a simple solution in which the UAVs do not distribute the information to the UGVs based on any specific criteria. In this approach, UAVs transmit the complete list of detected POIs (*PoI buffer*) to any UGV that enters their communication range, without applying prioritization, ordering, or filtering rules.

As mentioned previously, each UAV follows its predefined path while continuously broadcasting beacon messages to identify nearby agents. When a POI is detected, the UAV records its own position (x, y) in the POI buffer, that stores all the necessary information about the POIs.

When a UAV encounters a UGV within communication range, it transmits the entire contents of the *PoI buffer* to the UGV. No filtering, prioritization, or reordering is applied to the list of stored POIs. Consequently, all UGVs that interact with the same UAV receive the same list of POIs in the same order.

After receiving the list of POIs, the UGV visits the POIs in the order they appear in the received list, sequentially navigating to each waypoint until all POIs in the buffer have been visited.

Although the BA strategy can be useful for basic coordination between air and ground agents, it presents significant limitations. UGVs that interact with the same UAV will receive the same *PoI buffer*, regardless of their current location, direction, or state. As a result, multiple UGVs can be dispatched to the same POIs, causing redundant and unnecessary visits, and thus leading to suboptimal routes, and inefficient use of UGV energy resources.

In the following, we show the pseudocode 1 that describes the general behavior executed by each UAV during the mission. When a POI is detected, its position is stored in the buffer. When a UGV is detected, the UAV invokes the function *calculateDirections*, which determines the list of directions that will be transmitted to the UGV.

Pseudocode 2 presents the implementation of the function *calculateDirections* in BA. In this case, the function simply returns *PoI buffer* without applying any modification, meaning that all known POIs are sent to the UGV exactly as they were recorded.

Algorithm 1: UAV Behavior

if *foundPOI()* **then**
  $poiPos \leftarrow foundPOI().getPos()$;
  storePos(poiBuffer, poiPos);
else if *foundUGV()* **then**
  $directionsArray \leftarrow calculateDirections(poiBuffer)$;
  broadcastMessage($directionsArray$);

Algorithm 2: Base Algorithm – Calculate Directions

Function `calculateDirections`($poiBuffer$)**:**
  return $poiBuffer$;

2) Greedy Algorithm (GA)

As yet another Local Algorithm, the Greedy Algorithm (GA) introduces the first optimization for the baseline by improving the communication strategy between UAVs and UGVs.

To overcome the issues with BA, instead of broadcasting the entire list of POIs in a random order, in GA, the UAVs create a new list rearranging all the POIs based on proximity to the UGV at hand. Hence, we use a greedy algorithm to rearrange the collected points of interest stored in the *POI buffer*.

The idea is that each UGV will receive directions tailored to its current location, and since all UGVs will receive a personalized ordered set of points, the total time to find all POIs is significantly reduced.

The pseudocode 3 shows how an UAV distributes POI's directions to UGVs. The function orders the iteratively search for the closest POI to the current position and appends it to a new ordered list until all POIs have been processed. The resulting list is then transmitted to the UGV, as seen in 1.

Although GA improves the efficiency of routes by considering spatial proximity, it still does not address the issue of redundancy. Since UAVs do not track which POIs were already assigned to other UGVs, multiple UGVs may still receive directions to the same POIs.

3) Load Balancing Algorithm (LBA)

The Load Balancing Algorithm (LBA) addresses a limitation observed in both BA and GA: the lack of load distribution. In these previous strategies, UAVs may repeatedly send the same POIs to multiple UGVs, which can lead to redundant visits and inefficient use of resources.

To solve this, LBA introduces a simple mechanism that ensures each POI is assigned only by a given UAV. This is achieved by associating a boolean flag with every POI stored in the UAV's *PoI buffer*.

Initially, the flag is set to false, indicating that the POI has not yet been assigned to any UGV. When the UAV encounters a UGV, it iterates through the *PoI buffer* and selects only those POIs whose flag is still set to false. These POIs are added to a temporary list that will be transmitted to the UGV and the flags in the buffer are set to true, indicating that the POI has already been assigned.

In this approach, every time an UAV sends directions to an UGV, it records in *POI Buffer* which UGV the direction was sent to. A flag is added to *POI buffer* to indicate whether a POI's

DOI: https://doi.org/10.65218/jius.2026.123

Algorithm 3: Greedy Algorithm – Calculate Directions

```
Function calculateDirections(poiBuffer):
    orderedBuffer ← list() // Empty list
    remainingPoints ← copy of poiBuffer;
    currentPosition ← getCurrentPosition()
        // UAV position
    while remainingPoints in not empty do
        fPoint ← first point in remainingPoints;
        minDistance ←
         calculateDistance(currentPosition, fPoint);

        foreach point in remainingPoints do
            distance ←
             calculateDistance(currentPosition, point);

            if distance ¡ minDistance then
                closestPoint ← point;
                minDistance ← distance;

        append closestPoint to orderedBuffer;
        currentPosition ← closestPoint;
        remove closestPoint from remainingPoints;
    return orderedBuffer;
```

position has already been sent. An UAV will only send POI directions to a UGV if the flag is set to False.

The pseudocode of Algorithm 4 shows how an UAV distributes POI's directions to UGVs. The function scans the buffer and appends only the unassigned POIs to the message that will be sent to the UGV.

The mechanism prevents UAVs from repeatedly assigning the same POI to different UGVs, reducing redundancy and improving the tasks across ground agents. However, this approach also introduces a new limitation: if a UAV discovers many POIs before encountering any UGV, the first UGV encountered may receive all of them, while other UGVs may receive few or none.

Algorithm 4: Load Balancing Algorithm – Calculate Directions

```
Function calculateDirections(poiBuffer):
    balancedBuffer ← list() // Empty list
    foreach poi in poiBuffer do
        if poi.sent = False then
            poi.sent ← True;
            append poi to balancedBuffer;
    return balancedBuffer;
```

4) Load Balancing Algorithm Plus (LBA+)

The Load Balancing Algorithm PLUs (LBA+) extends LBA by addressing the imbalance that can occur when the first encountered UGV receives a large number of POIs.

In LBA, all unassigned POIs are transmitted to the first UGV encountered by the UAV. This behavior may overload a single UGV while leaving other UGVs underutilized. To resolve this, LBA+ introduces an assignment mechanism based on region.

Instead of sending all unassigned POIs, LBA+ only sends POIs that belong to the current region being explored by the UAV. POIs discovered in other regions remain stored in the UAV's buffer and can be assigned later to other UGVs that are geographically closer to those locations.

The pseudocode 5 shows how an UAV distributes POI's directions to UGVs. For each POI in the buffer, the function checks whether the POI lies within the UAV's current section (region). If this condition is satisfied, the POI is added to the list of directions sent to the UGV and marked as assigned.

This mechanism reduces the likelihood that a single UGV becomes overloaded and helps maintain a more balanced allocation of POIs across all ground agents.

Algorithm 5: Load Balancing Plus Algorithm – Calculate Directions

```
Function calculateDirections(poiBuffer):
    balancedBuffer ← list() // Empty list
    foreach poi in poiBuffer do
        section ← isInCurrentSection();
        if section = True then
            poi.sent ← True;
            append poi to balancedBuffer;
    return balancedBuffer;
```

B. Cooperative Algorithms

Cooperative Algorithms are designed to enable coordination among UAVs. In this context, UAVs act as information sharing agents that help propagate knowledge about the environment and discovered POIs to other UAVs operating in different regions of the map.

The following sections present two different Local Algorithms: Centralized Distributed Algorithm (CDA) and Base Station Algorithm (BSA).

1) Centralized Distributed Algorithm (CDA)

During the mission, UAVs will eventually come within communication range as they carry out their exploration of the map. The Centralized Distributed Algorithm (CDA) makes use of these encounters to exchange information about the mission among UAVs.

Predefined rendezvous points are predefined on the map, which serve as encounter spots for UAVs. Each UAV maintains an internal clock that triggers the encounter event. When the UAVs come within range, they start they rendezvous process.

UAVs periodically meet at predefined rendezvous points to exchange POI data when they come within communication range. Each UAV maintains an internal clock that triggers the event.

The mission of the UAV in the CDA algorithm in shown in Figure 6.

2) Base Station Algorithm (BSA)

In the Base Station Algorithm (BSA), each UAV stores the POIs it discovers during its mission and uploads them to a base station. The base station accumulates all the data collected by the UAVs and makes it available for future access.

The base station is placed in the corner of the map and since the UAVs are already required to return there for battery

DOI: https://doi.org/10.65218/jius.2026.123

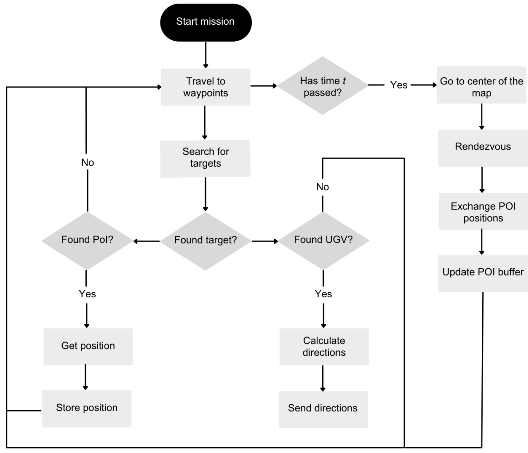


Fig. 6 CDA UAV mission flow chart

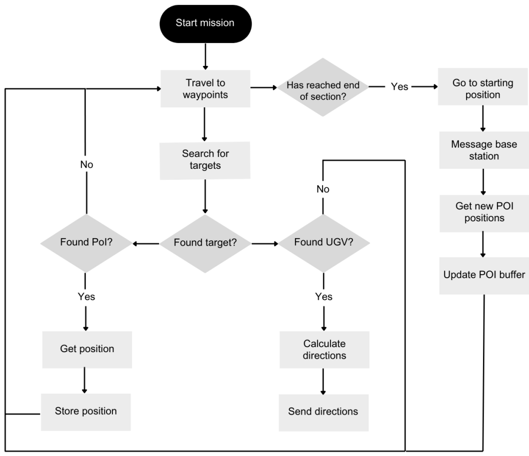


Fig. 7 BSA UAV mission flow chart

recharge, going back to the base station does not represent a deviation from their planned paths. However, the base station presents limitations as it can become a potential single point of failure.

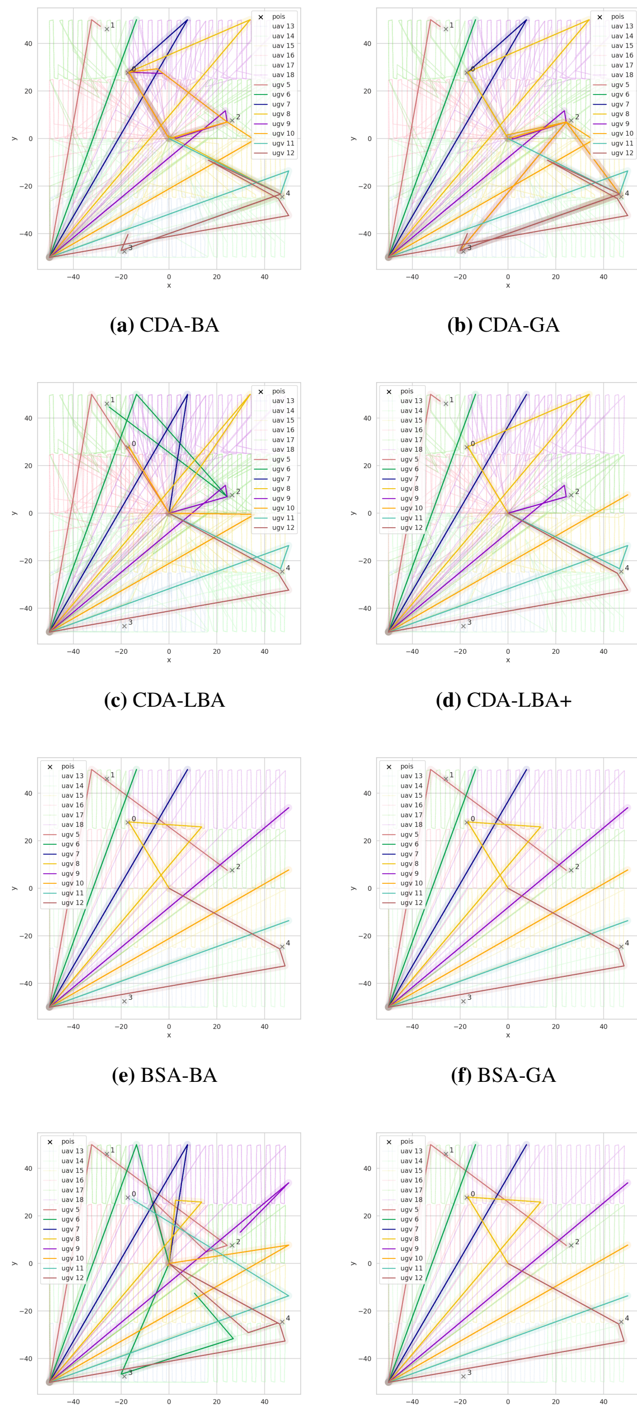
The mission of the UAV in the BSA algorithm in shown in Figure 7.

C. Local-Cooperative Algorithms

Although Local Algorithms provide mechanisms for guiding UGVs toward POIs, and cooperative algorithms focus on UAV-to-UAV coordination, both strategies can be combined to further improve overall mission performance. By combining both algorithms, UAVs can achieve a more efficient distribution of POIs among UGVs.

Using examples from our simulations, shown in Figure 8 a, it is possible to get a better idea of how the algorithms work. In the images, we can visualize six UAVs, represented by thin colored trajectories covering the different sections of the map, eight UGVs, depicted in thicker colored lines, and five POIs, represented by the black "X"s.

Figures 8(c) and 8(g) show a more chaotic trajectory for UGVs, suggesting that most of the UGVs have received information from UAVs and searched for POIs during the mission, while in the other algorithms, only a few UGVs participate in the mission, with several of them remaining idle.



(a) CDA-BA **(b)** CDA-GA

(c) CDA-LBA **(d)** CDA-LBA+

(e) BSA-BA **(f)** BSA-GA

(g) BSA-LBA **(h)** BSA-LBA+

Fig. 8 Selected simulation results exemplifying the different algorithms

V. Experimental Setup

In order to test and validate our approach, we used GrADyS-SIM NextGen[1] [30], a Python-based framework to simulate distributed algorithms operating on cooperating mobile nodes and swarms, developed in the Laboratory for Advanced Collaboration (LAC) at PUC-Rio.

[1] https://github.com/Project-GrADyS/gradys-sim-nextgen

 DOI: https://doi.org/10.65218/jius.2026.123

The framework is easy to use and is suited to simulate all types of mobility, communication, and delay behavior of groups of nodes. A singular feature of GrADyS-SIM NextGen is that it allows the same simulation code to be translated into MAVLink/Ardupilot, OMNeT++, for more realistic simulations, and to be deployed in drones and rovers for experiments in real-world settings.

A. GrADys-SIM NextGen

The simulator consists of three components: Mobility, responsible for controlling the node's movements, Communication, responsible for controlling communication between nodes, and Protocol, that manages the interaction between mobility and communication.

The Protocol component defines nodes, known as protocols, in the simulation. In our simulation, we define three protocols: UGV, UAV and POI.

These protocols implement methods that are able to send and receive messages to other protocols, move to waypoints on the 3D map and gather information about the current state of the protocol.

B. Simulation Parameters

Our simulation experiments involved testing several configurations for each Local-Cooperative algorithm combination. The scenarios were designed to evaluate a variety of behaviors in SAR missions.

We considered cases of sparse POI distributions that resemble cases such as locating isolated individuals in a wide area, while dense distributions of POIs simulate groups of individuals in an area after earthquakes or floods.

We also varied the number of UAVs and UGVs to assess the efficiency of the algorithms. Scenarios with many UAVs were used to test the capability of the algorithms to coordinate UAVs. Finally, we investigated the impact of varying the communication range of the UAV.

The values of the parameters used are summarized in Table 1. In total, we evaluated 81 distinct parameter configurations, each one executed 15 times. This results in 1215 simulation runs per algorithm that have the same POIs configurations. This number of configurations was chosen to represent a wide range of possible scenarios encountered by the algorithm [2].

Table 1 Simulation Parameters

Parameter	Values	Unit
Number of UGVs	2, 4 and 8	units
Number of UAVs	2, 6 and 12	units
UAV Communication Range	5, 10 and 20	m
POI Density	0.05, 0.15 and 0.25	points/m^2

For environment parameters, shown in Table 2, we did not vary the values during the missions.

C. Metrics

To compare the results of our simulations, we chose the time based metric of Average of the Total Time to Find All POI.

The Average of the Total Time to Find All POI measures how long it takes for the last POI to be visited by at least one UGV. In the Search and Rescue context, it means that a mission is only considered complete when all victims have been located. For each simulation, we collected the total mission time and compared all scenarios.

Table 2 Environment Parameters

Parameter	Values	Unit
UAV speed	5	m/s
UGV speed	1	m/s
UGV Communication Range	3	m
Area	100	m^2

However, some algorithms may find most POIs early in the mission but spend a long time finding the last few, whereas others may find only a few POIs initially and find the majority near the end. In this case, considering the SAR context, it may be better to have an algorithm that finds the majority of POIs quickly. To capture this, we collected the individual time each POI was found to evaluate how POIs detections are distributed throughout the mission time.

Finally, we collected how many times a POI is visited by each UGV, to compare redundancy between all algorithms.

VI. Results and Discussion

We compared all eight Local-Cooperative algorithms, using the metric discussed previously.

Figure 9 summarizes the experimental results across the evaluated configurations, varying four parameters presented in Section 5 (number of UGVs, number of UAVs, POI density and communication range). The metric is the Average Time, which measures the total time required for all POIs to be visited by at least one UGV.

In Figure 9(a), we see the results for the parameter of the number of UGVs. As the number of UGVs increases, the median time decreases for all algorithms. This happens because adding more UGVs enables simultaneous visitation of multiple POIs.

When comparing CDA based algorithms and BSA based algorithms, BSA based approaches have lower medians for a smaller number of UGVs, but for a higher number of UGVs they have similar medians.

Algorithms that use CDA as the Cooperative Algorithm have larger spreads than BSA, which means that BSA approaches show more consistent performance. This spread occurs because CDA depends on opportunistic rendezvous between UAVs. If two UAVs miss each other by only a few seconds, the information exchange is delayed for an entire rendezvous cycle, introducing stochasticity into the mission.

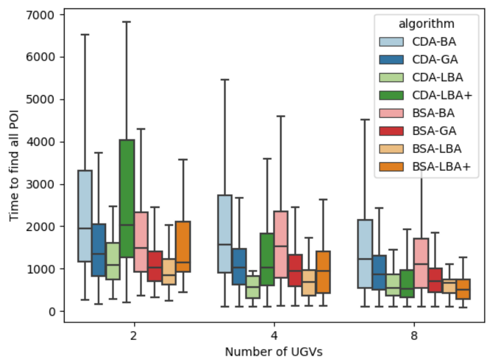
On the other hand, BSA is deterministic, as every UAV returns to the base station to exchange information. This periodicity creates a mission heartbeat, explaining the shorter and more predictable spreads observed in BSA boxplots. Although the base station provides greater consistency and reliability, it represents a single point of failure, since if it fails, the UAVs will not be able to communicate and the information may be lost.

As the number of UGVs grows, the medians are more influenced by the Local Algorithms. LBA and LBA+ show the lowest medians, showing that they consistently outperform the other Local Algorithms.

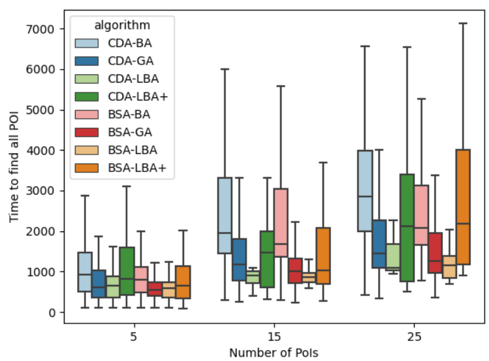
For BA algorithms (CDA-BA and BSA-BA), the decrease in mission time is lower than for the other algorithms, since there is no priorization when the UAVs transmit the detected POIs to UGVs. GA shows a more significant reduction in median time

[2]The source code is available at https://github.com/Project-GrADyS/A2G-Coord

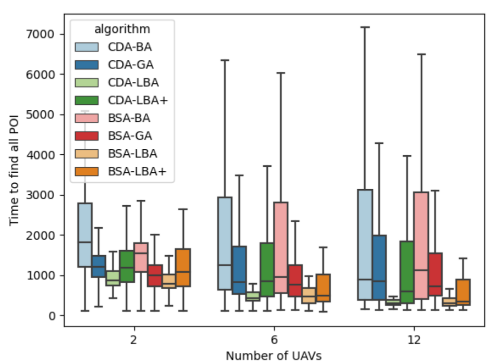
DOI: https://doi.org/10.65218/jius.2026.123



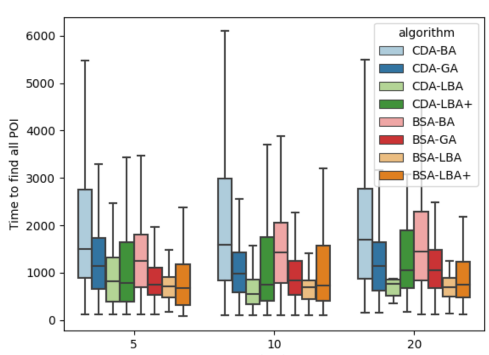
(a) Number of UGVs vs Average Time



(b) Number of POIs vs Average Time



(c) Number of UAVs vs Average Time



(d) Communication Range vs Average Time

Fig. 9 Experimental results

due to its greedy ordering of POIs based on proximity. LBA and LBA+ further improve performance as they demonstrate lower medians and narrower boxes, with LBA+ achieving the lowest medians overall.

As expected, in Figure 9(b), increasing the number of POIs (and therefore increasing POI density) shows an increase in mission time as a bigger number of POIs need to be found and visited. However, the rate of this increase depends heavily on the Local Algorithms. For algorithms with BA as the Local Algorithm, the medians grow fast and the interquartile ranges expand considerably, meaning that the algorithm does not handle multiple POIs efficiently. This could be explained by the fact that UGVs can repeatedly visit already discovered POIs or travel to POIs that are far away first.

As POI density increases, BA and GA experience the worst performance. This redundancy leads to information flooding, causing these algorithms to collapse under high-density scenarios, as the system spends a considerable portion of time processing and revisiting already known targets. In contrast, LBA and LBA+ demonstrate strong scalability by distributing POIs among UGVs and avoiding redundancy.

LBA and LBA+ display smaller medians and their performance remains stable even when the number of POIs doubles, showing that load balancing prevents excessive travel to waypoints.

The improvement of LBA+ over LBA arises from its task allocation mechanism. In LBA, when an UAV meets an UGV, it transfers all unsent POIS, which can cause that UGV to become overloaded while the others remain idle. LBA+ mitigates this by distributing POIs only from the current region, keeping additional POIs on hold to be assigned later to geographically closer UGVs. This distribution balances workload and reduces overall mission time.

When analyzing the parameter of the number of UAVs, in Figure 9(c), the graph reveals a decrease in mission time as the number of UAVs increases. That explanation here is similar to the number of UGVs parameter, with more UAVs, the spatial coverage is improved, and it takes less time to discover all POIs.

When examining the Communication Range parameter, in Figure 9(d), the graph shows that this parameter has a smaller impact on the mission time compared to the others. A larger communication range increases the likelihood of successful interactions between UAVs and UGVs, as well as among UAVs themselves. Consequently, information about detected POIs is disseminated more quickly among agents.

The most significant improvements occur when transitioning from 5m to 10m. In these scenarios, small increases in range significantly reduce communication delays and enable more frequent data exchanges. Beyond a certain threshold, further increases in communication range do not impact as much the mission time.

This parameter has a stronger impact on CDA-based algorithms, since CDA relies on opportunistic encounters between UAVs for information exchange. Increasing the communication range directly increases the probability of encounters, reducing delays in data propagation.

In contrast, BSA-based algorithms are less sensitive to variations in communication range since information exchange occurs when UAVs return to the base station.

Figure 10 shows the time distribution of POI discoveries by

 DOI: https://doi.org/10.65218/jius.2026.123

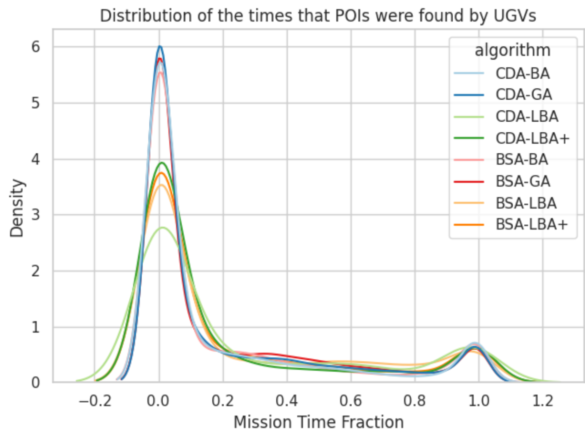


Fig. 10 Distribution of the times to discover POIs

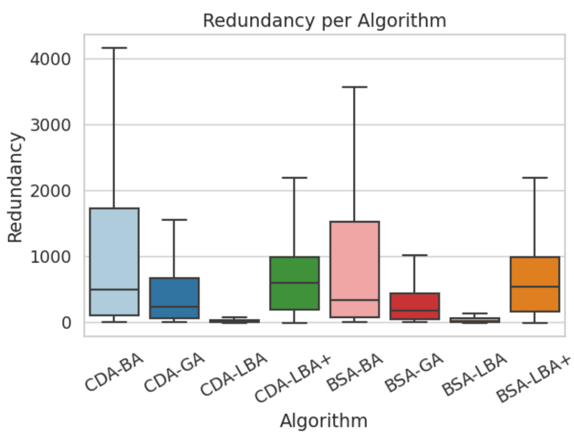


Fig. 11 Redundancy of the algorithms

the UGVs. It evaluates the efficiency of the algorithms in detecting POIs during the early stages of the mission.

We recorded every simulation time when each POI was detected for the first time and normalized it by the total mission time. Using Kernel Density Estimation (KDE), we generated a continuous probability density function that reveals the most frequent moments of discovery. Algorithms that show a higher density near the start of the mission(0.0) tend to locate POIs quickly.

BA and GA algorithms show a higher density near 0.0, meaning these algorithms locate many POIs very early. However, they take a long time to find the last POIs. This pattern reflects the phenomenon of local resource exhaustion, since without distributing the workload, UGVs converge on the easiest nearby POIs, leaving distant POIs unattended. Eventually, a single UGVs must travel across the map to reach the remaining POIs, increasing total time.

It reveals that even though BA and GA algorithms have that highest total time to find all POIs, based on the previous metric, they find most of the POIs quicker than LBA and LBA+ based algorithms.

In contrast, LBA and LBA+ sacrifice early speed to achieve better geographic distribution. Their discovery curves are smoother and more uniform, indicating that UGVs remain active throughout the area instead of clustering. For SAR missions where “no victim can be left behind”, the LBA approach is better suited since it provides a consistent mission and prevents the system from collapsing under the weight of information redundancy, ensuring that all the victims are reached in the shortest possible time.

Figure 11 shows boxplots of the algorithms based on their redundancy when detecting POIs. This means that algorithms with more redundancy are less effective because one POI is visited by multiple UGVs.

This redundancy is the direct cause for the worst results shown in Figure 9, since the most redundant algorithms (BA and GA) have the worst Average Time. When the UGVs visit the same POIs, it is a waste of resources, such as battery, and a waste of movement. While they were visiting this already visited POIs, they could be visiting other distant POIs. In the end, it causes delays on the mission.

Conversely, LBA maintains low redundancy since UAVs deliberately distribute POIs among UAVs.

VII. Conclusion

This work presented A2G-Coord, a family of distributed algorithms designed to coordinate UAVs and UGVs in search and rescue missions. The proposed approach explores how aerial agents can efficiently guide ground vehicles toward victims in environments where communication is opportunistic and GPS is unavailable to ground units. This problem is relevant, as SAR operations often occur in challenging and unpredictable scenarios, where fast and efficient coordination can directly impact mission success.

Given that unmanned vehicles are increasingly employed in such missions, it is advantageous to use UAVs together with UGVs, as they offer complementary capabilities in terms of mobility, sensing, capacity/ ability to transport water and first-aid supplies to the victims in a SAR.

Through simulations, we evaluated eight algorithms of A2G-Coord under varying conditions, including different number of UAVs, UGVs, communication ranges, and POI densities. We collected metrics during the simulations and all algorithms were compared under the same conditions.

Based on the results, we concluded that the performance of each algorithm is strongly dependent on the characteristics of the scenario.

Algorithms that use BA and GA as Local Algorithms are more suitable for scenarios where there is urgency in locating as many individuals as possible in a short amount of time. In these cases, the priority is rapid coverage rather than efficiency and redundancy is acceptable if it increases chances of quickly detecting victims.

On the other hand, algorithms based on LBA and LBA+ are more appropriate for scenarios where it is critical to ensure that no individual is left behind. These approaches focus on reducing redundancy and improving load balancing among UGVs, leading to a more through exploration of the environment as they provide a more reliable coverage.

When it comes to Cooperative Algorithms, BSA-based approaches are more suited for scenarios where reliability and predictability are essential to the success of the mission, while CDA-based approaches are suited for scenarios where the infrastructure is compromised.

It is important to emphasize that many real world aspects were omitted in this work. The communication model assumes ideal

DOI: https://doi.org/10.65218/jius.2026.123

conditions, without considering packet loss and interferences. Furthermore, the current system assumes static POIs, but in real world scenarios individuals in danger may move from time to time. Finally, obstacles were not represented in our simulations, which can impact ground agents. As such, this work represents an initial step toward air-to-ground coordination strategies.

This work can be extended to include real world scenarios such as adding energy constrains (for either UAV or UGV, or both), and problems of the wireless communication, such message loss, communication latency, obstruction and interferences in the wireless communication.

The algorithm family can also be extended by adding new algorithms to create more efficient Local or Cooperative Algorithms, or combining even more the algorithms that are already available. It is also possible to introduce adaptive mechanisms capable of selecting the most appropriate algorithm based on the characteristics of the scenario.

Finally, after more realistic simulations, A2G-Coord can be deployed in real world experiments. The use of the same code in both the simulator and the GrADyS-SIM NextGen framework facilitates this transition, enabling a smoother path from simulation to practical implementation. Real-world scenarios will reveal new challenges that were not identified during the simulations.

Acknowledgment

This study was financed in part by AFOSR grant FA8655-25-1-8035.

References

[1] S. M. S. Mohd Daud, M. Y. P. Mohd Yusof, C. C. Heo, L. S. Khoo, M. K. Chainchel Singh, M. S. Mahmood, and H. Nawawi, "Applications of drone in disaster management: A scoping review," *Science Justice*, vol. 62, no. 1, pp. 30–42, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1355030621001477

[2] E. T. Alotaibi, S. S. Alqefari, and A. Koubaa, "Lsar: Multi-uav collaboration for search and rescue missions," *IEEE Access*, vol. 7, pp. 55 817–55 832, 2019.

[3] K. Berns, A. Nezhadfard, M. Tosa, H. Balta, and G. D. Cubber, "Unmanned ground robots for rescue tasks," in *Search Rescue Robotics From Theory to Practice*. London, U.K: IntechOpen, 2017.

[4] J. P. Queralta, J. Taipalmaa, B. Can Pullinen, V. K. Sarker, T. Nguyen Gia, H. Tenhunen, M. Gabbouj, J. Raitoharju, and T. Westerlund, "Collaborative multi-robot search and rescue: Planning, coordination, perception, and active vision," *IEEE Access*, vol. 8, pp. 191 617–191 643, 2020.

[5] S. Ramasamy, J.-P. F. Reddinger, J. M. Dotterweich, M. A. Childers, and P. A. Bhounsule, "Cooperative route planning of multiple fuel-constrained unmanned aerial vehicles with recharging on an unmanned ground vehicle," in *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2021, pp. 155–164.

[6] Y. Ding, B. Xin, and J. Chen, "A review of recent advances in coordination between unmanned aerial and ground vehicles," *Unmanned Systems*, vol. 09, no. 02, pp. 97–117, 2021. [Online]. Available: https://doi.org/10.1142/S2301385021500084

[7] I. Munasinghe, A. Perera, and R. C. Deo, "A comprehensive review of uav-ugv collaboration: Advancements and challenges," *Journal of Sensor and Actuator Networks*, vol. 13, no. 6, p. 81, 2024.

[8] N. Michael, J. Fink, and V. Kumar, "Controlling a team of ground robots via an aerial robot," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007, pp. 965–970.

[9] F. Cladera, Z. Ravichandran, I. D. Miller, M. A. Hsieh, C. Taylor, and V. Kumar, "Enabling large-scale heterogeneous collaboration with opportunistic communications," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 2610–2616.

[10] I. D. Miller, F. Cladera, T. Smith, C. J. Taylor, and V. Kumar, "Stronger together: Air-ground robotic collaboration using semantics," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9643–9650, 2022.

[11] G. G. de Castro, T. M. Santos, F. A. Andrade, J. Lima, D. B. Haddad, L. d. M. Honório, and M. F. Pinto, "Heterogeneous multi-robot collaboration for coverage path planning in partially known dynamic environments," *Machines*, vol. 12, no. 3, p. 200, 2024.

[12] J. Wei and Z. Fang, "An optimal uav and ugv cooperative network navigation algorithm for bushfire surveillance and disaster relief," in *2024 16th International Conference on Computer and Automation Engineering (ICCAE)*, 2024, pp. 636–641.

[13] H. Qin, Z. Meng, W. Meng, X. Chen, H. Sun, F. Lin, and M. H. Ang, "Autonomous exploration and mapping system using heterogeneous uavs and ugvs in gps-denied environments," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1339–1350, 2019.

[14] T. Reimer, B. Olivieri, M. Cavalcanti, and M. Endler, "From air to ground: Coordinating uavs and ugvs in sar missions," in *2025 28th International Conference on Information Fusion (FUSION)*, 2025, pp. 1–6.

[15] S. Misra, A. Mukherjee, A. U. Rahman, and N. S. Raghuwanshi, "Rose: Random opportunistic and selective exploration for cooperative edge swarm of uavs," in *2020 International Conference on COMmunication Systems NETworkS (COMSNETS)*, 2020, pp. 368–374.

[16] S. Javaid, N. Saeed, Z. Qadir, H. Fahim, B. He, H. Song, and M. Bilal, "Communication and control in collaborative uavs: Recent advances and future trends," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 6, pp. 5719–5739, 2023.

[17] S. A. Ghauri, M. Sarfraz, R. A. Qamar, M. F. Sohail, and S. A. Khan, "A review of multi-uav task allocation algorithms for a search and rescue scenario," *Journal of Sensor and Actuator Networks*, vol. 13, no. 5, 2024. [Online]. Available: https://www.mdpi.com/2224-2708/13/5/47

[18] P. A. Kumar, N. Manoj, N. Sudheer, P. P. Bhat, A. Arya, and R. Sharma, "Uav swarm objectives: A critical analysis and comprehensive review," *SN Computer Science*, vol. 5, no. 6, p. 764, 2024.

[19] X. Chen, J. Tang, and S. Lao, "Review of unmanned aerial vehicle swarm communication architectures and routing protocols," *Applied Sciences*, vol. 10, no. 10, p. 3661, 2020.

[20] L. Lucchesi, M. Endler, and B. Souza, "Dynamic effects of communication delay, failure rates, and speed on uav swarm formation," in *Anais do XLII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*. Porto Alegre, RS, Brasil: SBC, 2024, pp. 588–601. [Online]. Available: https://sol.sbc.org.br/index.php/sbrc/article/view/29821

[21] Z. Ren, X. Li, Y. Miao, Z. Li, Z. Wang, M. Zhu, X. Liu, and R. H. Deng, "Intelligent adaptive gossip-based broadcast protocol for uav-mec using multi-agent deep reinforcement learning," *IEEE*

DOI: https://doi.org/10.65218/jius.2026.123

Transactions on Mobile Computing, vol. 23, no. 6, pp. 6563–6578, 2024.

[22] Y. Wang, Z. Hu, X. Wen, Z. Lu, and J. Miao, "Minimizing data collection time with collaborative uavs in wireless sensor networks," *IEEE Access*, vol. 8, pp. 98 659–98 669, 2020.

[23] M. Cavalcanti, B. Olivieri, T. Lamenza, and M. Endler, "Collaborative multi-uav data fusion for sar applications with moving targets," in *2025 28th International Conference on Information Fusion (FUSION)*, 2025, pp. 1–8.

[24] D. Liu, Y. Xu, Y. Xu, Y. Sun, A. Anpalagan, Q. Wu, and Y. Luo, "Opportunistic data collection in cognitive wireless sensor networks: Air–ground collaborative online planning," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8837–8851, 2020.

[25] A. Garg and S. S. Jha, "Decentralized critical area coverage using multi-uav system with guided explorations during floods," in *2023 IEEE 19th International Conference on Automation Science and Engineering (CASE)*, 2023, pp. 1–6.

[26] T. de Souza Lamenza, B. José Olivieri de Souza, and M. Endler, "Deep reinforcement learning approaches for sensor data collection by a swarm of uavs," *Advanced Intelligent Systems*, vol. n/a, no. n/a, p. e202501146, 2026. [Online]. Available: https://advanced.onlinelibrary.wiley.com/doi/abs/10.1002/aisy.202501146

[27] K. P. Jayalakshmi, V. G. Nair, and D. Sathish, "A comprehensive survey on coverage path planning for mobile robots in dynamic environments," *IEEE Access*, vol. 13, pp. 60 158–60 185, 2025.

[28] G. Fevgas, T. Lagkas, V. Argyriou, and P. Sarigiannidis, "Coverage path planning methods focusing on energy efficient and cooperative strategies for unmanned aerial vehicles," *Sensors*, vol. 22, no. 3, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/3/1235

[29] R. Bähnemann, N. Lawrance, J. J. Chung, M. Pantic, R. Siegwart, and J. Nieto, *Revisiting Boustrophedon Coverage Path Planning as a Generalized Traveling Salesman Problem*. Springer Singapore, 2021, p. 277–290.

[30] T. Lamenza, J. Kamysek, B. Souza, and M. Endler, "Developing algorithms for the internet of flying things through environments with varying degrees of realism," in *Anais Estendidos do XLII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*. Porto Alegre, RS, Brasil: SBC, 2024, pp. 33–40.