Integrating Ground Communication for Extended Drone Visual Line of Sight

Francesco Betti Sorbelli, Sajjad Ghobadi, Lorenzo Palazzetti, and Cristina M. Pinotti
Department of Computer Science and Mathematics, University of Perugia, Italy
Email: {name.surname}@unipg.it

***Abstract*—Unmanned Aerial Vehicles (UAVs) are increasingly permitted to operate within Visual Line of Sight (VLoS) under EU and US regulations. However, Beyond Visual Line of Sight (BVLoS) operations remain restricted, with waivers or certifications required. Extended Visual Line of Sight (EVLoS) offers a transitional solution, involving trained observers to assist pilots when visibility is obstructed. We propose enhancing EVLoS by integrating ground infrastructure, specifically city cameras and wireless communication networks already available on the ground, to replace human observers and enable BVLoS capabilities. Fixed and mobile cameras track drones to ensure regulatory compliance, while real-time data transmission via communication networks provides indirect oversight. The approach increases operational range, reliability, and redundancy through multi-hop connectivity. We introduce the *Minimum Latency Problem* (MLP), a UAV multi-trajectory optimization problem where UAVs are constantly tracked and monitored through ground antennas and city cameras, mimicking the human observers in EVLoS. Our goal is to minimize communication latency while ensuring that the number of antennas used for coverage is minimum. We prove MLP is NP-hard and propose an algorithm to solve it. Experiments on synthetic data demonstrate the effectiveness of our approach in matching coverage and latency requirements.**

I. Introduction

In Europe, Unmanned Aerial Vehicles (UAVs), or drones, have been permitted to fly in *Visual Line of Sight* (VLoS) since the EU Drone Regulations took effect on 2020 [1]. In the US, the Federal Aviation Administration (FAA) has allowed VLoS operations under Part 107 since 2016 [2]. These rules define VLoS as maintaining **unaided** visual contact with the drone throughout the flight to track its position, control its trajectory, and monitor airspace for hazards.

For ***Beyond*** *Visual Line of Sight* (BVLoS) operations, the FAA grants waivers case by case, requiring operators to prove safety compliance. As of May 2025, the FAA is developing Part 108 [3] to standardize BVLoS operations, enabling broader commercial applications like delivery, infrastructure inspection, and agriculture. In Europe, BVLoS falls under the "Specific" or "Certified" categories [4], based on risk level (EU 2019/947). The *Specific Category* requires a Specific Operations Risk Assessment (SORA) [5] and National Aviation Authority (NAA) approval. The *Certified Category* applies to high-risk operations, necessitating certification for the drone, operator, and sometimes for the remote pilot. Predefined risk assessments (i.e., SORA-based Standard Scenarios) have been introduced to streamline approvals. To bridge the gap toward BVLoS, ***Extended*** *Visual Line of Sight* (EVLoS) operations [6] have been considered under FAA Part 107 and EASA regulations. In EVLoS, trained observers maintain visual contact with the drone beyond the pilot's direct line of sight. Key elements include [7]: i) observers assisting the remote pilot when obstacles or distance obstruct visibility, ii) reliable communication between pilots and observers for real-time situational awareness, and iii) strategic observer placement and predefined protocols for contingency management.

In this paper, we propose *enhancing* EVLoS operations by integrating existing ground infrastructures, including **city cameras** and **communication networks**, to enable a transition toward BVLoS. Human observers are replaced with fixed cameras that transmit real-time footage to remote pilots and support command exchange. This solution offers several benefits: i) *Observer Substitution*: Cameras can track drones and maintain EVLoS compliance without human presence. ii) *Extended Range*: Network connectivity allows for seamless BVLoS operations beyond the pilot's direct view. iii) *Redundancy and Reliability*: Multiple camera perspectives reduce the risk of visual obstruction. iv) *Real-Time Data Transmission*: Ground networks ensure immediate relay of visual data, improving situational awareness. By combining visual infrastructure with communication systems, remote pilots gain an enhanced field of view that compensates for blind spots and improves operational control. Vision-based algorithms can further increase safety by autonomously tracking drones and issuing alerts in case of anomalies.

Our contributions are:

- We introduce the *Minimum Latency Problem* (MLP), an NP-hard problem in which multiple UAVs are monitored by city cameras and communicate with a remote pilot via ground antennas. The goal is to minimize communication latency while using the minimum number of antennas required to ensure full coverage of UAV trajectories.
- We evaluate our algorithms on synthetic data, modeling the ground network as a WiFi-based city surveillance system. Experiments assess antenna count and communication time under realistic random geometric graph (RGG) and grid assumptions.

This work was supported by the PRIN 2022 PNRR M4.C2.1.3 "BREADCRUMBS" project, funded by the EU – Next Gen EU, grant n. P2022K7ERB, CUP J53D23014990001, and by the SERICS PNRR M4.C2.1.3 "COVERT-Open Call" project, funded by the EU – Next Gen EU, grant n. PE00000014, CUP J93C23002310006. All authors are members of the GNCS-INdAM.

979-8-3503-9281-4/25/$31.00 ©2025 IEEE

The paper is organized as follows. Section II reviews related work. Section III defines the problem and solution. Section IV-A outlines the experimental setup. Section IV-B presents performance results. Section V concludes the paper.

II. Related Work

BVLoS operations offer UAVs significant autonomy, allowing flights beyond the pilot's direct visual contact. However, these operations require robust communication, navigation, and risk mitigation strategies to comply with evolving regulatory frameworks. To enable safe and reliable BVLoS missions, the work in [8] introduces multi-layer graph-based frameworks for UAV path planning, incorporating factors such as cellular network coverage, ground risk maps, and obstacle avoidance. The study demonstrates how real-world constraints can be modeled into connectivity-aware UAV routing algorithms, ensuring both safety and efficiency. Another critical aspect is flight scheduling for BVLoS corridors, where UAVs are assigned time slots to traverse controlled airspace while avoiding conflicts. In [9], scheduling strategies for unidirectional and bidirectional BVLoS corridors were proposed, optimizing traffic flow and minimizing delays.

Beyond theoretical frameworks, BVLoS has been successfully applied in real-world scenarios. For instance, the authors in [10] details UAV BVLoS operations in Antarctica, where drones mapped $7.5\,\mathrm{km}^2$ of fragile ecosystems at $350\,\mathrm{m}$ altitude, surpassing satellite resolution. Similarly, the work in [11] explores BVLoS for forest management, highlighting its potential for monitoring large, remote areas while identifying technological gaps in autonomy and collision avoidance.

Given the regulatory barriers and technical challenges of full BVLoS adoption, EVLoS serves as a practical bridge by allowing UAVs to fly beyond the pilot's direct sight while maintaining indirect oversight. Traditionally, EVLoS relies on human observers stationed along the UAV's flight path, relaying situational awareness to the pilot [6]. On the other hand, emerging smart city surveillance systems, integrating 5G, WiFi, and networked cameras, are increasingly becoming a reality. Several EU-funded projects have demonstrated how urban camera networks can provide real-time situational awareness in high-density environments [12], [13], [14].

Beyond 5G, WiFi-based sensing offers a privacy-preserving alternative to traditional CCTV for urban surveillance. By exploiting Channel State Information, it detects movement and human activity without using cameras, thereby protecting privacy. Recent studies demonstrate its effectiveness in outdoor settings, achieving up to 80% accuracy with low-cost IoT devices [15], [16]. When mounted on building facades, WiFi sensors enable UAV operators to track drones in real time, covering blind spots through direct communication. These emerging solutions support the replacement of human observers in EVLoS operations and help bridge the transition toward fully autonomous BVLoS missions.

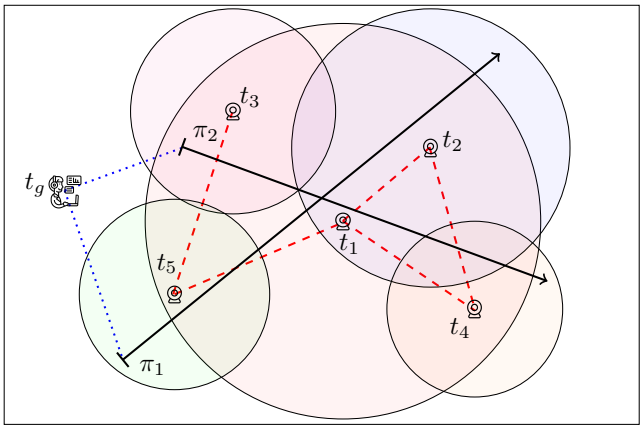


Fig. 1. Example of our scenario. There are two trajectories (π_1 and π_2, solid black lines), five observers (t_1, t_2, t_3, t_4, t_5), and the observer GCS (t_g). The coverage of each observer is the disk centered at the observers, and the connections among observers are shown in red dashed lines.

III. Problem Formulation

A. System Model

Consider a 2D area composed of a set of predefined straight *trajectories*, each assigned to a UAV. Each trajectory defines a corridor within which the UAV can safely fly. The area (see Fig. 1) features a ground communication network of city cameras, referred to as observers, which ensures coverage and connectivity between the UAV and the remote pilot, referred to as the Ground Control Station (GCS). If an observer's coverage area intersects a UAV's trajectory, we refer to it as a *coverage observer*, meaning a camera capable of monitoring a portion of the UAV's flight. Our goal is to leverage the existing ground communication network to enable bidirectional communication between the UAVs, the coverage observers, and the GCS, while minimizing *latency*. All points along the trajectories must be fully covered by the observers. The meaning of latency will be clarified in the following.

Let π_j denote the trajectory of the j-th UAV, which is a line segment representing its flight path. The *starting location* of the trajectory is σ_j, while its *destination location* is δ_j. Given l trajectories, each associated with a starting location and a destination location, we define the following sets: $\mathcal{S} = \{\sigma_1, \ldots, \sigma_l\}$ to be the set of starting locations, $\mathcal{D} = \{\delta_1, \ldots, \delta_l\}$ to be the set of destination locations, and $\Pi = \{\pi_1, \ldots, \pi_l\}$ to be the set of trajectories. The *length* of a trajectory $\pi_j \in \Pi$ is given by $L_{\pi_j} = \|\sigma_j - \delta_j\|_2$, where σ_j (resp., δ_j) corresponds to the starting (resp., ending) point of π_j and its position is taken as the reference position 0 (resp., L_{π_j}). The area also includes a set of n observers, denoted as $\mathcal{T} = \{t_1, \ldots, t_n\}$, to maintain connectivity for every point $x \in \pi_j$, with $0 \leq x \leq L_{\pi_j}$.

The communication range of each observer t_i is represented by a circle of radius r_i centered at (x_i, y_i). The intersection of this circle with the trajectory π_j defines the *interval* $I_j^i = [s_j^i, e_j^i]$, where s_j^i and e_j^i are the endpoints of I_j^i on π_j. Notice that, w.l.o.g. every trajectory is modeled as a straight line segment because it runs in a corridor authorized by NAA. The interval I_j^i represents the portion of the UAV's

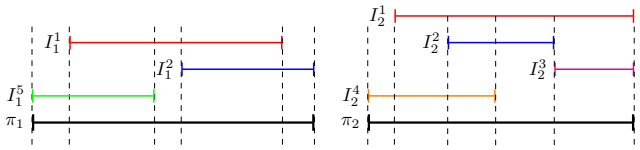


Fig. 2. Example with two trajectories and five observers.

trajectory π_j covered by the observer t_i. With slight abuse of notation, we say that t_i (or I_j^i) covers a portion of π_j. Notably, $0 \leq s_j^i \leq e_j^i \leq L_{\pi_j}$. Additionally, if the starting point σ_j is covered by the observer t_i, then $s_j^i = 0$; conversely, if the destination point δ_j is covered by t_i, then $e_j^i = L_{\pi_j}$. If there is no intersection between t_i and π_j, the interval I_j^i does not exist. The subset of observers $\mathcal{T}' \subseteq \mathcal{T}$ covers π_j if every point of the trajectory is covered by at least one observer in $\mathcal{T}'$.

The ground communication network is modeled as an undirected graph $G = (V, E)$, where V represents the set of vertices and E denotes the set of edges. The vertex set V consists of n vertices corresponding to the observers $\mathcal{T}$ in the city, plus one vertex where the GCS resides. Such observer vertex is denoted as t_g. We assume that all vertices are reachable directly/indirectly from the GCS. So, the ground communication network is connected. The edge set E is defined as follows. For any $t_i, t_j \in \mathcal{T}$, there is an edge $(t_i, t_j) \in E$ if the two observers communicate directly. Let $m = |E|$ and we assume that the edges are given while the observers remain static. Two observers $t_i, t_j \in \mathcal{T}$ are considered connected if there is a path between them in the graph G. We assume that each edge $(u, v) \in E$ is assigned a weight $w(u, v)$, representing the time required to transfer a fixed amount of data from u to v using the given technology. Additionally, let $\Delta(v, u)$ denote the length of the shortest path, measured in the sum of the transmission times along the path between the two nodes $v, u \in V$. Given a set of observers $\mathcal{T}' \subseteq \mathcal{T}$, we define finally the **latency** of $\mathcal{T}'$ from t_g as:

$$\varepsilon(t_g, \mathcal{T}') = \max_{u \in \mathcal{T}'} \Delta(t_g, u). \tag{1}$$

When $l > 1$, let $\mathcal{I}_j$ denote the set of observers that intersect a trajectory $\pi_j \in \Pi$. For an observer $t_i \in \mathcal{T}$, let $\mathbb{I}_i$ be the set of intervals on trajectories Π covered by t_i. Thus, we denote $\mathbf{I} = \{\mathbb{I}_1, \ldots, \mathbb{I}_n\}$ as the collection of subsets of intervals corresponding to the observers in $\mathcal{T}$.

Fig. 2 illustrates the intervals on two trajectories $\Pi = \{\pi_1, \pi_2\}$, and five observers $\mathcal{T} = \{t_1, t_2, t_3, t_4, t_5\}$. The observers that intersect π_1 are $\mathcal{I}_1 = \{t_1, t_2, t_5\}$. For observer t_1, the set $\mathbb{I}_1 = \{I_1^1, I_2^1\}$ consists of the intervals depicted in red. Therefore, $\mathbb{I}_2 = \{I_1^2, I_2^2\}$, $\mathbb{I}_3 = \{I_2^3\}$, $\mathbb{I}_4 = \{I_2^4\}$, and $\mathbb{I}_5 = \{I_1^5\}$. Moreover, $\mathbf{I} = \cup_{z=1}^{5} \mathbb{I}_z$.

B. Problem Definition

In this section, we formulate an optimization problem that aims to ensure bidirectional communication between every point in the set Π of trajectories and the GCS, while minimizing both the number of required observers and the communication latency. Precisely, the objective of the *Minimum Latency Problem* (MLP), is to find a subset of observers $\mathcal{T}^* \subseteq \mathcal{T}$ such that: i) every $\pi_i \in \Pi$ is fully covered by $\mathcal{T}^*$; ii) the latency $\varepsilon(t_g, \mathcal{T}^*)$ is minimum, i.e., for any other subset $\mathcal{T}' \subseteq \mathcal{T}$ that covers Π it holds that $\varepsilon(t_g, \mathcal{T}^*) \leq \varepsilon(t_g, \mathcal{T}')$; and iii) the size of $\mathcal{T}^*$ is minimum.

Problem 1. *Minimum Latency Problem (MLP)*

***Input**: A connectivity graph of the ground units $G = (V, E)$, a set of l trajectories $\Pi = \{\pi_1, \ldots, \pi_l\}$, a set of observers $\mathcal{T} = \{t_1, \ldots, t_n\}$.*

***Output**: Find the set $\mathcal{T}^*$ s.t. $\mathcal{T}^*$ covers Π, $\varepsilon(t_g, \mathcal{T}^*)$ is minimum, and the cardinality of the set $\mathcal{T}^*$ is minimum.*

We start by showing that MLP is *NP*-hard.

Theorem 1. *MLP is NP-hard.*

The idea for solving MLP is to sort the nodes of the ground communication network based on their distance from the GCS, i.e., t_g. Then, one ground node is added to the solution at a time until the current subset includes a set of observers $\mathcal{T}'$ that covers all the trajectories. Noting that this set may contain redundant observers, we compute the subset $\mathcal{T}^* \subseteq \mathcal{T}'$ of minimum cardinality that still covers the entire set Π.

To better illustrate the concept of redundancy, consider the example shown in Fig. 3. There are five observers whose distances from the GCS t_g are given by: $\Delta(t_g, t_1) = 3$, $\Delta(t_g, t_2) = 1$, $\Delta(t_g, t_3) = 2$, $\Delta(t_g, t_4) = 4$, and $\Delta(t_g, t_5) = 5$. The two trajectories are covered for the first time when the observers $\{t_1, t_2, t_3, t_4\}$ are added to the solution. However, this set is not minimal with respect to covering Π within a latency of 4. In fact, the optimal solution is $\mathcal{T}^* = \{t_1, t_4\}$.

Given the minimum latency threshold τ (in our example, $\tau = 4$), the optimal set $\mathcal{T}^*$ is computed by solving an instance of the Set Cover Problem (SCP), as follows. Each trajectory π_j is partitioned into consecutive intervals, called sub-trajectories, defined by projecting the endpoints of all intervals in $\mathcal{I}_j$ onto π_j, along with its start and end points σ_j and δ_j. The set of all such sub-trajectories is denoted by Γ. Each observer t_j whose distance from t_g is at most τ covers some sub-trajectories. Therefore, computing $\mathcal{T}^*$ is equivalent to selecting the smallest set of intervals (i.e., observers) that collectively cover all elements of Γ. For example, in Fig. 3, we have $\Gamma = \{\pi_1^1, \pi_2^1, \pi_2^2\}$. Among the observers $\{t_1, t_2, t_3, t_4\}$, the minimum set that covers Γ is $\mathbf{I} = \{\mathbb{I}_1, \mathbb{I}_4\}$ where $\mathbb{I}_4 = \{\pi_1^1\}$ and $\mathbb{I}_1 = \{\pi_2^1, \pi_2^2\}$, corresponding to t_4 and t_1, respectively. Note that although it is also possible to cover Π using only t_5, the latency between the GCS and t_5 is 5, which exceeds the minimum latency τ required to cover Π.

C. Proposed Algorithm

In this section, we propose an approximation algorithm, `Alg-LSC`, for solving MLP, illustrated in Algorithm 1.

`Alg-LSC` first computes the minimum-latency τ from the GCS that allows to find a feasible coverage of all the trajectories. This can be done in polynomial time. Specifically, it invokes the Dijkstra's algorithm to sort the observers based on their distance from the GCS. Then, the value of τ is determined by progressively increasing the distance from the

Algorithm 1: `Alg-LSC` Algorithm

Data: Connection graph $G = (V, E)$, set of trajectories Π
Result: Approximated subset $\mathcal{T}'$ of the optimal set of observers $\mathcal{T}^* \subseteq \mathcal{T}$ that covers Π with min latency τ

Run Dijkstra's alg. to find the distance of all observers from t_g and consider the observers $\mathcal{T}$ in increasing order based on their distances from t_g
$i \leftarrow 1$, found $\leftarrow$ false
while ***not*** *found* **do**
  $\mathcal{T}'_i \leftarrow$ use the i vertices closest to the GCS t_g returned by Dijkstra's
  Let τ_i be the maximum distance used in $\mathcal{T}'_i$
  Using $\mathcal{T}'_i$, create the collection $\mathbf{I} \leftarrow \cup_{t_j \in \mathcal{T}'_i} \mathbb{I}_j$
  if $\mathbf{I}$ *covers* Π **then**
    $\tau \leftarrow \tau_i$, found $\leftarrow$ true
  else
    $i \leftarrow i + 1$
$\mathcal{T} \leftarrow \mathcal{T}'_i$, $\mathbf{I} \leftarrow \cup_{t_j \in \mathcal{T}} \mathbb{I}_j$, $\tau \leftarrow \tau_i$
return $\mathcal{T}' \subseteq \mathcal{T}$ that covers Π obtained by applying the approximation algorithm for SCP proposed in [17] and the minimum latency τ

GCS until a feasible coverage of all trajectories is achieved using the vertices within that distance. The distance at which this occurs defines the minimum latency τ. Fixed $\tau > 0$, we can model the problem as an instance $(\Gamma, \mathbf{I})$ of SCP, where Γ represents the universe set (i.e., the set of sub trajectories) and $\mathbf{I}$ denotes the observer segments at distance at most τ from GCS. Finally, the `Alg-LSC` algorithm employs a sub-optimal approach to solve SCP proposed in [17].

Since each observer can intersect at most each trajectory, the size of the universe Γ is upper bounded by $\mathcal{O}(nl)$ and its construction takes $\mathcal{O}(nl \log(nl))$ time. Consequently, the greedy approximation algorithm for SCP in [17] returns a $\mathcal{O}\big(\log(nl)\big)$ approximation factor and runs in $\mathcal{O}\big(n^2 l\big)$ time. From the above discussion, we state the following:

Theorem 2. *`Alg-LSC` returns an approximate solution to MLP that achieves optimal latency, while the solution size of the minimum set of observers $\mathcal{T}^*$ is approximated within a factor of $\mathcal{O}\big(\log(nl)\big)$, where l is the number of trajectories and n the number of observers. The algorithm runs in $\mathcal{O}\big(n^2 l \log(nl)\big)$ time.*

IV. Performance Evaluation

This section begins by outlining the experimental setup, followed by the presentation of the numerical results.

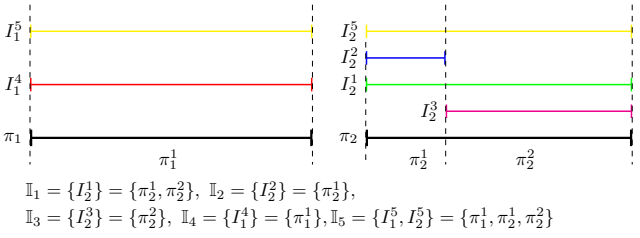


Fig. 3. Two trajectories and five observers. Let $\Delta(t_g, t_1) = 3$, $\Delta(t_g, t_2) = 1$, $\Delta(t_g, t_3) = 2$, $\Delta(t_g, t_1) = 4$, $\Delta(t_g, t_4) = 4$, and $\Delta(t_g, t_5) = 5$.

A. Settings

This section introduces the proposed setting, including the ground communication network layouts and the wireless technologies that determine edge weights.

a) Network Layouts: We consider a squared area of different sizes, and then we build the connection graph $G = (V, E)$ between a set of observers $\mathcal{T}$. However, to evaluate how the performance can vary depending on the graph, we decided to build different layouts, from completely random structures such as RGGs, to more regular structures such as grids. Below, we summarize the considered layouts:

- *RGG with fixed radius*: vertices (set V) are randomly placed, and edges E between any two vertices are established if their distance is at most r, i.e., the fixed communication radius. If the RGG is disconnected, the process is repeated until a connected graph G is obtained.
- *Regular Manhattan grid*: The area is partitioned into a $\gamma \times \gamma$ grid, so in γ^2 vertices (set V). These vertices are arranged to ensure full area coverage, with each internal observer having a connectivity radius that allows exactly four adjacent connections (above, below, left, and right).

Note that the vertices (set V) of the ground network do not act always as observers. Indeed, a node is candidate to act as observer only if its communication range intersects with a trajectory. So, depending on the scenario, a vertex can act as a relay node that forwards the communications or as an observer or both. Among the considered structures, RGG-based layouts are particularly suited for scenarios where the UAV must traverse an unknown or irregular area, such as a battlefield. In these cases, vertex locations can be arbitrary, and the inherent randomness of the RGG model makes it well-suited for capturing such unpredictable environments. In contrast, regular grid layouts, are more appropriate for urban settings, where vertices can be systematically placed at city blocks to ensure uniform coverage. These structured grids align with the organized layout of city streets, enabling efficient and predictable communication pathways between nodes.

b) Wireless Technologies: We consider two wireless technologies, the standard WiFi 5 and an Ad hoc WiFi [18] for data transfer within the network. We refer to WiFi 5 and the Ad hoc network as η_0 and η_1, respectively. The specific parameters of each technology are given in Tab. I (first row), where f_0 is the reference frequency, P_T is the transmitted power, $\mathbb{N}$ represents the noise power, and $\mathbb{B}$ denotes the bandwidth of the channel.

For every edge $(u, v) \in E$, we assume different Data Transfer Rate (DTR) that decreases as the distance between u and v increases, leading to longer data transfer times [19]. To compute the DTR for edge (u, v), we use Friis equation $P_R = P_T - 20 \log_{10}((4\pi d)/\lambda)$, where P_R is the power at the receiver, λ is the wavelength, and d represents the Euclidean distance between u and v. Note that by choosing different technology, P_R varies for the edge (u, v). We then convert this value to Watts using the formula $P_R \equiv 10^{\frac{P_R}{10}}$. Finally, the Shannon–Hartley formula is used to compute the theoretical

TABLE I
PARAMETERS FOR WIRELESS TECHNOLOGIES, RESOLUTIONS, AND STRUCTURES.

Wireless Tech.	Name	f_0 [GHz]	P_T [dBm]	$\mathbb{N}$ [W]	$\mathbb{B}$ [MHz]
Ad hoc	η_0	5	17	1×10^{-9}	40
WiFi 5	η_1	5	20	1×10^{-9}	80

Name	Image Width	Image Height	Channels
ρ_0	640	640	3
ρ_1	1280	720	3

Structure	Name	Area [km^2]	Vertices	Radius [m]	Technology
RGG	RGG_S	1	20	500	Ad hoc
	RGG_M	4	30	500	Ad hoc
	RGG_L	9	50	500	Ad hoc
Grids	Grid_S	1	100	100	WiFi 5
	Grid_M	4	16	500	Ad hoc
	Grid_L	9	36	500	Ad hoc

maximum DTR $\mathbb{D}$ (in bps) for data transmission between the observers, i.e., $\mathbb{D} = \mathbb{B}\log_2(1 + P_R/\mathbb{N})$. Given a data size s in bits, the data transfer time t between observers u and v can be computed as $t = \frac{s}{\mathbb{D}}$. It is worth mentioning that Ad hoc WiFi technology allows data transfer over long distances but at a slower speed. In contrast, WiFi 5 is limited to short distances but offers faster data transfer. For the data, we consider RGB images with two resolutions, as shown in Tab. I (second row), of 640×640 and 1280×720 pixels. We refer to the first and the last one as ρ_0 and ρ_1, respectively.

Finally, Tab. I (third row) summarizes the parameters used for each proposed structure. The subscripts S, M, and L, refer to "small", "medium", and "large areas", respectively. Note that increasing the area, the number of vertices increases, except for the Grid_S. Indeed, this represents a very dense urban area where the communication range is small (i.e., $100\,\text{m}$). This scenario can model the center of city, like Rome or Milan in Italy, where the ground nodes are connected surveillance cameras deployed by the municipality.

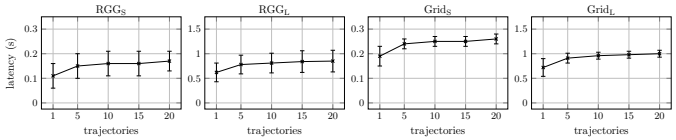
Once the connection graph is constructed over the area, a set of one or more trajectories, denoted as $\Pi = \{\pi_1, \ldots, \pi_l\}$, is randomly generated along with corresponding sets of starting points, $\mathcal{S} = \{\sigma_1, \ldots, \sigma_n\}$, and destinations, $\mathcal{D} = \{\delta_1, \ldots, \delta_l\}$. Specifically, we generate from 1 up to 20 trajectories. Each trajectory is represented as a line fully contained within the area and is randomly designed to have a minimum length of at least $\frac{2}{3}$ of the area side. If a generated trajectory is not entirely covered by the observers, it is discarded and re-generated until it meets the coverage requirement. In other words, if all the observers are considered, the feasibility SCP check is passed.

B. Numerical Analysis

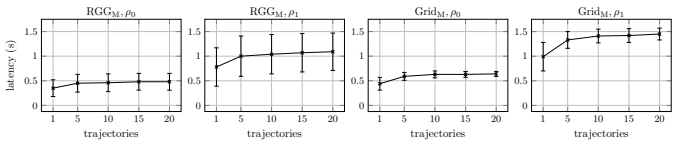
We evaluate the performance of solving MLP on randomly generated data, comparing the optimal `Opt` and the `Alg-LSC` solution, where `Opt` invokes an ILP solver when solving the SCP instance in Algorithm 1. The used ILP solver is Gurobi 12.0. Hence, `Opt` and `Alg-LSC` only differ for Line 12 in Algorithm 1.

We solved MLP on 33 randomly generated instances, computing the average results along with their standard deviation.

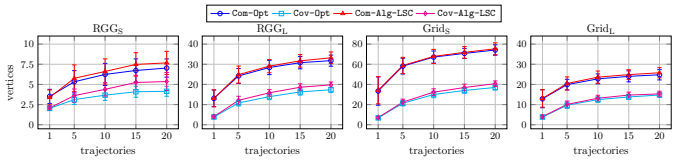
The results are structured as follows. We analyze first the minimum latency τ, common to both algorithms, and then we compare the number of coverage observers required to fully cover all trajectories across both network layouts.



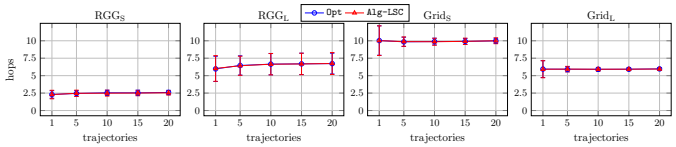
(a) Latency.



(b) Impact of resolution change on latency.



(c) Coverage and communication.



(d) Average hops.

a) Overall Latency: First, we show the results in terms of latency, i.e., the maximum time required to send an image from the farthest observer used in coverage, in Fig. 4a. As previously discussed, the latency remains identical for both algorithms because it is solved by invoking the Dijkstra's algorithm and it does not depend on the SCP. In our enhanced EVLoS, low latency is crucial for timely failure detection and response. Thus, latency serves as key indicator in assessing the actual applicability of our proposed solution in real-world scenarios. This is the most important performance to study the feasibility of our proposed solution.

On the left of Fig. 4a, we consider RGGs, while on the right, we consider grids, and we only consider the smallest resolution ρ_0. According to our network parameters, transferring an image takes no more than $0.25\,\text{s}$ in small layouts ($1\,\text{km}^2$) and no more than $1\,\text{s}$ in large layouts ($3\,\text{km}^2$). Our results show that the latency is quite short. In addition, our method avoids the typical overhead in EVLoS where human observers must relocate themselves, synchronize, before being active. Our results show that the time is suitable to take actions in case of failures. We believe that the round trip from the observer to the remote pilot, and vice versa, will be suitable for the guaranteeing safeness given that the thresholds currently set by the Canadian and Italian Civil Aviation Authorities are on the order of 10-15 s [20], which correspond to at maximum 1-2 m of displacements of the drone.

We also analyze latency in Fig. 4b varying the resolution of the images sent on flights. We compare the two resolutions

ρ_0 and ρ_1 (see Tab. I). The two resolutions are related by a factor of two, which explains why latencies in ρ_1 are approximately double those in ρ_0. So, if more images were sent, such as additional frames that create a short video, we can expect an increase in latency. However, the delay would not linearly increase since pipeline techniques could be employed to enhance throughput.

b) Observers: Secondly, we present the results in terms of the output $\mathcal{T}'$ of Problem 1, i.e., the subset of observers needed to fully cover all trajectories. We investigate both the vertices that act as relay nodes (i.e., the communicating vertices, called `Com`) and the vertices that are actually observers (i.e., the coverage vertices `Cov`). As said, due to the actual connections among the observers, communicating vertices may be required to forward the observer data to the remote pilot t_g. Since the main goal of the observers is to let the remote pilot maintain indirect oversight on the drones, it is important also to count how many communications nodes are involved.

Fig. 4c reports these results. On the left, we consider RGGs, while on the right, we consider grids. In these experiment, we consider the smallest resolution ρ_0. Let us consider the case with 20 trajectories. As expected, due to the communication range of $500\,\mathrm{m}$, in $\mathrm{RGG_S}$, the number of required nodes for coverage and communication is small (around 20% and 35% of the total observers, respectively), and their gap is also limited. This means that in small environments with highly connected networks, there is no need to use too many observers, which benefits latency as well. On the other hand, in larger environments like $\mathrm{RGG_L}$, although there are many observers, the number of required observers for coverage and communication is larger (around 35% and 65% of the total observers, respectively), and their gap also increases. In $\mathrm{Grid_S}$ and $\mathrm{Grid_L}$, similar percentage trends can be observed (approximately 35-40% and 70-75%, respectively). This is because in $\mathrm{Grid_S}$, we have a significantly higher number of observers but a limited communication radius, whereas in $\mathrm{Grid_L}$, we have only a third of the observers but with a larger radius. These results suggest that the two generated graphs are comparable despite the significant differences in area size.

The difference between the outputs of `Alg-LSC` and `Opt` accounts for the approximation made in `Alg-LSC`, and it is greater for RGG than for Grid. Regular displacement of the grid scenario is likely to have additional properties that are more effectively exploited by the greedy SCP. As a final remark, the number of required observers increases as the number of trajectories increases. Moreover, it is worthy to observe that the number of communication vertices is greater than or equal to the number of coverage observers. To explain this, we count average number of hops (i.e., number of used edges) during the flight of UAVs in Fig. 4d. As one can easily check, when the number of average hops is high, also the number of communication vertices grows. Nonetheless, the number of communication vertices is not as high as the number of coverage nodes multiplied by the number of hops because not all the observers work at the same time.

V. Conclusion

We proposed a system to enhance EVLoS UAV operations toward BVLoS by leveraging existing city cameras and communication networks. We introduced the MLP, an NP-hard problem which minimizes the number of ground antennas while keeping low communication latency. Simulations confirmed the feasibility and efficiency of our approach.

Future work includes real-world deployment, use of diverse communication technologies, improved antenna placement algorithms, and privacy-aware, dynamic camera selection strategies.

References

[1] C. Janke and M. U. de Haag, "Implementation of european drone regulations-status quo and assessment," *Journal of Intelligent & Robotic Systems*, vol. 106, no. 2, p. 33, 2022.

[2] R. J. Wallace *et al.*, "Evaluating methods of faa regulatory compliance for educational use of unmanned aircraft systems (uas)," *The Collegiate Aviation Review International*, vol. 35, no. 1, 2017.

[3] K. A. Butler, "The perception of pilots and support personnel in aviation/aerospace industry regarding future implementation of uas," Arizona State University, Tech. Rep., 2024.

[4] O. Anicho, A. K. Nagar, and J. C. Bansal, "Considerations for unmanned aerial system (uas) beyond visual line of sight (bvlos) operations," *Drones and Autonomous Vehicles*, vol. 1, no. 4, p. 10010, 2024.

[5] S. Bertrand *et al.*, "Handling ground risks for road networks in uas specific operations risk assessment (sora)," in *2024 International Conference on Unmanned Aircraft Systems*. IEEE, 2024, pp. 850–857.

[6] N. Rymer, A. Moore, S. Young, L. Glaab, K. Smalling, and M. Consiglio, "Demonstration of two extended visual line of sight methods for urban uav operations," in *AIAA AVIATION 2020 FORUM*, 2020, p. 2889.

[7] F. Betti Sorbelli, P. Chatterjee, P. Das, and C. M. Pinotti, "Risk assessment in bvlos operations for uavs: Challenges and solutions," in *20th Intl. Conf. on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT)*. IEEE, 2024, pp. 300–307.

[8] F. Betti Sorbelli, P. Chatterjee, F. Corò, S. Ghobadi, L. Palazzetti, and C. M. Pinotti, "A novel graph-based multi-layer framework for managing drone bvlos operations," *IEEE Trans. on Network and Service Management*, pp. 1–1, 2024.

[9] F. Betti Sorbelli, P. Chatterjee, F. Corò, S. Ghobadi, and C. M. Pinotti, "Scheduling of multiple uavs in bvlos operations along unidirectional and bidirectional paths," in *2024 IEEE 49th Conference on Local Computer Networks (LCN)*. IEEE, 2024, pp. 1–7.

[10] A. Zmarz *et al.*, "Bvlos uav missions for vegetation mapping in maritime antarctic," *Frontiers in Env. Science*, vol. 11, p. 1154115, 2023.

[11] R. J. a. L. Hartley *et al.*, "Bvlos unmanned aircraft operations in forest environments," *Drones*, vol. 6, no. 7, p. 167, 2022.

[12] "5g demonstrative apps - city camera security and surveillance system," https://commission.europa.eu/node/30386, acc: 2025-02-16.

[13] "5gcity," https://cordis.europa.eu/project/id/761508, acc: 2025-02-16.

[14] "Connectow demo day: A 5g for smart communities project," https://digital-strategy.ec.europa.eu/en/news/5g-connectivity-wavre-connectow, acc: 2025-02-16.

[15] J. Li, A. Sharma, D. Mishra, J. G. Davis, and A. Seneviratne, "Wifi sensing for outdoor surveillance," in *2023 57th Asilomar Conference on Signals, Systems, and Computers*, 2023, pp. 1713–1718.

[16] A. Ghosh and A. Chouksey, "Wifi for privacy-friendly surveillance," in *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, 2024, pp. 1–9.

[17] V. V. Vazirani, *Approximation algorithms*. Springer, 2001.

[18] TP-LINK, "Cpe510," https://www.tp-link.com/it/business-networking/pharos-cpe/cpe510/, acc: 2025-02-16.

[19] F. Betti Sorbelli, S. Ghobadi, and C. M. Pinotti, "Single-and multi-depot optimization for uav-based iot data collection in neighborhoods," *ACM Trans. on Sensor Networks*, vol. 21, no. 1, pp. 1–30, 2025.

[20] "Jarus guidelines on specific operations risk assessment (sora)," http://jarus-rpas.org/wp-content/uploads/2024/06/SORA-v2.5-Main-Body-Release-JAR_doc_25.pdf, acc: 2025-02-27.