A Novel Graph-Based Multi-Layer Framework for Managing Drone BVLoS Operations

Francesco Betti Sorbelli, *Member, IEEE*, Punyasha Chatterjee, *Senior Member, IEEE*, Federico Corò, *Member, IEEE*, Sajjad Ghobadi, Lorenzo Palazzetti, *Member, IEEE*, and Cristina M. Pinotti, *Senior Member, IEEE*

***Abstract*—Drones have become increasingly popular in a variety of fields, including agriculture, emergency response, and package delivery. However, most drone operations are currently limited to within Visual Line of Sight (VLoS) due to safety concerns. Flying drones Beyond Visual Line of Sight (BVLoS) broadens to new challenges and opportunities, but also requires new technologies and regulatory frameworks to ensure that the drone is constantly under the control of a remote operator. In this work, we propose a novel graph-based multi-layer framework that closely resembles real-world scenarios and challenges in order to plan drone BVLoS operations. Our framework includes layers of constraints such as ground risk, cellular network infrastructure, and obstacles, at different heights. From the multi-layer structure, a graph is constructed whose edges are weighted with a dependability score that takes into account the information of the layers, allowing efficient path planning of BVLoS missions, using algorithms such as Dijkstra's. Since the built graph can be really large, we also propose lighter graph-based corridors by considering only a limited portion of the original graph. Through extensive experimental evaluation on a real dataset, we demonstrate the effectiveness of our framework in solving the *Maximum Dependability Path Problem* (MDP2), which can be efficiently solved by applying the Dijkstra's algorithm.**

***Index Terms*—Drones, BVLoS, Connectivity, Ground risk**

I. Introduction

Nowadays, drones, also known as Unmanned Aerial Vehicles (UAVs), or more broadly, Unmanned Aerial System (UAS), are being used in a growing number of applications [1], [2] and play a significant role in the promotion of advanced air mobility, which involves the use of small automated aircraft in urban and suburban areas at low altitudes [3]. Previously, only ground vehicles and manned aerial vehicles have been utilized in these contexts. However, ground vehicles face limitations due to terrain and congested infrastructure, while the use of planes is prohibitively expensive for most people. Therefore, the use of UAS is becoming an attractive solution in various applications, also because they do not emit greenhouse gases.

Manuscript received Month xx, yyyy; revised Month xx, yyyy; accepted Month xx, yyyy. Date of publication Month xx, yyyy; date of current version Month xx, yyyy. (*Corresponding author: F. Betti Sorbelli.*)

This work was supported by the "BREADCRUMBS" project funded by the PRIN 2022 PNRR under grant no. P2022K7ERB. F. Betti Sorbelli, S. Ghobadi, and C. M. Pinotti are with the Dept of Computer Science and Mathematics, Univ. of Perugia, Italy; P. Chatterjee is with the School of Mobile Computing and Communication, Jadavpur Univ., India; F. Corò is with the Dept of Mathematics, Univ. of Padua, Italy; L. Palazzetti is with the Dept of Computer Science and Mathematics, Univ. of Florence, Italy.

F. Betti Sorbelli, F. Corò, S. Ghobadi, L. Palazzetti , and C. M. Pinotti are member of the Gruppo Nazionale Calcolo Scientifico-Istituto Nazionale di Alta Matematica (GNCS-INdAM).

Digital Object Identifier XXXXXXXX/ZZZ.YYYYYYYY

Urban environments are critical, and introducing UAS in cities poses to significant challenges, including concerns about public safety, cyber-security, and privacy. Specifically, urban areas, characterized by dense populations, increase the risk of UAS crashes, potentially involving people and, in the most severe scenarios, causing casualties. Typically, flying over people is allowed only under specific conditions to ensure an adequate level of safety. The categorization of UAS operations is based on their level of risk [4]. Operations that pose a low risk do not require prior authorization, whereas those that present a higher risk must obtain special authorization before proceeding. Currently, any operation that requires the drone to fly Beyond Visual Line of Sight (BVLoS) requires special authorization. To enable BVLoS missions, there are some methods available, such as a Notice to Airmen (NOTAM) authorization, Extended Visual Line of Sight (EVLoS) flights [5], or the exploitation of "corridors" [6], [7]. NOTAMs restrict a specific area, reducing the risk because only authorized UAV operations can use the area, and ground facilities are aware of the presence of the drone. An EVLoS flight simulates BVLoS operations by linking different VLoS operators in series. Although for different reasons, these methods are considered inflexible, complex, and impractical for enabling daily BVLoS flights in the near future.

Fig. 1. Our envisioned scenario involves an operator or a central unit piloting UAVs in BVLoS. The operator has limited visibility and follows the drone using the cellular infrastructure. The drone's path is selected by prioritizing robust wireless connections and ground safety.

To pursue reliable BVLoS flights, there are two great challenges regarding the *ground risk* and the *drone connectivity*. The *ground risk* is a measure of the risk in case of some drone malfunctioning. It depends on the concentration of people over areas, and on the density and the type of the ground buildings, tall trees, towers, etc [8]. In this paper, we assume that the drone flies below the legal height limit of $120\,\mathrm{m}$ ($400\,\mathrm{ft}$) [5], which reduces the risk of coming across other aircraft, which normally fly much higher. The presence of buildings, towers, and trees can slightly mitigate the risk for the people since they can act as shelters [9], but they also represent obstacles during the flight. A previous risk assessment helps UAVs to limit the risks of the flight.

By *drone connectivity* is meant that a UAV must communicate its position anytime so that its route can be monitored by some ground central unit (or operator), who can possibly modify the route by providing new tasks [10]. Connectivity is indeed the way to extend the drone operation with the same level of dependability guaranteed by the VLoS operations. Since thinking of deploying an ad hoc infrastructure for ground-UAV and UAV-UAV communications has a high cost and is not a sustainable solution, it is realistic to rely on already available ones, like those used for connecting ground users (cellular). There are many generations of cellular networks already deployed (e.g., 4/5G), characterized by different wavelengths, that can be used for BVLoS links [11], [12].

In order to enable BVLoS flights (see the scenario envisioned in Figure 1), we propose a novel *graph-based multi-layer framework* that considers at the same time the ground risk and the drone connectivity. The framework uses information from various layers such as no-fly zones, obstacles, and from the cellular infrastructures with the aim of planning BVLoS missions. We weight the edges of the graph with a dependability score that reflects the information extracted by the layers of the framework. Our ultimate goal is to plan a BVLoS mission from a source to a destination in the graph by following the path with the highest dependability.

A preliminary version of this paper is detailed in [13] that only contained the framework and graph model. From that, we enhanced the framework including the handover-success-rate probability function (detailed in Section IV-A). We also present a new lightweight graph-based corridor model. To find the path with the maximum dependability, we apply Dijkstra's algorithm on both the graph- and the corridor-models. Our contributions are summarized as follows:

- We revisit and extend a *graph-based multi-layer framework* that integrates crucial information from diverse layers, encompassing obstacles, ground risks, and wireless communication infrastructure, at varying altitudes.
- From this framework, we build a graph whose edges are weighted based on a dependability score derived from the layers, on which we optimally solve the *Maximum Dependability Path Problem* (MDP2).
- We also propose a lightweight yet sub-optimal solution that involves generating smaller graph-based corridors by focusing on specific segments of the original graph.
- We evaluate the effectiveness of our framework in solving the MDP2 by assessing various metrics, including the number of traversed vertices, handovers, height changes, and path dependability. Our experiments use a real dataset for the tower distribution.
- We compare the dependability performance of the graph- and the corridor-solutions.

The rest of the paper is organized as follows. Section II reviews the related work. Section III presents the model and multiple layers. Section IV and V introduce the graph and corridor structure, respectively. Section VI evaluates the performance of our framework. Section VII offers conclusions.

II. Related Work

This section covers existing works on risk-aware path planning and communications for UAVs flying BVLoS, but none of the works considers both factors together.

A. Risk Analysis and Path Planning

The use of drones for BVLoS operations presents various risks, such as collisions, communication loss, and equipment failure. To assess the impact of a UAV crash on the population, Primatesta et al. [8] propose a two-dimensional Risk Map that considers factors such as population density, no-fly zones, obstacles, and sheltering factors. Risk values are calculated using a risk assessment process that considers various parameters and uncertainties. The authors suggest using the Risk Map to identify the optimal risk path based on the Rapidly-exploring Random Tree (RRT*) and A* algorithms. It is worth noting that the framework presented in our work includes the Risk Map by Primatesta et al. as one of its components. Indeed, our approach can even plan a path only taking into account the ground risk. However, differently from Primatesta et al., we discretize the map and we use Dijkstra's algorithm to identify the optimal risk path. Our work extends Primatesta's work extending the framework to address the connectivity issue. The same author proposes in [14] a path planning strategy based on a variant of the RRT* algorithm, performing a risk assessment during the path planning phase. Unlike in the previous work of the authors [8], where the RRT* algorithm is used to minimize risk costs of a risk-based map, in the proposed strategy the risk assessment is performed during the path planning phase. Specifically, each time a new node is added to the exploration graph, a risk assessment procedure is done evaluating the flight direction and velocity, and estimating a probabilistic impact area. The probabilistic risk assessment approach takes into account the drone parameters and environmental characteristics.

In [15], authors present an approach for simplified path planning for UAVs in obstacle-dense high-risk areas. They reduce the complexity of the 3D planning problem to that of a 2D planning problem by leveraging regulatory restrictions and guidelines as well as mission-specific boundary conditions. They further suggest a multi-layered motion planning architecture, where each of the planners is tailored to a specific flight phase and displays deterministic behavior in memory and runtime complexity. However, the proposed 2D reduction relies on the assumption of a steady flight altitude for the drone. Moreover, the authors completely neglect both the communications requirements and the mandatory restriction of having a constant and reliable link between the drone and a ground base, e.g., a cell tower, to allow the BVLoS flight. In our proposed solution, the drone may change its altitude along the flight, and the same must adapt its path in order to have always reliable communication with a ground antenna to ensure connectivity during the flight.

In [16], authors survey the risk assessment for UAVs based on two UAS logistic delivery case studies. To examine the expected level of safety to ground risk and air risk, the case studies result in acceptable data to support the UAS logistic delivery with adequate path planning in the remote and suburban areas in Taiwan. Google routing is used for path planning to keep away from highly populated areas. Differently from our studies, the authors in [16] claim ideal communication between the drones and operators regardless of their relative position, and the potential presence of obstacles.

The authors in [17] present a path planning algorithm that accommodates real-time traffic and geo-fence constraints in low-altitude BVLoS airspace, by integrating RRT techniques with Detect and Avoid Alerting Logic. The proposed architecture differs from ours, firstly, because it does not assess the ground risk for path planning. Secondly, due to throughput requirements, they avoid storing the complete airspace using a reduced one based on a tree data structure. As a consequence, their devised algorithm cannot provide a guaranteed quality bound for the flight dependability. The authors in [18] suggest a strategy to assess the capacity of UAV corridors by relating the collision rate of the corridor and the failure rates of UAVs to the number of ground fatalities, while the ones in [19] present an optimization model for drone navigation that minimizes both the maximum threat level and the flight path length. In both the papers the main difference with respect to our framework is the complete absence of communication handling between the drones and ground stations during the path planning. Moreover, the authors in [18] restrict their studies only to corridors, whereas in [19] they claim fixed and steady drone flight altitude.

B. Cellular Communications and UAVs

One of the main technical challenges of BVLoS operations is maintaining a reliable link between the drone and operator as the drone moves. In particular, it is critical to maintain the connection during the handover, i.e., when the drone moves away from one ground station of the cellular network and switches to a new one. Several studies addressed the handover problem for BVLoS drones. In [20], authors study the performance of cellular-connected UAVs under 3D practical antenna configurations. Their results reveal that vertically-mobile UAVs are susceptible to altitude handover due to consecutive crossings of the nulls and peaks of the antenna side lobes. Diversely from our research, the core of the paper [20] is understanding the coverage variation in the communication network according to several antenna displacements, and drone infrastructures. The authors in [21] propose an approximation of the probability mass function of handover count (HOC) as a function of the UAV's velocity, HOC measurement time window, and ground station densities. However, the authors focus solely on the optimization of the quality of service experienced during the drone flight. Moreover, the paper does not consider the ground risk and assumes a steady flight for the drone at a constant altitude. Furthermore, the authors in [22] propose an experimental study on cell association and handover rates for drones, connected to an LTE-A network in a suburban environment. In our paper, we select the path of the drone including the cost of performing handovers. The contributions of this paper differ from ours because the scope is solely the handover optimization evaluated through a real test-bed. In [23], authors conduct a field trial measurement in an LTE network and suburban environment, at the National University of Malaysia campus. The measurement results show that by increasing flight height from the ground to $170\,\mathrm{m}$ the received signal power and the signal quality levels are reduced by $20\,\mathrm{dBm}$ and $10\,\mathrm{dB}$ respectively. However, the handover still occurred in good condition. By comparing the results of heights $0\,\mathrm{m}$ and $40\,\mathrm{m}$, it is revealed that when the drone flies above the height of buildings and trees it is almost served by nearby eNodeBs [24]. Consequently, the same authors in [25] propose a hardware and software configuration that utilizes the cellular network and Detect and Avoid (DAA) system to achieve a safe BVLoS UAV operation. They fly the UAV at the determined routes at a speed of $18\,\mathrm{km/h}$ at 4 different elevations ($65\,\mathrm{m}$, $85\,\mathrm{m}$, $105\,\mathrm{m}$, and $125\,\mathrm{m}$). They find that at $65\,\mathrm{m}$ height, planning the routes is a bit difficult since they need to avoid high-rise buildings and high trees. On the other side, by increasing the height, the network quality is degraded. Therefore, they foresee that the best altitude to operate the UAV is within the $85\,\mathrm{m}$ height range. Differently from our scope, the results reported by the previous papers are related to the development of a reliable drone jointly with a tracking infrastructure. Therefore, they revise the real performance of the built drone towards an effective BVLoS infrastructure.

In [26], authors set system parameters for the Ultra-reliable low latency communication (URLLC) of 5G focusing on information theoretic approximations for reliability and latency under finite block-length regime for different altitudes ($1.5\,\mathrm{m}$ to $120\,\mathrm{m}$). They find that the minimum distance between UAVs to avoid any crash should be around $0.2\,\mathrm{m}$ for $15\,\mathrm{m/s}$ UAV speed. The paper's main focus lies in conducting experiments to optimize mission parameters for signal reliability and throughput. Unlike our approach, it appears to neglect ground risk and overall mission dependability. Finally, authors in [27] introduce several urban airspace segmentation and present a future perspective of 6G-enabled dynamic UAV traffic management (UTM) ecosystems in 3D space for conflict-free UAV operations. In contrast to our approach, the authors of the paper delve into the technical details of the network segmentation optimization, providing a reasonable solution to tackle current network limitations. Therefore, they investigate BVLoS but only from the infrastructure development point of view.

In summary, the discussed papers serve as valuable test beds, offering insights into the intricate context of UAV communications modeling. These specific case studies highlight the complexity of the subject matter. However, it is crucial to note that these studies do not provide definitive conclusions or take-away lessons, making it challenging to draw any firm assumptions from their findings to be used in our context.

III. System Model and Multi Layers

Let us consider a 3D environment bounded by a *box* denoted by $\mathcal{B}$ characterized by a length $\mathcal{B}_L$, width $\mathcal{B}_W$, and height $\mathcal{B}_H$, that lies on the ground plane, with $\mathcal{B}_L$, $\mathcal{B}_W$, and $\mathcal{B}_H \in \mathbb{R}^+$. To discretize the environment, we divide $\mathcal{B}$ into a number of identical 3D *cells*, each with length, width, and height side ℓ_L, ℓ_W, and ℓ_H, respectively. We denote each cell with its relative position from the origin with a tuple $c = (x_c, y_c, z_c) \in \mathcal{B}$, where x_c, y_c, and $z_c \in \mathbb{N}$ are the x-coordinate, y-coordinate, and z-coordinate, respectively, of c in the discretized environment. The drone position is discretized as well, and without loss of generality, we assume that its actual position is at the center of the cell. In fact, the cell $c = (x_c, y_c, 1) \in \mathcal{B}$ lies at the lowest level at a height $\frac{\ell_H}{2}$ above the ground. The sizes of $\mathcal{B}$ are $n = \frac{\mathcal{B}_L}{\ell_L}$, $m = \frac{\mathcal{B}_W}{\ell_W}$, and $h = \frac{\mathcal{B}_H}{\ell_H}$, respectively, so there are nmh cells. So, $1 \leq x_c \leq n$, $1 \leq y_c \leq m$, and $1 \leq z_c \leq h$. For simplicity, in the rest of the paper, we assume the flying area to be flat. So, this assumption is not a limitation as any environment can be reconstructed using obstacle cells.

According to the drone mobility, the drone can only move to adjacent positions. In fact, given $c = (x_c, y_c, z_c) \in \mathcal{B}$, let $\mathcal{A}(c)$ be the set of *adjacent cells* of c where $\mathcal{A}(c) = \{c' \in \mathcal{B}, c' = (x_{c'}, y_{c'}, z_{c'}) \neq c : x_{c'} = x_c \pm \{0,1\}, y_{c'} = y_c \pm \{0,1\}, z_{c'} = z_c \pm \{0,1\}\}$. Note that c is excluded from $\mathcal{A}(c)$. Therefore, $\max |\mathcal{A}(c)| = 3^3 - 1 = 26$ as c can be located on the boundary of $\mathcal{B}$, reducing the number of available adjacent positions.

Since flying in BVLoS requires some essential information about the surrounding environment in $\mathcal{B}$, we propose a *graph-based multi-layer framework* where each layer contains useful data. Each cell is associated with a geo-referenced location and has three different values, one for each type of layer. The framework is composed of the following layers:

- **No-Fly zone/Obstacle Layer**: models the areas where the drone flight is forbidden/not allowed or possible;
- **Risk-map Layer**: models the risk to people on the ground in the event of a drone crash or malfunction;
- **Wireless infrastructure Layer**: models the feasible wireless communications and their quality.

A. No-Fly zone/Obstacle Layer

The no-fly zone/obstacle layer models the cells where drones cannot fly, such as buildings or trees, or close to airports. This layer is considered at various heights. If there are obstacles, drones can fly over them at a sufficient height. As a rule, the higher the drone flies, the fewer obstacles it will encounter. No-fly zones, on the other hand, are areas where drone flight is prohibited and are effective at any height.

Each cell $c \in \mathcal{B}$ has a *forbidden flight* probability $\mathcal{P}_{\text{FF}}(c)$. $\mathcal{P}_{\text{FF}}$ is a probability rather than a binary value because it reflects the likelihood that a given cell contains an obstacle or falls within a no-fly zone, thus rendering it unsuitable for flights. The amount of obstacles varies based on the type of environment, which can be classified as rural, suburban, or urban, with urban areas typically having a higher frequency, while rural areas tend to have fewer. If a cell $c = (x_c, y_x, z_c)$ contains no-fly zones or obstacles, also in any cell $c' = (x_c, y_c, z_{c'})$, with $1 \leq z_{c'} \leq z_c$, the flight is forbidden.

B. Risk-map Layer

The risk-map layer models the potential risks that unexpected events can cause to ground people. We do not consider air risk, i.e., collisions with other flying vehicles (manned or unmanned). The ground risk-map layer is evaluated depending on a concatenation of different probabilities, and it is defined at various heights. Moreover, it will be computed for the flying area and then given in input to our algorithm.

We refer mainly to [8], [28] for computing the risk. Given a cell $c \in \mathcal{B}$, the risk [8] is so defined:

$$\mathcal{R}(c) = \mathcal{P}_{\text{event}}(c)\ \mathcal{P}_{\text{impact}}(c)\ \mathcal{P}_{\text{fatality}}(c), \tag{1}$$

where $\mathcal{P}_{\text{event}}$, $\mathcal{P}_{\text{impact}}$, and $\mathcal{P}_{\text{fatality}}$ are the probabilities of event, impact, and fatality, respectively, and $0 \leq \mathcal{R}(c) \leq 1$. $\mathcal{P}_{\text{event}}$ is the probability [28] that the drone loses control with the consequent uncontrolled descent with a crash on the ground. It is the frequency to have a ground impact event defined as a rate per hour. Four descent event types are considered: ballistic descent, uncontrolled glide, parachute descent, and fly-away. $\mathcal{P}_{\text{impact}}$ is a 0-1 normalized function [28] that depends on the population area density, and on the size of the drone, where a higher population density, a larger crash area, and a larger drone size result in a higher probability of impact. In particular,

$$\mathcal{P}_{\text{impact}}(c) = \rho(c)\ A_{\text{exp}} \tag{2}$$

where $\rho(c)$ is the population area density, and A_{exp} is the *exposed lethal area*, which is:

$$A_{\text{exp}} = 2(r_{\text{p}} + r_{\text{uav}})\frac{h_{\text{p}}}{\tan(\psi)} + \pi(r_{\text{p}} + r_{\text{uav}})^2 \tag{3}$$

with r_{p} and h_{p} are the average radius and height of a person, respectively, r_{uav} is the radius of the drone, and ψ is the impact angle on the ground. $\mathcal{P}_{\text{fatality}}$, as a result of a drone impact, is a function [28] that depends on both the sheltering factor and the altitude of the drone at the time of impact, and measures the probability of killing people. The sheltering factor $S(c)$ at cell c, which is dependent on the area where the drone is operating, quantifies the level of shelter provided to people by elements such as buildings or trees [29]. An urban area, e.g., has a high value of sheltering factor, which can greatly reduce the kinetic energy at impact and hence the probability of fatalities. Furthermore, a greater drone altitude upon impact leads to a higher likelihood of fatality. Specifically [28],

$$\mathcal{P}_{\text{fatality}}(c) = \frac{1-\kappa}{1 - 2\kappa + \sqrt{\tau/\eta}\ \mathbb{E}} \tag{4}$$

where $\mathbb{E} = (\eta/E_{\text{imp}})^{\frac{3}{S(c)}}$, $\kappa = \min\{1, \mathbb{E}\}$, η is the impact energy needed to cause a fatality when $S(c) \to 0$, τ is the impact energy needed to obtain a fatality probability of 50% when $S(c) = 6$, and $E_{\text{imp}} = \frac{1}{2}\delta v_{\text{imp}}^2$ is the kinetic energy at impact where δ is the drone mass, and $v_{\text{imp}} = \sqrt{v_{\text{init}}^2 2g\zeta}$ is impact velocity, where v_{init} is initial vertical velocity, g is the acceleration due to gravity, and ζ is the height from which the drone is falling which depends on the layer.

So, each cell $c \in \mathcal{B}$ has a *ground-safeness* probability $\mathcal{P}_{\text{GS}}(c)$ which depends on the risk $\mathcal{R}(c)$. Specifically,

$$\mathcal{P}_{\text{GS}}(c) = 1 - \mathcal{R}(c), \tag{5}$$

and $0 \leq \mathcal{P}_{\text{GS}}(c) \leq 1$. Moreover, $\mathcal{P}_{\text{GS}}$ is the frequency rate to have a casualty [28], also called as the probability per hour to have a lethal accident. Parameters are detailed in Section VI.

C. Wireless Infrastructure Layer

The wireless infrastructure layer $\mathcal{W}$ defines the wireless connectivity in the area. In our proposed model, we assume the presence of a pre-existing wireless network infrastructure (e.g., through 4/5G networks) within the environment $\mathcal{B}$, which includes a set of *cellular towers* denoted as $\mathcal{T}$. Each tower $t \in \mathcal{T}$ is characterized by its discretized position in the environment, represented by a tuple $t = (x_t, y_t, z_t) \in \mathcal{B}$, where x_t, y_t, and z_t denote the x-coordinate, y-coordinate, and z-coordinate, respectively, of the tower. To simplify, we assume that all towers are located at the lowest level, with $z_t = 1$, and that there is no more than one tower in a single cell. This simplification can be easily removed if the graph construction is done on computers with large RAM or if the graph is saved to an external memory. In that case, the cell size can be reduced until there is one tower per cell. It becomes evident that the identifier tower t coincides with the cell c where it resides. However, a drone can receive and send its messages to any tower that is up to a certain distance from the drone itself as explained below. The communication quality depends on the *presence of obstacles*, and the *power of signal*.

1) Presence of Obstacles: An aspect to consider is the probability of being on the *Radio* Line of Sight (briefly, LoS) which increases if the drone height increases. Namely, the probability $\mathcal{P}_{\text{LoS}}$ of being LoS increases with the elevation angle between the drone and the tower. Not only the height of the drone but also the presence of obstacles impact $\mathcal{P}_{\text{LoS}}$, whose value decreases if the density of obstacles increases. So, given two parameters α and β [30] that model the environment (rural, suburban, urban) by assuming that the drone is flying at the cell c, the $\mathcal{P}_{\text{LoS}}$ with the tower t is defined as [30]:

$$\mathcal{P}_{\text{LoS}}(c,t) = \frac{1}{1 + \alpha\ e^{-\beta(\phi-\alpha)}}, \tag{6}$$

where $\phi = \arctan\left(\frac{z_c}{d_G}\right)$ is the drone/tower *elevation angle*, with d_G being the Euclidean *ground distance* that separates the two. Hence, by increasing the height z_c and fixing x_c and y_c, a drone can detect more towers due to increased elevation.

2) Power of the Signal: Drones can connect to towers if some physical constraints are met at the same time. We have to distinguish between the *transmitter* device, and the *receiver* device. We adopt the Friis transmission equation [31] (see Eq. (7)) in order to determine the quality of the BVLoS communication link if a drone in cell $c = (x_c, y_c, z_c)$ establishes a connection to a tower $t = (x_t, y_t, 1)$, i.e., by evaluating:

$$P_W(c,t) = P_W(t)\ G(t)\ G(c) \left(\frac{\lambda}{4\pi d_S}\right)^2, \tag{7}$$

where $P_W(c,t)$ is the power at the receiver, $P_W(t)$ is the transmitted power, $G(t)$ and $G(c)$ are the antenna gains of the transmitting and receiving devices, respectively, λ is the wavelength that represents the effective aperture area of the receiving antenna, and $d_S = \|c - t\|_2$ is the drone/tower Euclidean *slant distance*. Then, we *normalize* the value $P_W(c,t) \in [0,1]$ where 1 represents an excellent signal strength received at the drone, and 0 represents no signal/disconnection (details in Section IV-A). At different heights, the distance d_S varies, which in turn affects the power received by the drone. So, the higher the altitude, the longer the distance d_S, the less is the received power $P_W(c,t)$, and the less is the quality of the BVLoS link. So, by increasing the distance between the tower and the drone, the $\mathcal{P}_{\text{LoS}}$ and the power of the signal have an opposite behavior. Although the communication distances change in a continuum way, we represent them using our cell model. First, we model the $\mathcal{P}_{\text{LoS}}$ which increases with the elevation angle. Then, we exploit the signal quality.

We assume that the drone can communicate only with a subset of towers in its neighboring. Let $\gamma \in \mathbb{N}$ be the *visibility factor* at the drone, and for a given cell $c = (x_c, y_c, z_c) \in \mathcal{B}$, let $\mathcal{S}_\gamma(c)$ be the *visibility square area*, i.e., a square which contains a subset of cells of side length $2\gamma \cdot z_c + 1$ centered at cell c. Specifically,

$$\begin{aligned}\mathcal{S}_\gamma(c) = \{q \in \mathcal{B} : x_c - \gamma z_c \le x_q \le x_c + \gamma z_c \text{ and} \\ y_c - \gamma z_c \le y_q \le y_c + \gamma z_c\}.\end{aligned} \tag{8}$$

To determine the visible towers which a drone located in position c can communicate with, the square $\mathcal{S}_\gamma(c)$ is computed using the value of γ. This square is centered around the drone, and any towers within the square can be communicated with. As the height of the drone increases, the size of the square also increases. For instance, when $z_c = 1$, the side length is $2\gamma + 1$, for $z_c = 2$, it becomes $4\gamma + 1$, and so forth.

Now, let $\mathcal{U}_\gamma(c)$ be the subset of towers that the drone can "see" when it is in c, i.e., the towers that belong to the square $\mathcal{S}_\gamma(c)$ centered in c. Precisely,

$$\mathcal{U}_\gamma(c) = \bigcup_{t \in \mathcal{S}_\gamma(c)} t. \tag{9}$$

Under this simplification, we assume that a drone in cell c can communicate with any tower $t \in \mathcal{U}_\gamma(c)$. For any $t \notin \mathcal{U}_\gamma(c)$ we assume that there is no connection between the drone and the tower and so both $P_W(c,t) = \mathcal{P}_{\text{LoS}}(c,t) = 0$. Note that $|\mathcal{U}_\gamma(c)|$ increases (or at least does not decrease) with cell height. Moreover, for each neighbor of $c \in \mathcal{B}$, i.e., $\forall c' \in \mathcal{A}(c)$, let $\mathcal{U}_\gamma(c,c') = \mathcal{U}_\gamma(c) \cap \mathcal{U}_\gamma(c')$ be the subset of the same antennas that a drone would see if it flew from c to c'.

Each combination of cell $c \in \mathcal{B}$ and tower $t \in \mathcal{S}_\gamma(c)$ has a *link-reliability* probability $\mathcal{P}_{\text{LR}}(c,t)$. Specifically,

$$\mathcal{P}_{\text{LR}}(c,t) = \mathcal{P}_{\text{LoS}}(c,t)\ P_W(c,t), \tag{10}$$

that returns the product among the probability of being in LoS, and the normalized received power at the drone. So, it holds that $0 \le \mathcal{P}_{\text{LR}}(c,t) \le 1$.

D. The Multi-Layer Conceptual Construction

In this section, we show how we build our three layers, i.e., for the no-fly zones, risk-map, and wireless infrastructure layers, respectively, given the box $\mathcal{B}$. The pseudocode of the construction is reported in Algorithm 1.

Initially, all layers are empty (Algorithm 1, Line 1), then sequentially, the no-fly zone/obstacle layer with $\mathcal{P}_{\text{FF}}$ (Line 2), the risk-map layer using $\mathcal{P}_{\text{GS}}$ (Line 7), and finally, the wireless infrastructure layer with $\mathcal{P}_{\text{LR}}$ (Line 10)

Algorithm 1: Multi-Layer Conceptual Construction

```
1  Initialize empty layers
   // no-fly zone/obstacles layer
2  for $c = (x_c, y_c, 1) \in \mathcal{B}$ do
3      if is forbidden to fly in c depending on $\mathcal{P}_{FF}(c)$ then
4          $\max_h \leftarrow$ forbidden up here ($h$ in case of airports)
5          for $c' = (x_c, y_c, z_{c'}) \in \mathcal{B} : z_{c'} = 1, \dots, \max_h$ do
6              forbid flight in $c'$
   // risk-map layer
7  for $c \in \mathcal{B}$ : flight is allowed do
8      $\mathcal{R}(c) \leftarrow \mathcal{P}_{\text{event}}(c)\ \mathcal{P}_{\text{impact}}(c)\ \mathcal{P}_{\text{fatality}}(c)$ (see Eq. (1))
9      $\mathcal{P}_{\text{GS}}(c) \leftarrow 1 - \mathcal{R}(c)$ (see Eq. (5))
   // wireless infrastructure layer
10 for $c \in \mathcal{B}$ : flight is allowed do
11     for $t \in \mathcal{U}_\gamma(c)$ (see Eq. (9)) do
12         calculate $\mathcal{P}_{\text{LoS}}(c,t)$ (see Eq. (6))
13         calculate $P_W(c,t)$ (see Eq. (7))
14         $\mathcal{P}_{\text{LR}}(c,t) \leftarrow \mathcal{P}_{\text{LoS}}(c,t)\ P_W(c,t)$ (see Eq. (10))
```

IV. The Graph Structure

In this section, we devise a graph-based structure based on the multi-layer concept, used then by path planning algorithms.

A. Path Dependability

After we presented the multi-layer concept, and before building the graph-based structure, we need to introduce some fundamental prerequisites. The graph we will create has vertices that represent "cells" or "cells associated with a tower", and edges that represent "movements between adjacent cells" or "tower handovers". Precisely:

- v_c represents the cell c,
- v_c^t represents the status of residing in cell c while communicating with the tower t,
- $(v_c^t, v_{c'}^t)$ represents the movement from cell c to cell c' being connected to the tower t, and
- the pair of edges $(v_c^t, v_c); (v_c, v_c^{t'})$ represents the handover between tower t and t' in cell c.

The towers t and t' are two towers visible in $\mathcal{S}_\gamma(c)$ (note that the cell c specifies also the altitude and γ the visibility factor). The edges on the graph are weighted based on a *dependability* score, which refers to the level of trustworthiness and robustness of that edge, quantified by a probability. Such a probability/score is determined by *ground-safeness*, *link-reliability*, and *handover-success-rate*, indicating flying safety, communication strength, and tower connection success in the respective cell, respectively. Specifically, ground-safeness, link-reliability, and handover-success-rate are obtained from the framework information as follows.

In the next, consider two vertices v_c^t and $v_{c'}^t$ that represent adjacent cells c and c' with the same tower t. So, there exists a direct edge $(v_c^t, v_{c'}^t)$ from v_c^t to $v_{c'}^t$.

*a) **Ground-Safeness**:* The ground-safeness is the dependability of the link with respect to the ground and the possible related safety risks. It is defined as $\mathcal{P}_{\text{GS}}((v_c^t, v_{c'}^t)) = 1 - \mathcal{R}(c')$ (see Eq. (5)), where $\mathcal{P}_{\text{GS}} = 0$ represents the highest risk, whereas $\mathcal{P}_{\text{GS}} = 1$ represents no risk. It is independent of the tower t.

*b) **Link-Reliability**:* The link-reliability is defined as $\mathcal{P}_{\text{LR}}((v_c^t, v_{c'}^t)) = \mathcal{P}_{\text{LoS}}(c', t)\ P_W(c', t)$ (see Eq. (10)), which combines the probability to be in LoS, and the normalized received power by tower t at the destination cell c'. The normalized received power is obtained by applying Eq. (7) and normalizing it as follows:

$$P_W(c',t) = \begin{cases} 1 & \text{if } P_W(c',t) > -65 \\ 0.999 & \text{if } -75 < P_W(c',t) \leq -65 \\ 0.9 & \text{if } -85 < P_W(c',t) \leq -75 \\ 0.85 & \text{if } -95 < P_W(c',t) \leq -85 \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

where the right-hand values[1] are represented in dBm.

*c) **Handover-Success-Rate**:* When moving between two cells keeping the same tower, no handover happens and the link is not at risk. Whereas, when the drone moves from the tower t to t', there is a risk that the communication will be lost. We assume that the handover is handled inside a cell. Hence, we assume that the Handover-Success-Rate is a value $\mathcal{P}_{\text{HSR}} \in [0,1]$, that represents the success of establishing the communication via another nearby tower t'. The more the received signal strength at the tower t' and the more is the height of the drone, the more will be the handover-success-rate [22]. To represent the handover from t to t' in cell c, we concatenate the two edges (v_c^t, v_c) and $(v_c, v_c^{t'})$. Note that we do not create a connection directly from $(v_c^t, v_c^{t'})$ in order to reduce the number of edges from a quadratic number in the visible towers to a linear number.

The weight of the edge $\mathcal{P}((v_c^t, v_c)) = 1$, and the weight $\mathcal{P}((v_c, v_c^{t'})) = \mathcal{P}_{\text{HSR}}$. Precisely, $\mathcal{P}_{\text{HSR}}$ for $(v_c, v_c^{t'})$ depends on $P_W(c, t')$, and on z_c (height of c). In our experiments in Section VI, when c is at the highest altitude ($z_c = 3$), the weight $\mathcal{P}_{\text{HSR}}$ depends on P_W according to Eq. (12):

$$\mathcal{P}_{\text{HSR}}((v_c, v_c^{t'})) = \begin{cases} 1 & \text{if } P_W(c,t') > -75 \\ 0.975 & \text{if } -85 < P_W(c,t') \leq -75 \\ 0.95 & \text{if } -95 < P_W(c,t') \leq -85 \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

Specifically, it takes on different values based on intervals of the power $P_W(c,t')$: if $P_W(c,t')$ is above $-75\,\mathrm{dBm}$, $\mathcal{P}_{\text{HSR}}$ is set to 1, suggesting a strong wireless connection; if it is between -75 and $-85\,\mathrm{dBm}$, $\mathcal{P}_{\text{HSR}}$ is slightly reduced to 0.975, reflecting a slightly weaker but still reliable connection; if it falls further between -85 and $-95\,\mathrm{dBm}$, $\mathcal{P}_{\text{HSR}}$ decreases to 0.95, indicating a less reliable connection; and finally, if it is below $-95\,\mathrm{dBm}$, $\mathcal{P}_{\text{HSR}}$ is set to 0, indicating an unreliable or nonexistent connection.

For lower altitudes, i.e., $z_c = 2$ and $z_c = 1$, $\mathcal{P}_{\text{HSR}}$ is affected by a decreasing factor, which is independent of the position of the cell and depends only on the altitude of c:

$$\mathcal{P}_{\text{HSR}}((v_c, v_c^{t'})) = \begin{cases} 0.975\ \mathcal{P}_{\text{HSR}}((v_c, v_c^{t'})) & \text{if } z_c = 2 \\ 0.95\ \mathcal{P}_{\text{HSR}}((v_c, v_c^{t'})) & \text{if } z_c = 1 \end{cases} \quad (13)$$

As we will see in the next section, any edge $(v_c^t, v_{c'}^t)$ is associated with a dependability score given by the product of

[1]The power thresholds are suggested by the classification of the communications in *excellent*, *good*, etc given in [32].

the ground-safeness and the link-reliability. Any edge $(v_c^t, v_c^{t'})$ has the handover-success-rate as weight. We will conclude our construction by saying that any *path* in the weighted graph has a *dependability score* given by the product of the dependability score of the edges belonging to the path.

In the following section, we detail even more the graph construction and we give a graph-construction example.

B. Graph Construction

Let $\mathcal{G} = (V, E)$ be the weighted directed graph (called from now on as the *dependability graph*) which is defined by a set V of vertices and a set E of edges, built from the box $\mathcal{B}$ and the previous layers. We build $\mathcal{G}$ as follows. For each cell $c \in \mathcal{B}$, we create a vertex v_c that represents the cell c, plus $|\mathcal{U}_\gamma(c)|$ additional vertices to represent the towers to which it is possible to connect from c, denoted as v_c^t, where $t \in \mathcal{U}_\gamma(c)$. Formally, the set of vertices is $V = \{v_c : c \in \mathcal{B}\} \cup \{v_c^t : t \in \mathcal{U}_\gamma(c)\ \forall c \in \mathcal{B}\}$, while the set of edges E is somewhat more complicated. In general, there are two types of edges: *intra-edges* (i.e., tower handover) and *inter-edges* (i.e., moves between adjacent cells). An intra-edge is in the form of (v_c^t, v_c) or (v_c, v_c^t), while an inter-edge is in the form of $(v_c^t, v_{c'}^t)$. The pseudocode of the construction is in Algorithm 2.

Algorithm 2: Graph Construction

```
1  for $c \in \mathcal{B}$ do
2      create vertex $v_c$
3      for $t \in \mathcal{U}_\gamma(c)$ do
4          create vertex $v_c^t$
           // intra-edges (see Eqs. (12) (13))
5          add edge $(v_c, v_c^t)$ with cost $\mathcal{P}_{\text{HSR}} \neq 1$
6          add edge $(v_c^t, v_c)$ with cost $\mathcal{P}_{\text{HSR}} = 1$
7  for $c \in \mathcal{B}$ do
8      for $c' \in \mathcal{A}(c)$ do
9          for $t \in \mathcal{U}_\gamma(c, c')$ do
               // inter-edges (see Eqs. (10) (5))
10             add edge $(v_c^t, v_{c'}^t)$ with cost $\mathcal{P}_{\text{LR}}(c', t)\ \mathcal{P}_{\text{GS}}(c')$
```

Regarding the *intra-edges*, these are used to connect a drone located in a given cell to all the possible visible towers in that cell. For each $v_c^t \in V$ we add a directed edge $(v_c^t, v_c) \in E$ with cost $\mathcal{P}_{\text{HSR}} = 1$ (no handover) and a directed edge $(v_c, v_c^t) \in E$ whose cost $\mathcal{P}_{\text{HSR}}$ (do handover) depends on the signal quality and the drone's height (Algorithm 2, Line 1). Crossing this latter edge represents the drone connecting to a different tower. Regarding the *inter-edges*, these are used when the drone moves between adjacent locations. For each neighbor of $c \in \mathcal{B}$, i.e., $\forall c' \in \mathcal{A}(c)$, we connect pairs of vertices that represent the same tower t in the two different cells c and c', i.e., the towers in $t \in \mathcal{U}_\gamma(c, c')$. For each of these pairs, we add a directed edge $(v_c^t, v_{c'}^t) \in E$ with cost $\mathcal{P}_{\text{LR}}(c', t)\ \mathcal{P}_{\text{GS}}(c')$. Crossing these edges represents the drone moving, e.g., from cell c to c', thus considering the ground-safeness and the link-reliability of the coming cell.

An example is reported in Figure 2. Consider $\gamma = 1$. The yellow vertex located at cell $(2, 3)$ has 8 neighbors, which correspond to the cells located within the green highlighted square. For the purposes of this example, we will assume that the drone height is fixed and so we omit the z_c coordinate when referring to a cell. Moreover, there are 3 towers: t_a, t_b, and t_c. Among these, only t_a and t_b are in range with position $(2, 3)$, and therefore intra-edges from $v_{(2,3)}$ are only created between $v_{(2,3)}^a$ and $v_{(2,3)}^b$. The blue cell is in position $(3, 3)$ which is in range with all 3 towers. As we can see, both $v_{(2,3)}$ and $v_{(3,3)}$ share the towers t_a and t_b, and hence inter-edges $(v_{(2,3)}^a, v_{(3,3)}^a)$, and $(v_{(2,3)}^b, v_{(3,3)}^b)$, are created. The same can be repeated for the red cell in $(3, 2)$.

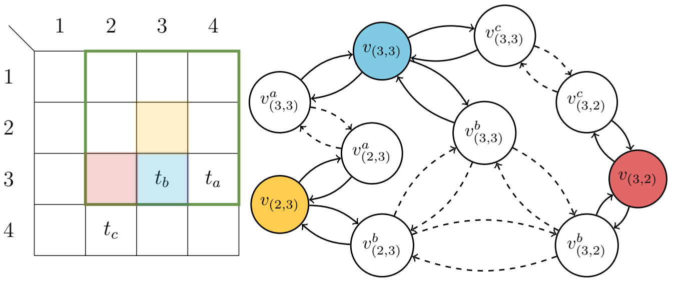


Fig. 2. Example of subgraph with 3 cells: $(2, 3)$ (yellow), $(3, 3)$ (blue), and $(3, 2)$ (red); also, $h = 1$ and $\gamma = 1$. The grid shows the cells inside the box. Solid and dashed edges are the intra-edges and the inter-edges, respectively.

In conclusion, we discuss the size of the constructed graph. The number $|V|$ of vertices of $\mathcal{G}$ is upper-bounded by $|V| \leq nmh + nm \sum_{z_c=1}^{h} |\mathcal{S}_\gamma(c)|$, where $|\mathcal{S}_\gamma(c)| \in \mathcal{O}(\gamma^2 z_c^2)$ (see Eq. (8)). Concerning the edges, we can upper bound $|E| \leq nm \sum_{z_c=1}^{h} |\mathcal{S}_\gamma(c)| + nm \sum_{z_c=1}^{h} 26|\mathcal{S}_\gamma(c)|$, where the first summation refers to the intra-edges, while the second summation refers to the inter-edges. Hence, $(|V| + |E|) \in \mathcal{O}(nm\ \gamma^2 h^3)$, where nmh is the number of cells in $\mathcal{B}$ and γ is the visibility factor.

C. Problem Formulation and Solution

Given a box $\mathcal{B}$, the graph-based multi-layer structure, and a starting and a destination cell $s, g \in \mathcal{B}$ respectively, we are in a position to solve the *Maximum Dependability Path Problem* (MDP2) whose objective is to find a drone path Λ^* that starts in s and finishes in g such that the dependability of $\Lambda^* = \{(s, \cdot), \ldots, (\cdot, g)\}$ is maximized. Recall that edges $e \in E$ are in the form of (v_c^t, v_c) or (v_c, v_c^t) (intra-edge), and $(v_c^t, v_{c'}^t)$ (inter-edge), and that their cost is $\mathcal{P}(e)$ assigned according to Algorithm 2. Formally [13],

$$\Lambda^* = \arg\max_{\Lambda} \prod_{e \in \Lambda} \mathcal{P}(e) \tag{14}$$

In a nutshell, it is easy to see that MDP2 can be solved by applying the Dijkstra's algorithm [33]. Precisely, to maximize Eq. (14), one can apply any $\log$ function (with base > 1) to the product, thus converting the product into a sum. To maximize the argument of the $\log$ function in the right side of Eq. (16),

$$\Lambda^* = \arg\max_{\Lambda} \log\left(\prod_{e \in \Lambda} \mathcal{P}(e)\right) \tag{15}$$

$$= \arg\max_{\Lambda} \sum_{e \in \Lambda} \log\left(\mathcal{P}(e)\right) \tag{16}$$

is the same as minimizing Eq. (17) since all the addends are negative

$$\Lambda^* = \arg\min_{\Lambda} \sum_{e \in \Lambda} |\log(\mathcal{P}(e))|. \tag{17}$$

Hence, the original objective of maximizing dependability from s to g is then equivalent to minimizing the path cost from s to g, where the path cost is defined as the sum of the absolute value of the logarithms of the original weights (probabilities). The path returned by Dijkstra's algorithm solves MDP2, and it is the path with maximum dependability in $\mathcal{B}$. The dependability score can be computed by elevating the base of the $\log$ to the path cost returned by Dijkstra's algorithm.

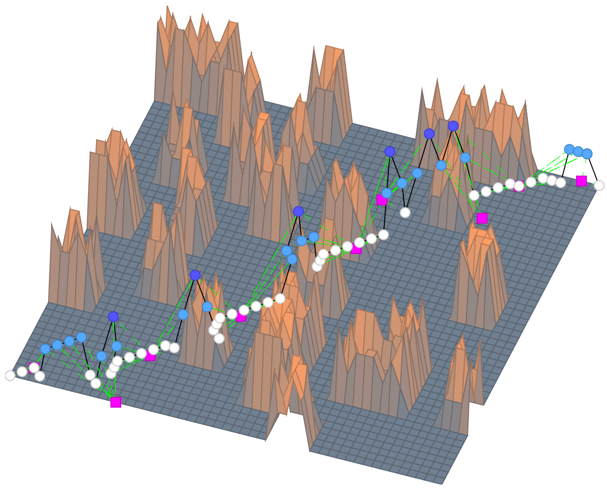


Fig. 3. A path example with $s = (0, 0, 1)$ and $g = (49, 49, 1)$.

Figure 3 illustrates a hypothetical small instance of MDP2 with random values. $s = (0, 0, 1)$ is on the left, and $g = (49, 49, 1)$ on the right. The drone's path is in black; the traversed vertices at the level 1 in white, at level 2 in cyan, and at level 3 in blue; the used towers in magenta; the actual connections among the drone and the towers in dashed green; and the obstacles clearly visible.

Hence, we can plan BVLoS operations effectively since the Dijkstra's algorithm is polynomially solvable. Note that the optimality is related *only* within the discretized $\mathcal{B}$ area. In fact, better paths from a given source to a destination could pass through other vertices outside $\mathcal{B}$.

V. The Corridor Structure

In Section IV, given a box $\mathcal{B}$, we proposed a graph-based structure in order to optimally solve MDP2 by invoking the Dijkstra's algorithm. Although this approach is optimal in the discretized area, the main disadvantage is that it considers the whole $\mathcal{B}$, which requires too much space to store the graph, and it is also computationally heavy.

We propose a much lighter but sub-optimal approach to solve MDP2 that considers a subset of cells of $\mathcal{B}$ that we call *Corridor* $\mathcal{C}$, i.e., a subset of cells along a way between the starting and the destination cell. More precisely, we create a corridor with a given width $\mathcal{C}_W$ (in terms of number of cells) that starts at the starting cell $s = (x_s, y_s, z_s)$ and follows some specific structures, and ends at the destination cell $g = (x_g, y_g, z_g)$. The length of the corridor $\mathcal{C}_L$ depends on the position among s and g. Moreover, the height of the corridor is equal to the height of the box $\mathcal{B}$, i.e., $\mathcal{C}_H = \mathcal{B}_H$.

The detailed construction is depicted in the next section.

A. Construction

We start with some definitions, regardless of the positions of both s and g, that help us to define our approach clearly.

- **Horizontal (Vertical) block**, denoted as **H-block** (**V-block**), is a set of adjacent cells aligned horizontally (vertically) within the box. It consists of cells arranged in a single row (column). The length of a H-block (V-block) is the number of cells it contains in that row (column).
- **Horizontal (Vertical) straight corridor** consists of a set of $2\nu + 1$ adjacent H-blocks (V-blocks), where ν is an integer number that represents the distance of the farthest H-block (V-block) from the central one (see Figure 4, left). So, $\mathcal{C}_W = 2\nu + 1$. In the example, $\nu = 2$.
- **Horizontal (Vertical) zig-zag corridor** consists of a set of V-blocks (H-blocks) repeatedly added on the left or right (above or below) while each block is shifted one cell to the top or bottom (to the right or left). It holds that $\mathcal{C}_W = 2\nu + 1$. In Figure 4, right, there is a horizontal zig-zag corridor and $\nu = 1$.

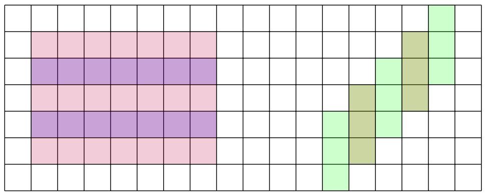


Fig. 4. Examples of different types of corridors: (left) a horizontal straight corridor formed by many H-blocks, (right) a horizontal zig-zag corridor formed by many V-blocks.

We now proceed by explaining each step of the proposed corridor-based approach. According to the coordinates of the starting and the destination cell, there can be 16 different cases (see Figure 5). So, based on any of these cases, we will create different kinds of corridors between the starting and the destination cells. The pseudocode of the proposed approach is given in Algorithm 3. We briefly denote horizontal and vertical as H and V, respectively.

Algorithm 3: Corridor Construction

```
1  $s = (x_s, y_s, 1), g = (x_g, y_g, 1)$
2  $x' \leftarrow |x_s - x_g|$, $y' \leftarrow |y_s - y_g|$
3  if $x' \geq y'$ then
4      if $g$ is on the top-right or bottom-right of $s$ then
5          create H zig-zag corridor $s \rightsquigarrow t = (x_s + y', y_g, 1)$
6      else
7          create H zig-zag corridor $s \rightsquigarrow t = (x_s - y', y_g, 1)$
8      create H straight corridor $t \rightsquigarrow g$
9  else
10     if $g$ is on the top-right or top-left of $s$ then
11         create V zig-zag corridor $s \rightsquigarrow t = (x_g, y_s + y', 1)$
12     else
13         create V zig-zag corridor $s \rightsquigarrow t = (x_g, y_s - y', 1)$
14     create V straight corridor $t \rightsquigarrow g$
```

W.l.o.g., we only consider the cases in which the destination cell $g = (x_g, y_g, 1)$ is on the top-right of the starting cell $s = (x_s, y_s, 1)$. The procedure for the other cases, when the destination cell is on the top-left, bottom-right, and bottom-left of the starting cell, is analogous. We now analyze different cases (e.g., Cases 1 to 5) enumerated in Figure 5.

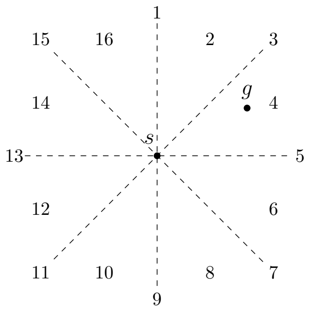


Fig. 5. All possible different cases according to the coordinates of the starting cell $s = (x_s, y_s, 1)$ and the destination cell $g = (x_g, y_g, 1)$.

The idea behind the proposed corridor-based approach is to start from the starting cell s and use zig-zag corridors until the destination cell g can be reached by a straight corridor. Note that the length $\mathcal{C}_L$ of the corridor connecting s and g is $|x_s - x_g| + 1$ for horizontal corridors ($|y_s - y_g| + 1$ for vertical corridors), and $\mathcal{C}_W = 2\nu + 1$.

(i) **Case 1**: If $y_g = y_s$, we create a V straight corridor with $\mathcal{C}_L = |y_s - y_g| + 1$ between s and g (see Figure 6, left). Note that s and g belong on the first and last cell in the central V-block.

(ii) **Case 2**: The corridor is built by combining a V zig-zag corridor with length $\mu = |x_s - x_g| + 1$, and a V straight corridor with length $|y_g - y_s - \mu|$ (see Figure 6, right). Note that the corridor construction starts from s and grows toward g.

(iii) **Cases 3 and 4**: For Case 3, we create a H zig-zag corridor with $\mathcal{C}_L = |x_s - x_g| + 1$. For Case 4, the corridor is constructed by combining a H zig-zag corridor of length $\mu = |y_s - y_g| + 1$ and a H straight corridor of length $|x_g - x_s - \mu|$ (see Figure 7, left).

(iv) **Case 5**: If $x_g = x_s$, corridors are similarly created as (i) but horizontally (see Figure 7, right).

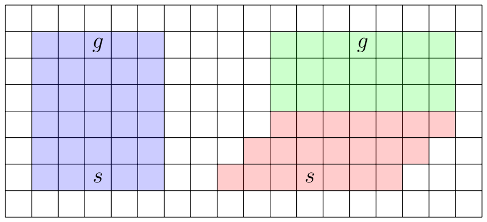


Fig. 6. Corridor construction based on the positions of the starting and the destination cells: (left) Case 1 with $\nu = 2$, (right) Case 2 with $\nu = 3$.

After creating the corridor, the proposed approach constructs the corresponding graph $\mathcal{G}^C \subseteq \mathcal{G}$ only by taking the corridor and the set of towers (including the towers outside the corridor[2]) into account. So, the resulting subgraph $\mathcal{G}^C$ is then used to optimally find a local solution, by minimizing

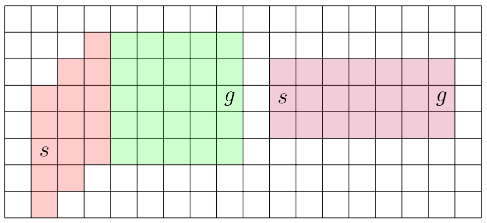


Fig. 7. Corridor construction based on the positions of the starting and the destination cells: (left) Case 4 with $\nu = 2$, (right) Case 5 with $\nu = 1$.

Eq. (17), that would be sub-optimal for the original global graph $\mathcal{G}$. The advantage of the proposed approach is that the width of the corridor can be suitably chosen as an input parameter, especially when the volume of the box starts to be quite large. Also, by selecting different widths the resulting computed paths will be different. This of course affects the computational complexity of the problem. More precisely, if $c_l = |\mathcal{C}_L|$ and $c_w = |\mathcal{C}_W|$, it holds that $|V| \in \mathcal{O}(c_l c_w \ \gamma^2 h^3)$, and $|E| \in \mathcal{O}(c_l c_w \ \gamma^2 h^3)$, where $c_l c_w h \ll nmh$ is the number of cells in the corridor. Note that $c_w \in \Theta(\nu)$ and c_l is related with the distance between s and g, and can be either n or m.

VI. Experimental Evaluation

We perform an extensive experimental evaluation when approaching the MDP2. In Section VI-A we describe the used parameters in our scenarios, while in Section VI-B and Section VI-C we evaluate the path dependability when solving MDP2 in the general graph and in the corridor structure, respectively. Our results are founded on the assumptions made for ground-safeness, link-reliability, and normalized power in Sections III and IV. For the towers, we extract their position from a real dataset [34]. Our algorithms are in C++17[3], and run on an AMD Ryzen 3 PRO computer with 32 GB of RAM.

A. Setting

We have selected three distinct environments that vary in complexity to assess the performance of our model:

- The *rural environment* features a low population density, a low building density (low sheltering), and few antennas, allowing a clear LoS. However, the limited number of towers may pose difficulties in maintaining consistent communications between the drone and the operator.
- The *suburban environment* has a higher population and building density than the rural area and a larger number of antennas, providing a balance between rural and urban connectivity, and ground risk.
- The *urban environment* has the highest population and building density, resulting in a high sheltering factor, numerous obstacles, and a large number of antennas. This makes it easier to maintain connectivity.

We consider a box $\mathcal{B}$ with $\mathcal{B}_L = \mathcal{B}_W = 10\,\text{km}$, and $\mathcal{B}_H = 90\,\text{m}$. Each cell $c \in \mathcal{B}$ has $\ell_L = \ell_W = 10\,\text{m}$, and $\ell_H = 30\,\text{m}$. Accordingly, we have $n = m = 1000$ and $h = 3$, so we limit to 3 different heights.

[2]Each cell knows about all nearby towers, not just the ones in the corridor.

[3]Implementation is available at https://github.com/ulisse91/BVLOS.

Concerning the no-fly zone/obstacle layer, we randomly generate obstacles on the ground with $\mathcal{P}_{\text{FF}}$ that depends on the environment. To be more precise, the rural environment is set to $\mathcal{P}_{\text{FF}} = 5\%$, while it is set to 15% and 25% for the suburban and urban environments, respectively. We generate potential obstacles on these layers, increasing their height up to the maximum allowed. As a general trend, the urban environment presents more restrictions on the drone flight path compared to rural or suburban settings.

TABLE I
TABLE OF PARAMETERS.

Parameter	Description
$\mathcal{B}_L, \mathcal{B}_W, \mathcal{B}_H$	Width, length, and height of the box $\mathcal{B}$: $10\,\text{km}$, $10\,\text{km}$, and $90\,\text{m}$
ℓ_L, ℓ_W, ℓ_H	Width, length, and height of each cell within the box: $10\,\text{m}$, $10\,\text{m}$, and $30\,\text{m}$
n, m	Number of cells in the length and width: 1000 and 1000
h	Number of cells in the height: 3
$\mathcal{P}_{\text{FF}}$	Probability of obstacles in the no-fly zone/obstacle layer: 5% (rural), 15% (suburban), 25% (urban)
$\mathcal{P}_{\text{event}}$	Descent prob. events: $\frac{1}{200}$ (ballistic), $\frac{1}{200}$ (uncontrolled glide), $\frac{1}{100}$ (parachute), $\frac{1}{250}$ (fly-away)
$\mathcal{P}_{\text{impact}}$	Impact prob. based on ρ and A_{exp}
$r_{\text{p}}, h_{\text{p}}$	Radius and height of a person: $0.248\,\text{m}$ and $1.587\,\text{m}$
r_{uav}	Radius of the vehicle: $0.88\,\text{m}$
ψ	Impact angle on the ground: 1.04
$\mathcal{P}_{\text{fatality}}$	Probability of fatality based on sheltering factors
$S(c)$	Sheltering factors: 0 (no obstacles), 2.5 (sparse tree), 5 (vehicles and low buildings), 7.5 (high buildings), 10 (industrial building)
τ	Threshold energy for fatality: $100\,\text{kJ}$
η	Energy absorbed per unit area: $34\,\text{J}$
δ	Mass of the drone: $3.75\,\text{kg}$
v_{init}	Initial velocity of the drone: $20\,\text{m/s}$
$G(t), G(c)$	Gains of the antennas: 1 (isotropic signal)
$P_W(t)$	Transmitted power at the tower t: 1–5 W
$P_W(c,t)$	Normalized received power
$\mathcal{P}_{\text{HSR}}(c,t)$	Handover-Success-Rate probability
λ	Wavelength of the transmitted signal: 0.43–0.05 m
α, β	For $\mathcal{P}_{\text{LoS}}$: $\alpha = 0.1, \beta = 750$ (rural), $\alpha = 0.3, \beta = 500$ (suburban), $\alpha = 0.5, \beta = 300$ (urban)
$\mathcal{C}_W = \omega$	Corridor width: 5, 11, 21, 31

About the risk-map layer, we used general parameters from Primatesta et al. [8]. Four descent event types and $\mathcal{P}_{\text{event}}$ probabilities are considered (see Table I). $\mathcal{P}_{\text{impact}}$ (see Eq. (2)) depends on ρ and on A_{exp}: ρ is publicly available depending on the city we choose (see Table III), while for computing A_{exp} we refer to Table I. Interestingly, with the used parameters we have that the exposed lethal area $A_{\text{exp}} \approx 6\,\text{m}^2$ while our area cell is $100\,\text{m}^2$. To simulate different environments, we randomly select a sheltering factor for the rural environment from $0, 2.5, 5$, for the suburban environment from $2.5, 5, 7.5$, and for the urban environment from $5, 7.5, 10$.

Moreover, with regard to the wireless infrastructure layer, we used a real dataset [34] for retrieving the real GPS positions of cellular towers. As mentioned above, we assume at most one tower for each cell (in our case, a square of $100\,\text{m}^2$). Since power and wavelength information is not available for all towers, we randomized these values. We also assumed isotropic antennas with unitary gains. In particular, knowing that 2/4/5G towers have an average power P_W of 1–5 W and frequencies in the range 0.7–6 GHz, which correspond to wavelengths λ between 0.43–0.05 m, we randomized these values within the ranges. Then, we normalize these values according to Eq. (11). Furthermore, concerning the probability of being LoS, we used the α and β parameters from Al-Hourani et al. [30]. In particular, for the rural environment, we set $\alpha = 0.1, \beta = 750$, for the suburban environment, we set $\alpha = 0.3, \beta = 500$, and for the urban environment, we set $\alpha = 0.5, \beta = 300$. Note that the ground and slant distances are Euclidean, although we discretized the positions. Finally, the $\mathcal{P}_{\text{HSR}}$ weights are computed according to Eqs. (12) and (13).

As just explained, we generate the edge weights on the graph by considering three different components. For inter-egdes, the weight is the product of the link-reliability ($\mathcal{P}_{\text{LR}}$) and ground-safeness ($\mathcal{P}_{\text{GS}}$). For the intra-edges, the weight is the handover-success-rate ($\mathcal{P}_{\text{HSR}}$). A component can be fixed for each edge of the graph to 1 to disable it, i.e., if we do not want that the path dependability depends on it. Hence, the edge dependability can be selectively tuned through a *mode* $\mathcal{M}$. Table II reports the modes $\mathcal{M}$ that can be used, where 1 means "enabled" and 0 means "disabled". For instance, $\mathcal{M} = 5$ considers only weights for $\mathcal{P}_{\text{HSR}}$ and $\mathcal{P}_{\text{LR}}$, neglecting $\mathcal{P}_{\text{GS}}$ (risk-less environment). So, in this example, we set $\mathcal{P}_{\text{GS}} = 1$ to disable ground-safeness. The default mode is $\mathcal{M} = 7$ with all the components enabled.

TABLE II
COMBINATIONS OF PROBABILITIES WHEN SETTING THE EDGE WEIGHTS.

$\mathcal{P}_{\text{HSR}}$	$\mathcal{P}_{\text{GS}}$	$\mathcal{P}_{\text{LR}}$	$\mathcal{M}$	$\mathcal{P}_{\text{HSR}}$	$\mathcal{P}_{\text{GS}}$	$\mathcal{P}_{\text{LR}}$	$\mathcal{M}$
0	0	0	0	1	0	0	4
0	0	1	1	1	0	1	5
0	1	0	2	1	1	0	6
0	1	1	3	1	1	1	7

Finally, we varied the visibility factor γ according to the tower density. The whole construction asymptotically takes $\mathcal{O}(nm\gamma^2h^3)$, and γ significantly impacts the generation of the framework. We handle γ values up to 25 in rural area, while we can only handle $\gamma = 5$ in very dense urban area.

B. Results with Optimal Strategy

In this section, we report the experimental results of our proposed framework and solution. We propose three different experiments. In the first one (Section VI-B1), we show the running time and size of the framework when varying γ for a given environment. In the second one (Section VI-B2), we solve the MDP2 by tuning different edge weights. Finally, in the third one (Section VI-B3), we investigate how the maximum dependability path changes when varying γ.

1) Size of Multi-Layer Framework: In this experiment, we compare the framework construction in different environments considered according to the population density of the cities. In particular, densities ρ around $200/\text{km}^2$, $1000/\text{km}^2$, and $6000/\text{km}^2$, are assumed to be rural, suburban, and urban, respectively. For each environment, we test 3 different instances with relevant Italian cities from an available dataset of towers [34]. For each instance, we also vary the visibility factors depending on the number of towers. Finally, we save some interesting indicators, as reported in Table III.

TABLE III
GENERATION OF THE FRAMEWORK UNDER DIFFERENT PARAMETERS: RUNNING TIME AND FRAMEWORK'S SIZE.

Env.	City	ρ	$\lvert\mathcal{T}\rvert$	γ	$\mathcal{W}$ time	$\mathcal{G}$ time	$\lvert V\rvert$	$\lvert E\rvert$
Rural	Caltanissetta	141	624	5	11 s	18 s	3.8 M	28 M
				25	240 s	294 s	25 M	0.7 G
	Grosseto	172	1161	20	179 s	294 s	29 M	0.8 G
	Viterbo	162	1124	20	181 s	477 s	28 M	0.8 G
Suburb.	Modena	1010	4254	15	144 s	1445 s	55 M	1.5 G
	Piacenza	866	2642	15	125 s	548 s	35 M	0.9 G
	Verona	1287	6626	10	79 s	744 s	40 M	1.0 G
Urban	Milano	7430	27947	5	45 s	653 s	40 M	1.0 G
	Roma	2135	21243	5	38 s	374 s	31 M	0.7 G
	Torino	6526	15401	7	58 s	773 s	41 M	1.0 G

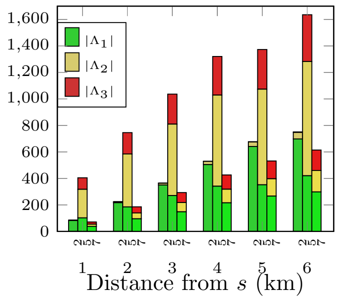
In Table III, the first, second, and third groups of three rows represent urban, suburban, and urban cities, respectively. In particular, the city of Caltanissetta is reported with two different values of γ (5 and 25) for a brief comparison.

The heavy computation time is mainly associated with the creation of the wireless infrastructure layer $\mathcal{W}$, and the graph $\mathcal{G}$. The running time results show that the value of γ has a significant impact on the creation of $\mathcal{W}$. In general, a larger value of γ implies a longer time to create $\mathcal{W}$ regardless of the number of towers. Moreover, having a large number of towers also leads to a significant increase in the time required to create $\mathcal{G}$. In Table III, e.g., the running time and the number of vertices and edges in the Caltanissetta city (which has 624 towers) vary greatly depending on whether γ is set to 5 or 25. In fact, when $\gamma = 25$ it takes around $20\times$ more time to build the whole structure than when $\gamma = 5$. On the other hand, the Milano city (which has 27947 towers) takes $4\times$ and $36\times$ more than Caltanissetta city with $\gamma = 5$ for building $\mathcal{W}$ and $\mathcal{G}$, respectively. This highlights the importance of the choice of γ in determining the computational efficiency of the model. Indeed, for the city of Modena, we managed to set $\gamma = 15$ obtaining a graph with 55 millions of vertices and more than 1.5 billions of edges.

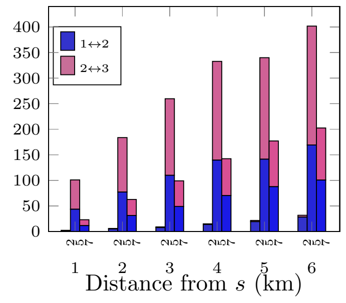
2) Maximum Dependability Path Analysis: In this experiment, we evaluate and analyze the generated paths by varying a few parameters. We consider the city of Roma ($|\mathcal{T}| = 21243$, $|V| = 31\,\text{M}$, and $|E| = 0.7\,\text{G}$) with $\gamma = 5$ and $\mathcal{M} \in \{2, 5, 7\}$. In other words, we consider the scenarios where only ground risk is considered ($\mathcal{M} = 2$), only ground risk is neglected ($\mathcal{M} = 5$), and the general one ($\mathcal{M} = 7$). We select $s = (500, 500, 1)$ as the source vertex (i.e., the center cell of the box), and we consider any destination vertex g. Then, we solve MDP2 by invoking the Dijkstra's algorithm. Recall that MDP2 aims to maximize Eq. (14) (*maximum dependability path*), which is the same as to minimize Eq. (17) (*shortest path*). We would like to remind the reader that Dijkstra's algorithm, when applied to a specific source vertex, calculates *all* the shortest paths from that vertex to any other vertex within the graph. However, in general, it can be terminated once the desired destination has been determined.

In Figure 8, the y-axis represents the number of traversed edges $|\Lambda|$ (grouped per height, where the subscripts 1, 2, and 3 refer to a height of $30\,\text{m}$, $60\,\text{m}$, and $90\,\text{m}$, respectively), the number of adjacent height swaps, the count of handovers (Hs), and the path dependability (Eq. (14)). These values are illustrated in Figure 8a, Figure 8b, Figure 8c, and Figure 8d, respectively. In the x-axis, we report the Euclidean distance in kilometers among s and g (i.e., $1, 2, \ldots, 6$) grouped by mode (i.e., $\mathcal{M} = 2, 5, 7$). In Figure 8, we randomly select 50 destination vertices g at different Euclidean distances in kilometers from s. Specifically, there are 50 vertices within each annulus of width $1\,\text{km}$ from s (see Figure 10).

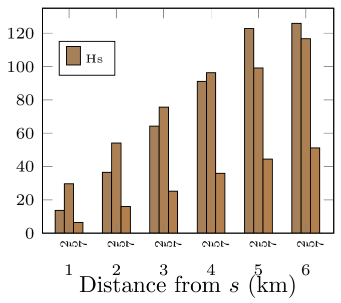
In $\mathcal{M} = 2$, only ground-safeness ($\mathcal{P}_{\text{GS}}$) is considered. As expected, we observe paths mainly along the lowest height because $\mathcal{P}_{\text{fatality}}$ depends on the height (higher speed of impact), and so the drone is not incentivized to swap too many times. In general, around 5% of the vertices are traversed at $h = 2$, a very few at the highest level, and the remaining at the lowest one. The number of swaps is really small compared to the other modes. In $\mathcal{M} = 2$, since $\mathcal{P}_{\text{HSR}} = 1$, changing tower has no cost. As handovers are considered in our model to be "ground-risk-less" events, the drone performs a significant number of them. In principle, the dependability of the paths slightly decreases when the Euclidean distance increases from the source s. This is reasonable since the drone has to travel along more edges.



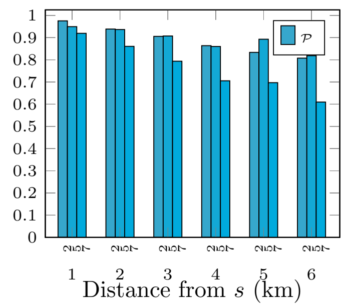
(a) Traversed edges.



(b) Swaps.



(c) Handovers.



(d) Dependability.

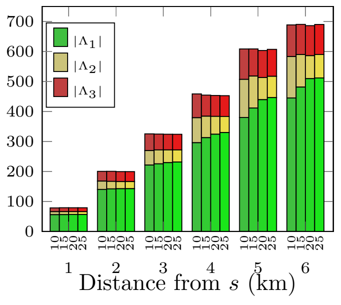
Fig. 8. Results when testing Roma fixing $\gamma = 5$ and varying $\mathcal{M} = 2, 5, 7$.

When $\mathcal{M} = 5$, both link-reliability ($\mathcal{P}_{\text{LR}}$) and handover-success-rate ($\mathcal{P}_{\text{HSR}}$) are taken into account, while the environment is considered risk-free. So, the drone is highly motivated to operate at higher altitudes where communication quality is superior, as the associated ground-risk is not considered. In fact, LoS probability ($\mathcal{P}_{\text{LoS}}$) increases with the drone's elevation. At higher altitudes, the drone can take advantage of better tower visibility, potentially facilitating more handovers. As a result, around half of the total vertices are located at $h = 2$, with the remainder roughly split between $h = 1$ and $h = 3$. So, the number of swaps also increases significantly, with a greater emphasis on swaps between the highest altitudes ($2 \leftrightarrow 3$) than the lowest ones ($1 \leftrightarrow 2$).

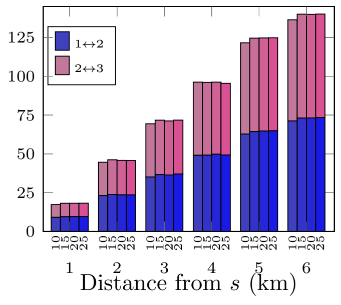
Finally, in $\mathcal{M} = 7$, we consider all components together. In general, these results present peculiarities from both $\mathcal{M} = 2$ and $\mathcal{M} = 5$. In fact, the number of traversed vertices is low as for $\mathcal{M} = 2$, but differently from it, the vertices at $h > 1$ are much more. Here, half of the vertices are at the lowest height, and the other ones are equally split at the remaining two heights. This is because $\mathcal{M} = 7$ takes into account the risk, so the drone is often constrained to fly at higher heights. Interestingly, the number of handovers here is halved with respect to the other evaluated modes. This is probably due to the fact that the drone has to fly at the lowest heights, where $\mathcal{P}_{\text{HSR}}$ is lower. In conclusion, the dependability is lower in general because it is weighted by all the components, i.e., risk, link quality, and handover-success-rate.

Although we analyze the path dependability, we know that dependability strongly depends on the assumptions. At the moment, we mixed real and random inputs. For example, the position of the tower is real, but the tower wavelength is randomly selected. Clearly, this strongly affects our results. However, the contribution of our paper is to show that the MDP2 can be solved polynomially on a very complex framework that merges many layers and allows customization of each layer. The relationship between the dependability results and the assumptions done will be the object of further investigations in the future.

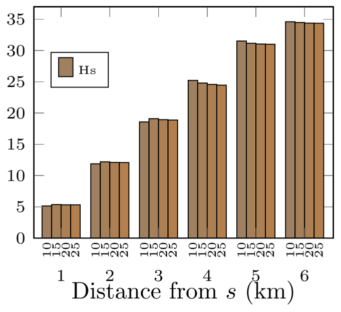
3) *Impact of Visibility Factor on Paths:* In this experiment, we want to see how the path changes when γ changes too. We consider the city of Caltanissetta ($|\mathcal{T}| = 624$) with $\mathcal{M} = 7$ and $\gamma \in \{10, 15, 20, 25\}$. We recorded $|V| = 6\,\mathrm{M}$ and $|E| = 0.1\,\mathrm{G}$ with $\gamma = 10$, while when $\gamma = 25$ we have $|V| = 24\,\mathrm{M}$ and $|E| = 0.7\,\mathrm{G}$. We select $s = (500, 500, 1)$ as the source vertex, and as before, we consider any destination vertex g by solving MDP2 by invoking the Dijkstra's algorithm. In the x-axis, we also report the used visibility factor γ.



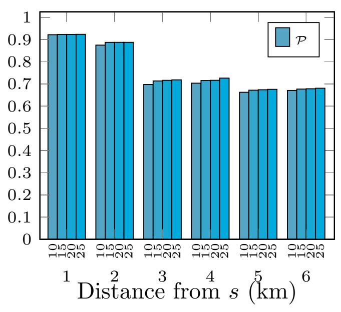
(a) Traversed edges.



(b) Swaps.



(c) Handovers.

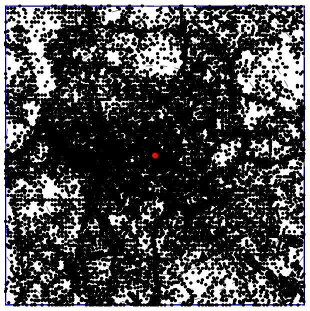


(d) Dependability.

Fig. 9. Results with Caltanissetta fixing $\mathcal{M}$=7 and varying γ=10, 15, 20, 25.

In Figure 9, we report the experimental results. The Dijkstra's algorithm took $18\,\mathrm{s}$, $31\,\mathrm{s}$, $50\,\mathrm{s}$, and $64\,\mathrm{s}$ when $\gamma = 10, 15, 20, 25$, respectively.

In general, the results follow the previous trends. However, here it is interesting to see that the path dependability values with $\gamma = 10$ are only slightly worse than those with $\gamma \geq 15$. This is probably because the impact of γ saturates beyond a certain threshold. This suggests that we should keep γ as small as possible in order to minimize the required running time for building the whole structure. We cannot neglect the fact that when we decrease the visibility factor, the number of unfeasible paths could significantly increase. In other words, if the distribution of towers is very sparse, it can happen that many destinations could be unreachable due to missing edges. In general, when γ is small, the number of handovers is larger because the drone can "see" fewer towers, and therefore it necessarily has to connect with more antennas. For the same reason, the number of traversed edges is also larger. On the other hand, when γ increases, the drone tends to decrease the number of handovers and fly at lower heights due to the observations mentioned above: the lower the altitude, the lower the risk, and the stronger the signal strength.


(a) Roma.

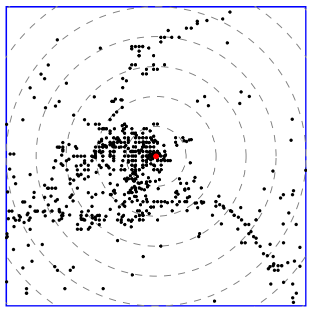

(b) Caltanissetta.

Fig. 10. Distribution of towers.

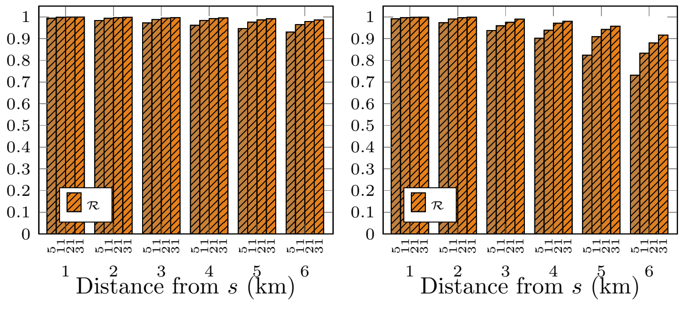
Another interesting aspect is the dependability, that shows a strange behavior. In fact, initially it smoothly decreases until $5\,\mathrm{km}$, then it starts to slightly increase. This depends on the tower distribution (e.g., see Figure 10) and on the way we group the destination vertices. Figure 10 shows that in Caltanissetta the number of towers is small ($|\mathcal{T}| = 624$) and its distribution is unbalanced, while in Roma there are really many towers ($|\mathcal{T}| = 21243$) evenly distributed. However, the way we group the results may be counter-intuitive. When we group the destination vertices within a certain annulus, we still consider the whole graph, and therefore the graph can go beyond the outer part of the annulus. So, it could happen that vertices at distance $5\,\mathrm{km}$ are only reachable going beyond that distance and coming back, due to the unbalanced distribution of towers that make impossible the flight of the drone, since there are uncovered areas. Therefore, according to the optimality property of the sub-paths of the shortest path algorithm, vertices at further distances will have a smaller cost (so, larger dependability) and are preferred as intermediate vertices by the Dijkstra's algorithm.

C. *Results with Corridors Strategy*

In this section, we report the experimental results of our proposed framework when the corridors are employed. The goal of this section is to compare, for the same source-destination pairs, the dependability value returned by MDP2 invoked on the complete graph and on the corridors. We trade the accuracy of the dependability value for lower computational complexity.

In this case, we simply evaluate the path dependability by proposing two different experiments: Roma ($\gamma = 5$ and varying $\mathcal{M} = 2, 7$) and Caltanissetta ($\mathcal{M} = 7$ and varying $\gamma = 10, 20$). The width of the corridors, which must be an odd number, is $\mathcal{C}_W = \omega = \{5, 11, 21, 31\}$ cells.

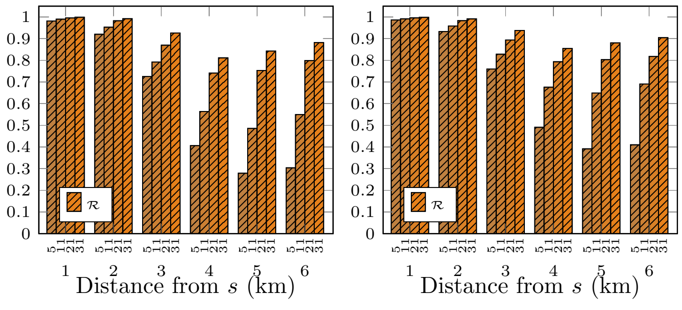
In these experiments, the number of unfeasible solutions can be really large with narrow corridors due to the fact that the straight or zig-zag corridors, from the source to the destination, can traverse areas without towers or no-fly zones. So, in order to present comparable results, we initially find 50 feasible random instances on the most constrained scenario (e.g., in Caltanissetta, it is with $\gamma = 10$ and $\omega = 5$), and keep these for the subsequent, and less constrained, scenarios.



(a) $\mathcal{M} = 2$. (b) $\mathcal{M} = 7$.

Fig. 11. Corridor results in Roma fixing $\gamma = 5$ and varying $\mathcal{M} = 2, 7$.

Figure 11 shows the experimental results for the city of Roma when corridors of different widths are used (i.e., $\omega = 5, 11, 21, 31$), with their corresponding values reported on the x-axis. In the y-axis, instead, we report the ratio $\mathcal{R} \in [0, 1]$ among the dependability of the corridor solution and the optimal one. In general, we can observe the same trend of the plots already seen in Figure 8 for increasing values of the distances from the source s. Obviously, the results in Figure 8 are optimal with respect to the whole box area, while the ones in Figure 11 are a little bit worse due to the fact that the drone flies only on a limited sub-area formed by narrow corridors. However, in the presented scenario the dependability obtained in the corridor solutions is always above the 70% of the optimal solution in $\mathcal{M} = 7$, and above the 95% in $\mathcal{M} = 2$. As expected, when we increase ω, the path dependability does not decrease. Although the path dependability is slightly affected, in the urban area, one can see that it is not significantly impacted. On the other hand, we were able to handle γ values up to 30 with a very reasonable amount of memory and time required.



(a) $\gamma = 10$. (b) $\gamma = 20$.

Fig. 12. Corridor results in Caltanissetta fixing $\mathcal{M}$=7 and varying γ=10, 20.

Finally, Figure 12 reports the corridor experiments on the rural city of Caltanissetta. Recall that this city is small, and the number of towers is very small (see Figure 10). So, when a narrow corridor is employed (e.g., $\omega = 5$), the path dependability can dramatically drop. In these cases, the dependability obtained in the corridor solutions when $\gamma = 10$ is at least 90% of the optimal solution for vertices within 2 km, and then decreases to around 30% of the optimal. Instead, when $\gamma = 20$ the results are a little bit better in the order of 40% of the optimal for the farthest vertices up to 6 km. Moreover, during the experiments, we noted many unfeasible solutions, especially for narrow corridors, due to the limited visibility of towers. In general, and as expected, the worst results appear when γ is small, ω is small, and the distance from the source increases. So, it is challenging to find a suitable path from the source to a far destination just relying on limited visibility and narrow corridors in the rural areas.

D. Towards a More Realistic Model

While our proposed framework is innovative, it currently exhibits several limitations which, by the way, allowed us to present concisely our framework. Our multi-layer model relies on a discretized environment, where drones fly at the center of cells, which does not always result in the best path with respect to path length compared to a straight line joining the start and destination points. We can incorporate a post-processing step by computing the Euclidean path within the selected cells or by connecting two adjacent cells with lines with different inclinations (but finite numbers) depending on the final destination. Another area for improvement lies in the visibility factor for drones. Currently, this is defined in a fixed and predetermined number of squares due to their compatibility with cell-based environments, but this implementation lacks realism as it is surely related to tower technologies. For example, we could compare dependability using 4G towers versus 5G towers, directional antenna versus omnidirectional antenna, and single tower versus multiple towers in a cell which possess different characteristics like range and bandwidth. In general, the obstacle layer can be enhanced in our framework by including real maps obtained from public services. The height of the terrain, which we neglected in our preliminary model, would also impact drone path planning. All these aspects will increase the graph's complexity. It then will urge to find a dynamic solution that extends the graph on demand.

VII. Conclusion and Next Steps

In this paper, we present a highly customizable novel graph-based multi-layer framework for planning in advance drone operations in BVLoS scenarios, based on real-world challenges. Our framework considers no-fly zones, obstacles, ground risk, and wireless communication infrastructure to ensure a reliable link between the drone and the operator. The customizability allows us to adjust various functions such as handover-success-rate, signal strength, ground risk, etc. The framework provides a polynomial graph construction that enables to efficiently solve the MDP2. In addition, we propose a lighter but sub-optimal solution based on corridors, which can guarantee satisfactory results. We also evaluate the applicability and feasibility of our framework on a real tower dataset. While our framework is novel, it currently exhibits several limitations, as outlined in Section VI-D. So, it is imperative to evaluate more realistic models in future research.

References

[1] A. Khochare, F. Betti Sorbelli *et al.*, "Improved algorithms for co-scheduling of edge analytics and routes for uav fleet missions," *IEEE/ACM Trans. on Networking*, vol. 32, no. 1, pp. 17–33, 2024.

[2] F. Betti Sorbelli, F. Corò, S. K. Das, L. Palazzetti, and C. M. Pinotti, "Drone-based bug detection in orchards with nets: A novel orienteering approach," *ACM Transactions on Sensor Networks*, 2024.

[3] K. Namuduri *et al.*, "Advanced air mobility: Research directions for comm., navigation, and surveillance," *IEEE Vehicular Technology Magazine*, vol. 17, no. 4, pp. 65–73, 2022.

[4] E. Denney, G. Pai, and M. Johnson, "Towards a rigorous basis for specific operations risk assessment of uas," in *2018 IEEE/AIAA 37th Digital Avionics Systems Conf. (DASC)*. IEEE, 2018, pp. 1–10.

[5] "Unmanned aircraft systems beyond visual line of sight aviation rulemaking committee," https://www.faa.gov/regulations_policies/rulemaking/committees.

[6] EASA, "UAS - Unmanned Aircraft Systems," www.easa.europa.eu.

[7] H. Rastgoftar, H. Emadi, and E. M. Atkins, "A finite-state fixed-corridor model for uas traffic management," *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[8] S. Primatesta, A. Rizzo, and A. la Cour-Harbo, "Ground risk map for unmanned aircraft in urban environments," *Journal of Intelligent & Robotic Systems*, vol. 97, no. 3, pp. 489–509, 2020.

[9] M. Milano, S. Primatesta, and G. Guglieri, "Air risk maps for unmanned aircraft in urban environments," in *Intl. Conf. on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2022.

[10] Y. Aydin, G. K. Kurt, E. Ozdemir, and H. Yanikomeroglu, "Authentication and handover challenges and methods for drone swarms," *IEEE Journal of Radio Frequency Identification*, 2022.

[11] G. Vallero, M. Deruyck, M. Meo, and W. Joseph, "Base station switching and edge caching optimisation in high energy-efficiency wireless access network," *Computer Networks*, vol. 192, 2021.

[12] D. Renga and M. Meo, "Can high altitude platforms make 6g sustainable?" *IEEE comm. Magazine*, 2022.

[13] F. Betti Sorbelli, P. Chatterjee, F. Coro, L. Palazzetti, and C. M. Pinotti, "A novel multi-layer framework for bvlos drone operation: A preliminary study," in *IEEE Conf. on Computer comm. Workshops*, 2023, pp. 1–6.

[14] S. Primatesta *et al.*, "A risk-based path planning strategy to compute optimum risk path for unmanned aircraft systems over populated areas," in *ICUAS*. IEEE, 2020, pp. 641–650.

[15] M. Ortlieb and F.-M. Adolf, "Rule-based path planning for unmanned aerial vehicles in non-segregated air space over congested areas," in *A39th Digital Avionics Systems Conf.* IEEE, 2020, pp. 1–9.

[16] P.-C. Shao, "Risk assessment for uas logistic delivery under uas traffic management environment," *Aerospace*, vol. 7, no. 10, p. 140, 2020.

[17] S. Balachandran *et al.*, "A path planning algorithm to enable well-clear low altitude uas operation beyond visual line of sight," in *Air Traffic Management Research and Development Seminar*. sn, 2017, p. 26.

[18] Y. Kim and J. Bae, "Risk-based uav corridor capacity analysis above a populated area," *Drones*, vol. 6, no. 9, p. 221, 2022.

[19] A. V. Savkin and H. Huang, "The problem of minimum risk path planning for flying robots in dangerous environments," in *2016 35th Chinese Control Conf. (CCC)*. IEEE, 2016, pp. 5404–5408.

[20] R. Amer, W. Saad, B. Galkin, and N. Marchetti, "Performance analysis of mobile cellular-connected drones under practical antenna configurations," in *ICC 2020*. IEEE, 2020, pp. 1–7.

[21] M. M. U. Chowdhury, P. Sinha, and I. Güvenç, "Handover-count based velocity estimation of cellular-connected uavs," in *2020 IEEE 21st Intl. Workshop on SPAWC*. IEEE, 2020, pp. 1–5.

[22] A. Fakhreddine, C. Bettstetter *et al.*, "Handover challenges for cellular-connected drones," in *Proceedings of the 5th Workshop on Micro Aerial Vehicle Networks, Systems, and Applications*, 2019, pp. 9–14.

[23] M. A. Zulkifley *et al.*, "Mobile network performance and technical feasibility of lte-powered unmanned aerial vehicle," *Sensors*, vol. 21, no. 8, p. 2848, 2021.

[24] K. Watanabe and M. Machida, "Outdoor lte infrastructure equipment (enodeb)," *Fujitsu Sci. Tech. J*, vol. 48, no. 1, pp. 27–32, 2012.

[25] M. A. Zulkifley *et al.*, "Mobile communications and parachute systems for safe beyond visual line of sight (bvlos) uav operation," in *2022 IEEE 6th Intl. Symposium on Telecom. Tech. (ISTT)*. IEEE, 2022, pp. 22–27.

[26] M. Ozger *et al.*, "Towards beyond visual line of sight piloting of uavs with ultra reliable low latency communication," in *IEEE global comm. Conf. (GLOBECOM)*. IEEE, 2018, pp. 1–6.

[27] R. Shrestha, R. Bajracharya, and S. Kim, "6g enabled unmanned aerial vehicle traffic management: A perspective," *IEEE Access*, vol. 9, pp. 91 119–91 136, 2021.

[28] S. Primatesta, G. Guglieri, and A. Rizzo, "A risk-aware path planning method for unmanned aerial vehicles," in *2018 Intl. Conf. on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2018, pp. 905–913.

[29] K. Dalamagkidis *et al.*, *On integrating unmanned aircraft systems into the national airspace system: issues, challenges, operational restrictions, certification, and recommendations*. Springer, 2009.

[30] A. Al-Hourani, S. Kandeepan, and S. Lardner, "Optimal lap altitude for maximum coverage," *IEEE Wireless comm. Letters*, vol. 3, no. 6, pp. 569–572, 2014.

[31] R. C. Johnson, "Antenna engineering handbook," *Antenna engineering handbook/Richard C. Johnson*, 1993.

[32] "Mobile signal strength recommendations," https://wiki.teltonika-networks.com/view/Mobile_Signal_Strength_Recommendations.

[33] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*. MIT press, 2022.

[34] U. Labs, "Opencellid - largest open database of cell towers & geolocation," https://opencellid.org.

Francesco Betti Sorbelli holds a Ph.D. in Computer Science from the Univ. of Florence, Italy (2018). He has been a postdoc researcher at Missouri Univ. of S&T (2020) and the Univ. of Perugia (2018, 2021-2022). Currently, he is an assistant professor at the Univ. of Perugia, focusing on algorithms, combinatorial optimization, and unmanned vehicles.

Punyasha Chatterjee received her B.Tech., M.Tech., and Ph.D. degrees in information technology from the Univ. of Calcutta, India, in 2003, 2005, and 2018, respectively. She is currently an Associate Professor at the School of Mobile Computing and Communication, Jadavpur Univ., India. Her research interests include wireless sensor networks, Internet of Things, distributed algorithms, and unmanned vehicles.

Federico Corò holds a Ph.D. in Computer Science from the Gran Sasso Science Institute. He worked as a postdoctoral researcher at various institutions, including La Sapienza Univ. of Rome, Missouri Univ. of S&T, and the Univ. of Perugia. Currently, he is an assistant professor at the Univ. of Padua, specializing in theoretical computer science, with a focus on combinatorial optimization, network analysis, and algorithm design.

Sajjad Ghobadi received the Bachelor and Master degrees in Computer Science from the Univ. of Tabriz and the Institute for Advanced Studies in Basic Sciences, Zanjan, Iran. He holds a Ph.D. in Computer Science from the Gran Sasso Science Institute, L'Aquila, Italy, and he is currently a postdoctoral researcher at the Univ. of Perugia, Italy, under the supervision of Cristina M. Pinotti. His research interests lie in algorithms design, combinatorial optimization, and algorithmic fairness.

Lorenzo Palazzetti received the Bachelor and Master degrees in Computer Science from the Univ. of Perugia, Italy, in 2018 and 2020, respectively. He completed his Ph.D. in April 2024 under the supervision of Prof. Cristina M. Pinotti at the Univ. of Florence, Italy. His research interests include design and analysis of algorithms, combinatorial optimization, unmanned vehicles.

Cristina M. Pinotti received the Master degree cum laude in Computer Science from the Univ. of Pisa, Italy, in 1986. In 1987-1999, she was Researcher with the CNR in Pisa. In 2000-2003, she was Associate Professor at the Univ. of Trento. Since 2004, she is a Full Professor at the Univ. of Perugia. Her current research interests include the design and analysis of algorithms, combinatorial optimization, emerging technologies, wireless sensor networks, and communication networks.