Surveillance Planning with Safe Emergency Landing Guarantee for Fixed-wing Aircraft

Petr Váňa*, Jakub Sláma, Jan Faigl

Czech Technical University in Prague, Faculty of Electrical Engineering, Department of Computer Science, Technicka 2, 166 27, Prague, Czech Republic

Abstract

In this paper, we study Emergency Landing Aware Surveillance Planning (ELASP) to determine a cost-efficient trajectory to visit a given set of target locations such that a safe emergency landing is possible at any point of the multi-goal trajectory. The problem is motivated to guarantee a safe mission plan in a case of loss of thrust for which it is desirable to have a safe gliding trajectory to a nearby airport. The problem combines computational challenges of the combinatorial multi-goal planning with demanding motion planning to determine safe landing trajectories for the curvature-constrained aerial vehicle. The crucial property of safe landing is a minimum safe altitude of the vehicle that can be found by trajectory planning to nearby airports using sampling-based motion planning such as RRT*. A trajectory is considered safe if the vehicle is at least at the minimum safe altitude at any point of the trajectory. Thus, a huge number of samples have to be evaluated to guarantee the safety of the trajectory, and an evaluation of all possible multi-goal trajectories is quickly computationally intractable. Therefore, we propose to utilize a roadmap of safe altitudes combined with the estimation of the trajectory lengths to evaluate only the most promising candidate trajectories. Based on the reported results, the proposed approach significantly reduces the computational burden and enables a solution of ELASP instances with tens of locations in units of minutes using standard single-core computational resources.

Keywords: Unmanned Aerial Vehicle, Surveillance Planning, Emergency Landing Guarantee

1. Introduction

Aerial surveillance missions are deployments of aerial vehicles to (repeatedly) visit a set of locations [1] and collect information about the areas of interests, e.g., in traffic [2] or environment monitoring [3]. The mission planning problem can be formulated as a problem to determine a cost-efficient multi-goal trajectory visiting the given set of target locations such that the trajectory satisfies the motion constraints of the used vehicle. In the case of fixed-wing aircraft, the problem can be formulated as the Dubins Traveling Salesman Problem (DTSP) [4] where the curvature-constrained trajectory is determined for Dubins vehicle [5]. Once a plan is determined, it can be expected that the vehicle will follow the planned trajectory. However, the aircraft can experience *Loss of Thrust* (LoT) [6], and it is then necessary to quickly determine a safe emergency landing trajectory [7].

In this paper, we study generalized surveillance missions where LoT is taken into account during multi-goal trajectory planning to guarantee the aircraft can safely land if LoT occurs during the mission. The proposed problem is called *Emergency Landing Aware Surveillance Planning* (ELASP) that stands to determine a cost-efficient multi-goal trajectory to visit a set of target locations such that the trajectory is safe and motion constraints of the vehicle are satisfied. The trajectory is considered to be safe; if the existence of a safe emergency landing trajectory is guaranteed at any point of the trajectory, see an example in

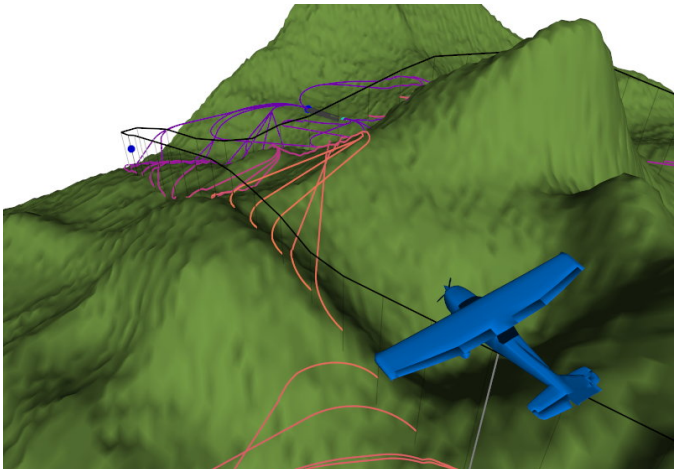


Figure 1: An example of safe trajectory (black line) for Cessna 172 airplane accompanied with possible emergency landing trajectories (colored lines according the actual altitude) [8].

*Corresponding author

Email addresses: vanapet1@fel.cvut.cz (Petr Váňa), slamajak@fel.cvut.cz (Jakub Sláma), faiglj@fel.cvut.cz (Jan Faigl)

Fig. 1. Thus, for fixed-wing aircraft, the addressed problem combines combinatorial challenges of the curvature-constrained multi-goal trajectory planning with planning safe emergency landing trajectories.

Several approaches to the DTSP have been proposed [9] that includes sampling-based methods [10], quick heuristic algorithms [11], and decoupled approaches, where the sequencing part is determined as a solution of the Euclidean TSP that is followed by determination of the curvature-constrained multi-goal trajectory [12]. However, it is necessary to consider 3D trajectories for an emergency landing, and therefore, the approaches for the DTSP have to be generalized to 3D trajectory parametrization, e.g., using Bézier curves [13] or the Dubins Airplane model [14, 15] as in [16]. Regarding planning emergency landing trajectories, the existing point-to-point trajectory generation approaches can be utilized, e.g., the Dubins Airplane model, but the problem has also been explicitly addressed in [17, 18].

Although surveillance planning and emergency landing have been addressed in the literature separately, the computationally challenging ELASP problem, to the best of the authors' knowledge, has not yet been adequately addressed. Therefore, we address the introduced ELASP problem by the proposed $\text{ELASP}_{\text{BASE}}$ approach. This baseline solution is further improved by reducing the computational requirements by using an estimation of the trajectory lengths and lazy evaluation technique to determine only the most promising safe trajectories in the so-called $\text{ELASP}_{\text{LAZY}}$ algorithm.

In the baseline $\text{ELASP}_{\text{BASE}}$ method, a pre-generated roadmap of possible landing trajectories for the mission area is determined using the RRT*-based algorithm. Then, each candidate multi-goal trajectory is examined for the minimum safe altitude during the multi-goal trajectory planning, which is computationally demanding. We propose to address the computational challenges arising from the multi-goal planning by trajectory cost estimation to evaluate only the most promising candidate trajectories in $\text{ELASP}_{\text{LAZY}}$, which is further speeded up by utilizing a pre-computed roadmap of safe altitudes.

The employed cost estimation of $\text{ELASP}_{\text{LAZY}}$ follows the idea of [19] to reduce the number of motion planning queries by using Euclidean distances for welding robots visiting a finite set of locations, where the multi-goal trajectory consists of a sequence of point-to-point trajectories. However, in the addressed ELASP problem, we need to generate safe trajectories with the emergency landing guarantee. Furthermore, the fixed-wing vehicle is curvature-constrained, and therefore, the vehicle heading angle at each location is considered to guarantee the final multi-goal trajectory is smooth and feasible. Therefore Euclidean distance employed in [19] is not a proper estimation for ELASP. Since each candidate trajectory has to be examined for the minimum safe altitude, which is computationally demanding, selecting the most promising candidate trajectories based on the easy to compute tight lower bounds on the trajectory cost might have a considerable impact on the overall performance. We propose to utilize the so-called terrain trajectory as such a lower bound. Based on the herein reported results, it significantly improves the performance, and together with a pre-computed roadmap of safe altitudes, they enable a solution of the ELASP problem using a single core of the standard desktop computer, which is promising value for future deployment on real aircraft.

The contributions of the presented work are considered as follows.

- Introduction of the ELASP problem with a baseline solution based on relatively straightforward $\text{ELASP}_{\text{BASE}}$, which is, however, impractically computationally demanding.
- $\text{ELASP}_{\text{LAZY}}$ with significantly decreased computational requirements compared to the baseline $\text{ELASP}_{\text{BASE}}$, which requires tens of minutes instead of hours for tens of target locations and the same single-core computational environment.
- Necessary condition for a penalty function of the relaxed solution in the case the target locations to be visited are at the altitude that does not allow safe emergency landing to the nearby airports.
- Results on empirical evaluation of the ELASP algorithms performance.

The rest of the paper is organized as follows. An overview of the related work on planning multi-goal and emergency landing trajectories is summarized in the following section. The ELASP problem is formally introduced in Section 3. The construction of the supporting roadmap of emergency landing trajectories is described in Section 4. The baseline $\text{ELASP}_{\text{BASE}}$ algorithm is described in Section 5. The improved $\text{ELASP}_{\text{LAZY}}$ is presented in Section 6. The necessary condition on the penalty function is detailed in Section 7. Results on the empirical evaluation of ELASP solutions are reported in Section 8. The final concluding remarks are in Section 9.

2. Related work

The introduced ELASP is a generalization of surveillance planning to visit a set of given target locations [11, 13, 20], where it is explicitly requested to guarantee the existence of safe emergency landing for any point of the surveillance multi-goal trajectory for the fixed-wing aircraft. Thus, ELASP is related not only to combinatorial multi-goal planning with curvature-constrained vehicles but also emergency landing to plan safe gliding trajectory to a nearby airport. Therefore, existing approaches of these related fields are briefly summarized in the rest of this section to justify the novelty of ELASP and proposed solutions.

The Dubins vehicle model [5] can be utilized for trajectory planning for fixed-wing aerial vehicles with curvature-constrained trajectories. Dubins vehicle is advantageous because of a closed-form solution for the point-to-point trajectory of the 2D plane with the minimum turning radius and defined vehicle heading at the initial and terminal locations. Nevertheless, the vehicle changes its altitude during the emergency landing, and therefore, a 3D curvature-constrained trajectory is needed. The 3D extension of the formerly 2D Markov-Dubins problem [5] is studied in [21], where the authors proved the necessary conditions for the optimal path. However, a closed-form solution is not available so far.

The authors of [22] proposed a heuristic method for the 3D trajectory generation that can connect arbitrary configurations of the vehicle, considering the constrained pitch angle. In [14], such a model of the 3D curvature-constrained trajectory is called the Dubins Airplane model. Necessary implementation details of the trajectory generation method are addressed in [15] considering the kinematics of a real fixed-wing vehicle, which makes the method computationally usable in multi-goal trajectory planning [16, 23]. Furthermore, the computation of trajectories for the Dubins Airplane model has been recently improved in [24], where the authors also present quickly to compute tight lower bound estimates. Although Dubins Airplane model simplifies a real aircraft dynamics, it is a sufficient model for path planning purposes [15, 25]. Moreover, a trajectory consisting of several Dubins maneuvers may introduce discontinuity in pitch angle between consecutive maneuvers as the pitch angle could be changed very quickly [15]. As an alternative to these constructive methods, seventh order Bézier curves are utilized in [26] that enable to change the speed of the vehicle, but is computationally demanding, which is also the case of low order Bézier curves utilized in [13].

Having a method to determine a feasible (and eventually optimal) point-to-point trajectory connecting two configurations of the vehicle corresponding to the target locations in 3D, the multi-goal trajectory planning can be addressed as a problem to determine the optimal sequence of visits to the given set of target locations. The introduced ELASP is a variant of the surveillance planning [11] that stands to determine the sequence of visits together with the most suitable configurations at which the given target locations (or areas) are visited such that the overall cost of the final trajectory is minimized. Regarding the existing computationally efficient point-to-point trajectory planning, the ELASP problem can be seen as the 3D variant of the Dubins Traveling Salesman Problem (DTSP) [4] where the 2D Dubins vehicle model is substituted by the Dubins Airplane model [16].

The existing DTSP approaches can be roughly divided into decoupled and sampling-based methods [9]. The decoupled methods separate the combinatorial part (to determine the optimal sequence) from the continuous optimization of the optimal configurations to visit the locations, e.g., by a solution of the Euclidean TSP followed by determination of the heading angles at the locations. For example, the headings can be found by a simple heuristic such as Alternating Algorithm [4] or by continuous optimization using tight lower bound estimation of the Dubins multi-goal trajectory [12].

In sampling-based methods, possible heading angles at each target location are sampled into a finite set of configurations [27], and the problem is formulated as an instance of the Generalized Asymmetric TSP. Then, the instance can be transformed by the Noon-Bean transformation [28] to an instance of the regular TSP that can be solved using existing solvers, e.g., optimally by Concorde [29] or heuristically using LKH [30]. Alternatively, soft-computing techniques based on genetic [31], memetic [32], or unsupervised learning [33] algorithms can be utilized to address both combinatorial and continuous optimization parts of the DTSP at the same time.

The main difference of the ELASP problem to the DTSP is in the requirements on the emergency landing trajectory for any point of the planned surveillance trajectory to guarantee safe landing in the case of LoT. For an aircraft modeled as Dubins vehicle, planning the emergency landing trajectory can be based on the determination of the overall altitude loss of the shortest Dubins maneuver [5] from a particular location of the vehicle to the closest landing site. Such a concept utilizing the Dubins Airplane model is used in [17] to compute gliding trajectories for the accident on the Hudson river [7]. However, neither obstacles nor altitude of the terrain in the vicinity of the landing site is considered in [17]. This issue has been addressed by A*-based algorithm [34], evolutionary approach [35], but also using asymptotically optimal RRT* algorithm in [36]. Besides, the emergency landing trajectory can be further generalized by considering the influence of the wind [18, 37], which, however, does not address multi-goal planning and nearby terrain.

3. Problem Statement

The studied *Emergency Landing Aware Surveillance Planning* (ELASP) problem stands to find a cost-efficient safe closed-loop trajectory $\mathcal{R}$ for a fixed-wing vehicle to visit n given target locations $S = \{s_1, s_2, \ldots, s_n\}$, $s_i \in \mathbb{R}^3$, while motion constrains of the fixed-wing aircraft are satisfied. A trajectory is considered to be safe, if the vehicle can land safely to a nearby airport in a case of the total *Loss of Thrust* (LoT); thus, there exists a safe gliding trajectory to the airport for any point of $\mathcal{R}$. The ELASP combines the three following challenges that should all be addressed simultaneously as they are mutually dependent.

The first challenge is to **determine a sequence of visits** $\Sigma = \{\sigma_1, \ldots, \sigma_n\}$, $\sigma_i \in \{1, \ldots, n\}$ to the given locations S such that the total travel cost of $\mathcal{R}$ is minimized. Finding Σ is a sequencing part of ELASP that can be seen as a variant of the TSP where the travel costs are given by

the travel cost of individual safe trajectories connecting the locations S.

The second challenge is to **satisfy motion constraints** of the fixed-wing vehicle that is modeled as the Dubins Airplane model [14]. The state q of the vehicle combines 3D position $(x, y, z) \in \mathbb{R}^3$, heading angle $\theta \in \mathbb{S}$, and pitch angle $\psi \in \mathbb{S}$, i.e., $q = (x, y, z, \theta, \psi)$. The vehicle dynamics can be expressed as

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\theta} \end{bmatrix} = v \begin{bmatrix} \cos\theta\cos\psi \\ \sin\theta\cos\psi \\ \sin\psi \\ u_\theta\,\rho^{-1} \end{bmatrix}, \tag{1}$$

where ρ stands to the minimum turning radius, and the heading angle θ is controlled by $u_\theta \in [-1, 1]$. Following [14], it is assumed that the time constant of changing the pitch angle ψ is significantly lower than for the heading angle, and thus ψ is considered as the control input and abrupt changes of ψ are allowed within the given interval

$$\psi \in [\psi_{\min}, \psi_{\max}]. \tag{2}$$

A natural condition for the aircraft is to fly above the terrain (or obstacles). Based on the vehicle state (1), the configuration space is considered as $\mathcal{C} = \mathbb{R}^3 \times \mathbb{S}^2$ and the free part of $\mathcal{C}$ is defined as

$$\mathcal{C}_{free} = \{q = (x, y, z, \theta, \psi) \,|\, q \in \mathcal{C}, z > \mathcal{T}_{\text{alt}}(x, y)\}, \tag{3}$$

where $\mathcal{T}_{\text{alt}} : \mathbb{R}^2 \to \mathbb{R}$ is a function of the terrain altitude (or obstacles) at the given position (x, y). Since it can be assumed the aircraft always operates above the terrain and it is not allowed to fly under obstacles, e.g., under a bridge, w.l.o.g., we can model the world as a terrain elevation map.

The closed-loop trajectory $\mathcal{R}$ is formed by the trajectories $\mathcal{R} = \{\mathcal{R}_1, \ldots, \mathcal{R}_n\}$ that are connected at visiting the configurations $Q = \{q_{\sigma_1}, q_{\sigma_2} \ldots, q_{\sigma_n}\}$, corresponding for (x, y) with the locations S, in the order defined by the sequence Σ. Thus, the i-th trajectory starts at the visiting configuration q_{σ_i} located at s_{σ_i} or directly above it in the case of an insufficient altitude for a safe emergency landing (detailed in the following text). The safe trajectory has to guarantee the vehicle is always moving above the terrain. Therefore, each trajectory of $\mathcal{R}$ has to be in C_{free}. All the trajectories are normalized such that $\mathcal{R}_i(0)$ be the initial and $\mathcal{R}_i(1)$ be the final configuration of the i-th trajectory segment, and thus $\mathcal{R}_i : [0, 1] \to \mathcal{C}_{free}$.

The third challenge is to **guarantee safety** of the found trajectory $\mathcal{R}$. Let assume there are m given landing sites with touchdown configurations $\Xi = \{\xi_1, \ldots, \xi_m\}$. Then, an emergency landing trajectory $\Gamma : [0, 1] \to \mathcal{C}_{free}$ must exits from any point τ of each trajectory $\mathcal{R}_i \in \mathcal{R}$; thus, there must be Γ such that $\Gamma_\tau(0) = \mathcal{R}_i(\tau)$ for all $\tau \in [0, 1]$. Furthermore, the emergency landing trajectory Γ ends at a particular landing site j or directly above it, and $\Gamma(1)$ is therefore from the set $\widehat{\xi_j}$ of the configurations above the landing site ξ_j. The landing trajectory is allowed to end above the selected landing configuration because it is assumed the vehicle can decrease its altitude relatively quickly by a specific maneuver type, even in the case of LoT [36]. Therefore, $\Gamma(1)$ above the landing site maximizes safety as it keeps the vehicle at the highest altitude possible. Besides, each landing trajectory is strictly above the terrain, and it includes possible altitude loss for the case of LoT.

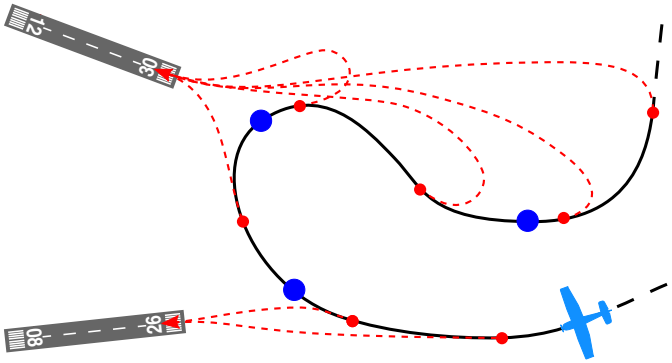


Figure 2: An example of ELASP trajectory (black line) that visits target locations (blue dots). The planned trajectory contains decision points (red dots) from which the pilot can decide to execute the emergency landing (red dotted line) to an airport. To guarantee a safe landing is possible from any point along the planned trajectory, it is necessary to guarantee maintaining high enough altitude to reach the next decision point safely, i.e., reaching the decision point or perform an emergency landing even in the case of LoT.

In the problem formulation, it is also desirable to address the situation when the safety requirement on the trajectory may not allow the vehicle to precisely visit the locations S at the defined altitude because no feasible solution exists if the locations are too low to guarantee safe emergency landing. For this reason, it is suitable to relax the precise visitation of the location $s_i \in S$ at the specified altitude, and s_i is considered to be visited if the visiting configuration is directly above the target location. For such a relaxed visitation, the safe trajectory can be determined for a sufficiently high altitude. Since we primarily aim to find the trajectories visiting the defined locations S, an additional penalty $\mathcal{P} : \mathbb{R} \to \mathbb{R}$ is introduced to the objective function to minimize altitude difference Δ between the visiting configuration q_i and the corresponding target location s_i. The penalty function is selected to be proportional to Δ with the penalty multiplier β with the positive penalty for non-negative Δ (the case the visiting configuration is above the target location) and an infinite penalty for negative Δ

$$\mathcal{P}(\Delta) = \begin{cases} \beta\Delta & \text{if} \quad \Delta \geq 0, \\ \infty & \text{if} \quad \Delta < 0. \end{cases}. \tag{4}$$

All three challenges are integrated into the *Emergency Landing Aware Surveillance Planning (ELASP)* that makes the problem very challenging. An example

ELASP trajectory visiting the given location is depicted in Fig. 2 together with possible emergency landing trajectories from the so-called decision points that are determined by the employed sampling-based trajectory planning of the proposed solution. The ELASP problem is formulated as the optimization Problem 3.1.

Problem 3.1 Emergency Landing Aware Surveillance Planning (ELASP) problem**.**

$$\min_{\mathcal{R},\Sigma,Q} \sum_{i=1}^{n} \left(\mathcal{L}(\mathcal{R}_i) + \mathcal{P}\left([q_{\sigma_i}]^z - [s_i]^z\right)\right) \tag{5}$$

$$\text{s.t.} \quad \mathcal{R}_{i+1}(0) = \mathcal{R}_i(1), \qquad \forall i \in \{1,2,\ldots,n\}, \tag{6}$$

$$\mathcal{R}_i(0) = q_{\sigma_i}, \qquad \forall i \in \{1,2,\ldots,n\}, \tag{7}$$

$$[s_i]^{xy} = [q_i]^{xy}, \qquad \forall i \in \{1,2,\ldots,n\}, \tag{8}$$

$$\forall d \in [0,1],\ \exists \Gamma, j: \quad \Gamma(0) = \mathcal{R}_i(d) \wedge \Gamma(1) \in \widehat{\xi}_j, \tag{9}$$

$$\mathcal{R}_i \text{ meets (1) and (2)}, \qquad \forall i \in \{1,2,\ldots,n\}, \tag{10}$$

$$\forall \check{q} \in R_i: \ \check{q} \in \mathcal{C}_{free} \qquad \forall i \in \{1,2,\ldots,n\}, \tag{11}$$

where $\mathcal{L}(\mathcal{R}_i)$ denotes the length of the i-th trajectory, $[\cdot]^{xy}$ denotes a projection of the configuration to 2D position in xy plane, and $[\cdot]^z$ the projection in z-axis. Notice $\sigma_{n+1} \triangleq \sigma_1$, and $\mathcal{R}_{n+1} \triangleq \mathcal{R}_1$, respectively, to simplify the notation for the closed-loop trajectory.

The final trajectory $\mathcal{R}$ is closed-loop and continuous, which is ensured by (6). The existence of the safe emergency landing trajectory at any point of the final trajectory $\mathcal{R}$ is assured by (9).

4. Roadmap of Possible Emergency Landing Trajectories and Determination of Safe Altitude

One of the computational challenges in the studied ELASP is to determine the safe altitude for a specific configuration of the vehicle because it is needed in the determination of a single safe trajectory. Moreover, a vast number of safe trajectories need to be determined in multi-goal planning. The challenge is addressed by a pre-compute roadmap of possible landing trajectories. The roadmap is utilized to retrieve a safe altitude for a particular point of the examined trajectory that is used to update the altitude profile of the trajectory to guarantee it is safe. A dense roadmap of possible landing trajectories is constructed by the RRT*-based algorithm adopted from [36] that is summarized in Algorithm 1.

The construction is based on growing a tree for each landing site, but all the trees are connected into a single graph to compute the complete roadmap efficiently. Contrary to [36], where informed randomized sampling is utilized to provide safe emergency landing trajectory from the current position of the vehicle, uniform sampling is used for ELASP, because the vehicle position is unknown.

Algorithm 1: RRT*-based construction of possible emergency landing trajectories (adopted from [36])

Input: $\Xi = \{\xi_1, \ldots, \xi_m\}$ – Set of the landing sites
Input: $\mathcal{T}_{\text{alt}}$ – Altitude of the terrain (or obstacles)
Input: t_{plan} – Maximum time to create the roadmap
Output: G – Roadmap of landing trajectories
Output: $\mathcal{A}$ – Minimum safe altitudes for G

Function `SafeLandingMap()`:
  $G \leftarrow \{\boldsymbol{V} \leftarrow \Xi, \boldsymbol{E} \leftarrow \emptyset\}$
  $\mathcal{A}(\xi_i) \leftarrow \mathcal{T}_{\text{alt}}(\xi_i),\ \forall \xi_i \in \Xi$
  while $t < t_{plan}$ **do**
    $\tilde{q}_{\text{rand}} \leftarrow$ `SampleUniform()`
    $\tilde{q}_{\text{nearest}} \leftarrow$ `Nearest`$(\tilde{q}_{\text{rand}}, G)$
    $\tilde{q}_{\text{new}} \leftarrow$ `Steer`$(\tilde{q}_{\text{nearest}}, \tilde{q}_{\text{rand}})$
    $Q_{\text{n}} \leftarrow$ `Near`$(\tilde{q}_{\text{new}}, G)$
    $\tilde{q}_* \leftarrow \text{argmin}_{\tilde{q}_{\text{n}} \in Q_{\text{n}}} \left[\mathcal{A}(\tilde{q}_{\text{n}}) + \mathcal{H}(\tilde{q}_{\text{new}}, \tilde{q}_{\text{n}})\right]$
    $\mathcal{A}(\tilde{q}_{\text{new}}) \leftarrow$
      $\max\left[\mathcal{T}_{\text{alt}}(\tilde{q}_*, \tilde{q}_{\text{new}}), \mathcal{A}(\tilde{q}_*) + \mathcal{H}(\tilde{q}_{\text{new}}, \tilde{q}_*)\right]$
    $\boldsymbol{V} \leftarrow \boldsymbol{V} \cup \{\tilde{q}_{\text{new}}\}$; $\boldsymbol{E} \leftarrow \boldsymbol{E} \cup \{(\tilde{q}_*, \tilde{q}_{\text{new}})\}$
    $G \leftarrow$ `Rewire`(Q_{n}, G)
  return $G, \mathcal{A}$

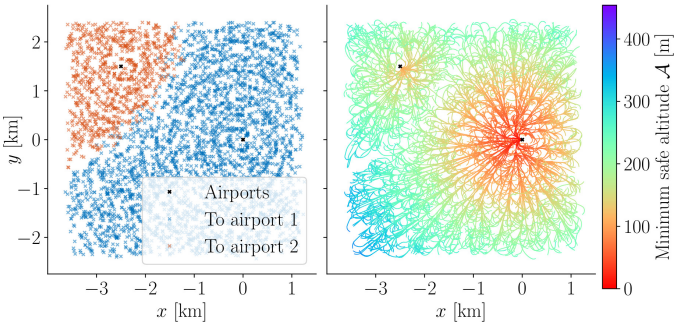


Figure 3: An example of the 2D projection of the roadmap generated by Algorithm 1 for two unidirectional landing sites at different altitudes without obstacles. Each sample of the roadmap is represented as small cross (left) with the color determining the corresponding airport for safe emergency landing. The minimum safe altitude (right) includes both altitude loss of the landing trajectory and altitude of the selected airport. The first airport is located at the coordinates $(0, 0, 0)$ km heading east (90°) and the second is located at $(-2.5, 1.5, 0.1)$ km heading north-west (315°).

The RRT* is employed to determine the minimum altitude loss for possible landing trajectories. Based on the model of the gliding trajectory proposed in [36], the configuration space of the vehicle is simplified to the 2D position (x, y) and the corresponding heading angle θ. Thus, the simplified configuration is $\tilde{q} = (x, y, \theta) \in SE(2)$, i.e., the simplified configuration space is $\tilde{\mathcal{C}} = SE(2)$, which significantly reduces the computational burden. The altitude of $\tilde{q}$ is then considered as the minimum altitude for the safe emergency landing that is denoted as the function $\mathcal{A}: \tilde{\mathcal{C}} \to \mathbb{R}$. The particular minimum altitude $\mathcal{A}$ is influenced by the altitude of the selected landing site, altitude loss $\mathcal{H}: \Gamma \to \mathbb{R}$ of the corresponding landing trajectory Γ, and the altitude of the terrain. An example of the gener-

ated roadmap for two landing sites is visualized in Fig. 3 and the evolution of the roadmap is depicted in Fig. 4.

The roadmap G is iteratively constructed for a given computational time t_{plan}, where at each iteration, a random configuration is uniformly sampled over the whole planning area (Line 5, Algorithm 1). Then, the nearest configuration in the current roadmap G is found by `Nearest`(), and a connection to the randomly selected node is determined by 2D Dubins maneuver in the `Steer`() procedure. Further, a part of the trajectory longer than the steer constant is cut off, and a new configuration at the maneuver's end is returned (Line 7, Algorithm 1). Possible parent nodes for the new configuration $\tilde{q}_{\text{new}}$ are determined by `Near`(). The best parent node is selected according to the overall altitude loss to the landing site. Finally, an attempt to optimize the graph G is performed in `Rewire`() by re-connecting existing edges to lower the minimum safe altitude.

Having the roadmap G, a minimum safe altitude at some configuration $\tilde{q}_{\text{act}}$ can be retrieved by trying to connect $\tilde{q}_{\text{act}}$ to the closest configurations Q_{near} with the corresponding landing trajectory that starts at $\tilde{q}_i \in Q_{\text{near}}$. The trajectory with the minimum required altitude is then returned. The query on the minimum safe altitude is summarized in Algorithm 2.

Algorithm 2: Retrieve the minimum safe altitude

Input: G – Roadmap of landing trajectories
Input: $\mathcal{A}$ – Minimum safe altitudes for graph nodes
Input: $\tilde{q}_{\text{act}}$ – Simplified configuration to query
Output: $\mathcal{A}(\tilde{q}_{\text{act}})$ – Minimum safe altitude for $\tilde{q}_{\text{act}}$

1 $Q_{\text{near}} \leftarrow \texttt{Near}(\tilde{q}_{\text{act}}, G)$
2 $\tilde{q}_* \leftarrow \text{argmin}_{\tilde{q}_i \in Q_{\text{near}}} [\mathcal{A}(\tilde{q}_i) + \mathcal{H}(\tilde{q}_{\text{act}}, \tilde{q}_i)]$
3 $\mathcal{A}(\tilde{q}_{\text{act}}) \leftarrow \max [\mathcal{T}_{\text{alt}}(\tilde{q}_*, \tilde{q}_{\text{act}}), \mathcal{A}(\tilde{q}_*) + \mathcal{H}(\tilde{q}_{\text{act}}, \tilde{q}_*)]$

The pre-computed roadmap G helps to reduce the computational burden, but the examined trajectory has to be densely sampled to guarantee it is safe. Hence, the number of examined trajectories significantly affects the computational requirements, and thus the maximal size of the ELASP instance can be solved in a reasonable computational time. Therefore, reducing the number of examined trajectories can significantly improve the scalability of the ELASP solution.

5. Baseline ELASP$_{\text{BASE}}$ Solver for Emergency Landing Aware Surveillance Planning (ELASP)

The proposed baseline solution of ELASP employs the pre-computed roadmap G to examine a trajectory is safe by querying the minimum safe altitude for particular samples of the examined trajectory. The challenging ELASP is decomposed into five subproblems that are solved separately to make the problem computationally feasible. The baseline ELASP$_{\text{BASE}}$ is summarized in Algorithm 3 and the subproblems are as follows.

1. A dense roadmap G of possible landing trajectories with the associated minimal altitudes $\mathcal{A}$ to each node of G is generated by Algorithm 1 to enable the minimum safe altitude queries for any location of the vehicle in the planning area (Algorithm 3, Line 1).
2. Possible vehicle heading angles at each target location are uniformly sampled, and a vector of all configurations W is created (Algorithm 3, Line 2).
3. Safe trajectories $\mathcal{R}_{\text{all}}$ are computed between all sampled configurations W using the pre-computed G and $\mathcal{A}$ (Algorithm 3, Lines 3–8).
4. Trajectories $\mathcal{R}_{\text{all}}$ are used to create an instance of the GATSP to determine the sequence of visits using an existing TSP-like solver (Algorithm 3, Lines 9), e.g., Concorde [29] or LKH [38] by the transformation to the TSP using Noon-Bean transformation [28].
5. For the determined sequence of visits, the corresponding trajectories are connected into the final trajectory, and any altitude discontinuities are removed if occur (Algorithm 3, Lines 10).

The individual steps are further detailed in the rest of this section to provide a complete description of the algorithm supporting its reimplementation.

Algorithm 3: ELASP$_{\text{BASE}}$ – Baseline solver of ELAPS problem

Input: $\Xi = \{\xi_1, \ldots, \xi_m\}$ – Set of the landing sites
Input: $\mathcal{T}_{\text{alt}}$ – Altitude of the terrain (or obstacles)
Input: $S = \{s_1, s_2, \ldots, s_n\}$ – Set of target locations
Input: k – Number of heading samples
Output: $\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_n\}$ – Sequence of visits
Output: $\mathcal{R} = \{\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_n\}$ – Final trajectory

1 $G, \mathcal{A} \leftarrow \texttt{SafeLandingMap}(\Xi, \mathcal{T}_{\text{alt}})$ `// call Alg. 1`
2 $W \leftarrow \texttt{SampleHeading}(S, k)$
3 $\mathcal{R}_{\text{all}} \leftarrow \emptyset$
4 **foreach** $q_i^a \in W$ **do**
5 **foreach** $q_j^b \in W$ **do**
6 **if** $i = j$ **then**
7 **continue**
8 $\mathcal{R}_{\text{all}} \leftarrow \mathcal{R}_{\text{all}} \cup \texttt{SafeT}(q_i^a, q_j^b, G, \mathcal{A})$
9 $\Sigma, \Sigma', \mathcal{R}' \leftarrow \texttt{SolveGATSP}(\mathcal{R}_{\text{all}})$
10 $\mathcal{R} \leftarrow \texttt{AssurePitchLimits}(\mathcal{R}')$

In the first step, a dense roadmap of possible emergency landing trajectories is determined by Algorithm 1 to enable queries on the minimum safe altitude for the specific configuration of the vehicle. Thus, only a single roadmap G is created for all queries to reduce the computational burden since otherwise, it would not be computationally tractable to compute a landing trajectory for each configuration separately by the RRT*algorithm.

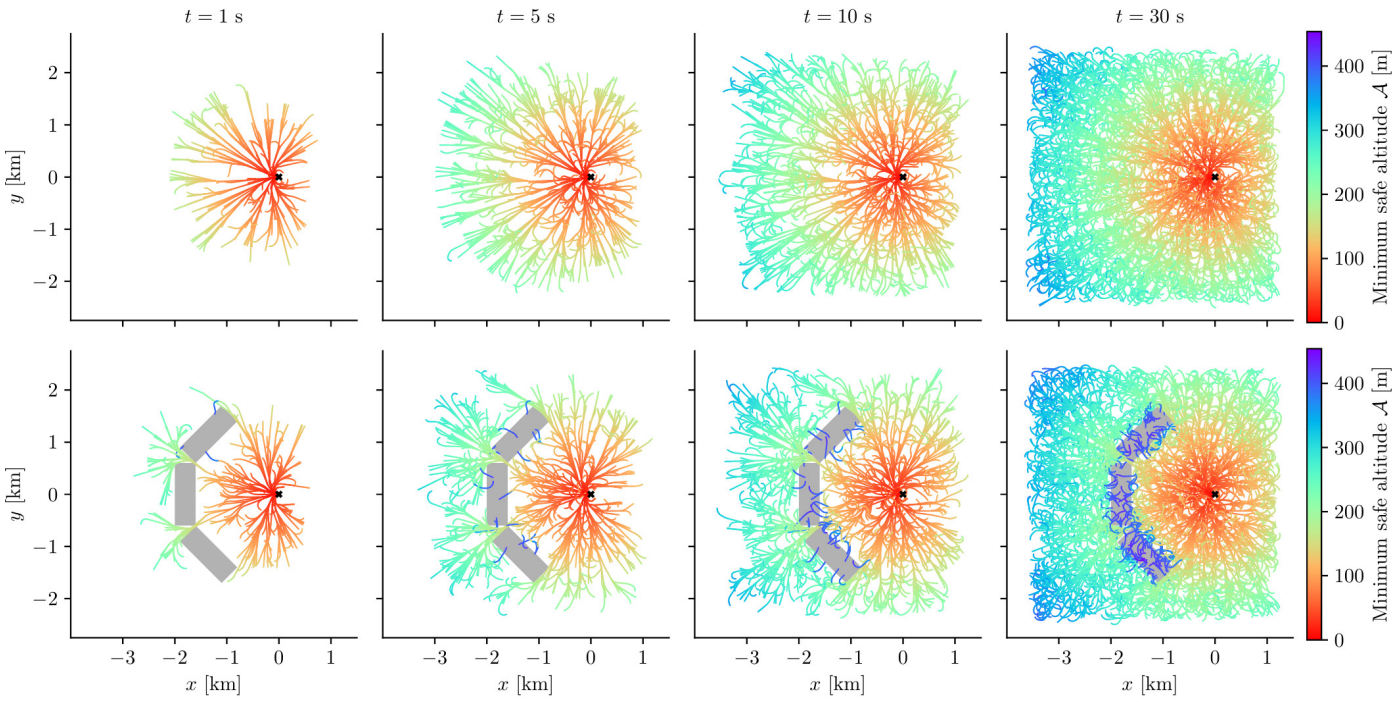


Figure 4: An evolution of the roadmap generated by Algorithm 1 over time t with a single unidirectional runway surrounded by a flat terrain. Two scenarios are compared: without any obstacle (top), and with three 450 m high rectangular obstacles. The color of the landing trajectories represents the minimum altitude for a safe emergency landing.

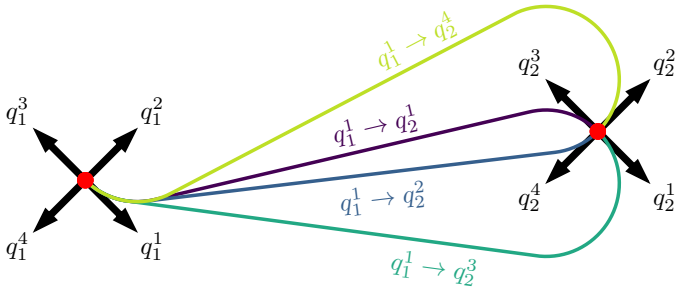


Figure 5: Example of the sampled heading angles for $k = 4$ and two target locations s_1 and s_2 with all connections from created configuration q_1^1 of s_1 to all configurations of s_2 $(q_2^1, \ldots, q_2^4)$.

Once the roadmap G is constructed, safe trajectories between the target locations S are computed. Since the fixed-wing vehicle is curvature-constrained, particular trajectories between the configurations depend on the vehicle heading angle at the locations. Therefore, possible heading angles are uniformly sampled for each location and the procedure `SampleHeading`() creates k configurations for each of n target locations that are all stored in a single vector of configurations

$$W = \bigcup_{i=1}^{n} \{q_i^1, \ldots, q_i^k\}. \tag{12}$$

The number of heading samples k influences the size of the GATSP instances; however, similarly to instances of the DTSP, too high number does not necessarily improve solution quality [11], the influence of k to the algorithm performance is reported in Section 8. Notice that safe trajectories are computed between the configurations W, except those corresponding to the same target locations, similar to the sampling-based solution of the DTSP [27], see Fig. 5.

Determination of all safe trajectories $\mathcal{R}_{\text{all}}$ is very demanding because a single safe trajectory needs to be sampled, and the safe altitude has to be determined by Algorithm 2 for each such a sample in the procedure `SafeT`(). Furthermore, the number of safe trajectories quickly grows with the number of locations n and also with the number of headings k. In the ELASP$_{\text{BASE}}$ algorithm, $k^2(n^2 - n)$ safe trajectories are determined in the total, and the computation thus becomes quickly intractable.

The procedure `SafeT`() first connects two selected configurations using the Dubins Airplane model [14] to meet the minimum turning radius constraint. However, such a trajectory might not be safe, because in the case the vehicle is at too low altitude, it is not guaranteed there is a safe emergency landing trajectory for any point of the trajectory. Therefore, the trajectory generated by the Dubins Airplane model is discretized into simplified configurations $\tilde{\mathcal{Q}} = \{\tilde{q}_1, \tilde{q}_2, \ldots, \tilde{q}_{\text{end}}\}$ uniformly with the step d_{step}. The sampled configurations of the trajectory are further referred to as the decision points. Then, the minimum safe altitude is determined for each decision point independently using Algorithm 2.

The existence of a safe landing trajectory has to be guaranteed for any point of the trajectory and not only for the decision points. Hence, the safe altitude $\mathcal{A}_{\text{safe}}$ is determined, such that the altitude loss $\mathcal{H}$ of the trajectory

between two decision points is added to the minimum altitude of the previous decision point. Further, the minimum/maximum pitch angle $\psi_{\min/\max}$ is utilized to limit the slope of the trajectory. The determination of the safe altitude $\mathcal{A}_{\text{safe}}$ can be expressed as

$$\mathcal{A}_{\text{safe}}\left(\tilde{q}_j\right) = \max \begin{bmatrix} \mathcal{A}\left(\tilde{q}_{j+1}\right) + \mathcal{H}\left(\tilde{q}_j, \tilde{q}_{j+1}\right) \\ \mathcal{A}_{\text{safe}}\left(\tilde{q}_{j-1}\right) + \tan(\psi_{\min})\, d_{j-1,j} \\ \mathcal{A}_{\text{safe}}\left(\tilde{q}_{j+1}\right) - \tan(\psi_{\max})\, d_{j,j+1} \end{bmatrix}, \quad (13)$$

where $d_{j,j+1}$ is the Euclidean distance in the 2D projection between the decision points $\tilde{q}_j$ and $\tilde{q}_{j+1}$.

One may notice that (13) is recursive. Thus, the altitude is increased in two passes through all samples: (i) in a forward way, for limiting the maximum descent by the minimum allowed pitch angle; and (ii) in a backward way, for limiting the ascend by the maximum pitch angle.

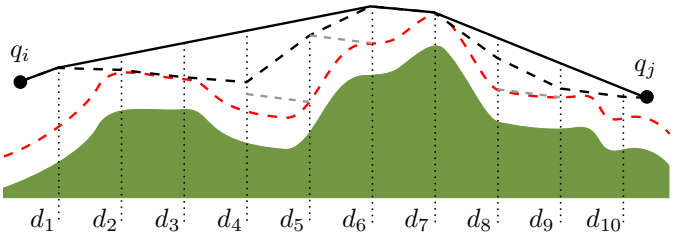


Figure 6: An example of constructing safe trajectory between q_i and q_j with the decision points at the distances $d_1, \ldots, d_{10}$. The safe altitude is shown in the red and all decision points of the trajectory has to be at higher altitude. Moreover, gliding to the (at least) next decision point must be guaranteed to guarantee a safe landing between the samples. Such a gliding trajectory is visualized as the grey dashed line, which is shown between the decision points at d_5 and d_6. Besides, the pitch limit constrains must be assured. That is why the decision point at d_4 is even higher than needed for the safe flight between d_4 and d_5. Similarly, the decision point at d_8 is higher to assure the pitch limits during the descent. A safe trajectory between q_i and q_j is shown as the black dashed line; however, such a trajectory is longer than needed, and it can be shortened by assuming only its concave envelope, as it is shown by the solid black line representing the final safe trajectory between q_i and q_j.

The determined safe altitude $\mathcal{A}_{\text{safe}}$ guarantees safe landing, but the total trajectory cost is minimized in the ELASP formulation (5). Therefore, it is required the safe trajectory has the final altitude profile $\mathcal{A}_{\text{F}}$ with the concave property that holds for any sample of the trajectory except the trajectory endpoints that correspond to the given target locations. The property holds because otherwise, unnecessary altitude changes prolong the trajectory, see Fig. 6. The concavity constraint can be expressed based on the lengths of two consecutive trajectories with distances $d_{j-1,j}$ and $d_{j,j+1}$ as

$$\mathcal{A}_{\text{F}}\left(\tilde{q}_j\right) \geq \frac{d_{j,j+1}\ \mathcal{A}_{\text{safe}}\left(\tilde{q}_{j-1}\right) + d_{j-1,j}\ \mathcal{A}_{\text{safe}}\left(\tilde{q}_{j+1}\right)}{d_{j-1,j} + d_{j,j+1}}. \quad (14)$$

Similarly to $\mathcal{A}_{\text{safe}}$, the final altitude profile is consecutively determined by a two-pass algorithm that checks the concavity forward and backward. The decoupled approach is proposed in which the vertical profile of the final sequence is post-processed to remove any vertical discontinuities at target locations if it occurred.

Here, it is worthy to remind that some locations $s_i \in S$ may not be reachable by a safe trajectory. For that reason, the formulation of Problem 3.1 allows the visiting configuration q_i can be above s_i that is penalized. However, in a practical implementation, the penalty function $\mathcal{P}$ needs to be chosen appropriately, which is further discussed in Section 7.

The fourth step of the proposed ELASP algorithm is to compute the best sequence to visiting the target locations using the determined safe trajectories $\mathcal{R}_{\text{all}}$. The sequencing part of ELASP is addressed as the Generalized Asymmetric TSP (GATSP) to visit a set of sets that are defined by the k sampled configurations $\{q_i^1, q_i^2, \ldots, q_i^k\}$ for each target location s_i, and the travel costs correspond to the costs of the corresponding trajectories from $\mathcal{R}_{\text{all}}$. The solution is the shortest closed-loop trajectory connecting exactly one sampled configuration from each set, and thus each target location is visited. The created instance of the GATSP is transformed to the Asymmetric TSP using Noon-Bean transformation [28], and the final sequence of visits can be determined optimally using Concorde [29]. However, a faster heuristic solution [30] is employed for the herein reported results, and the available LKH [38] is utilized to decrease the computational burden.

Notice that the procedure `solveGATSP()` returns not only the sequence of visits Σ to the target locations S, but also the particular selected sampled configurations $\Sigma' = \{\sigma'_1, \sigma'_2, \ldots, \sigma'_n\}$. Thus, the i-th trajectory $\mathcal{R}_i$ starts at $q_{\sigma_i}^{\sigma'_i}$ and terminates $q_{\sigma_{i+1}}^{\sigma'_{i+1}}$.

The final trajectory is created by concatenating the individual trajectories according to the sequence determined as the GATSP solution. However, a straightforward concatenation may contain discontinuities because of altitude profiles modified to meet the pitch angle constraints or the minimum safe altitude. Hence, the trajectory ends may have a higher altitude than the target locations S, and thus the same procedure for increasing the altitude of the samples, as during the safe trajectory generation, is utilized to get the final feasible trajectory. As a result, the proposed algorithm guarantees that the final trajectory $\mathcal{R}$ is closed-loop visits all the given target locations, meet the pitch angle and concavity constraints, and it is possible to land safely from any point along $\mathcal{R}$.

6. Improved ELASP$_{\text{LAZY}}$ Solver with Lazy Evaluation of Safe Trajectories

The ELASP can be solved by the proposed baseline ELASP$_{\text{BASE}}$, which, however, shows to be computationally feasible only for small instances. Therefore, we propose to follow the idea of using lower bound estimates on trajectory costs in the sequencing part of the planning that has been originally proposed for welding robots in [19].

The sequencing part is initially solved with the approximate trajectory costs, that are easy to compute. Then, the solution is refined to guarantee the trajectories are safe, and a new sequence is determined with the updated costs until all the trajectories in the solution are safe.

The improved algorithm is called ELASP$_{\text{LAZY}}$ because of the employed lazy evaluation of the safe trajectories that significantly improve computational performance. It follows the baseline algorithm with a pre-generation of emergency landing trajectories and sampling possible heading angles to form a set of configurations W. First, the trajectories between the configurations W are approximated by the so-called *terrain trajectories* that are determined by the `TerrainT`() procedure. The main simplification in the terrain trajectory determination is from using the altitude profile of the terrain below the trajectory that is updated to meet the maximum pitch angle condition(13) and the concavity constraint (14). The terrain trajectory is collision-free and fulfills all motion constraints of the aircraft, but the safe emergency landing is not guaranteed. However, its determination is significantly faster because no queries on the minimum safe altitude are needed.

The sequencing part is solved as the GATSP using transformation to the Asymmetric TSP that is solved by fast heuristic LKH [38]. Initially, the found sequence described by Σ, Σ' contains only estimations on the safe trajectories computed by `TerrainT`() procedure. Thus, the estimates are replaced by the safe trajectories determined by the `SafeT`() procedure. The solution of the sequencing part is repeated until the found sequence contains only the safe trajectories. Then, identically as in ELASP$_{\text{BASE}}$, the final found sequence is processed, and its vertical profile is updated to remove any vertical discontinuities by the `AssurePitchLimits`() function.

The proposed ELASP$_{\text{LAZY}}$ is summarized in Algorithm 4 and its real benefits in improving the computational performance, and thus improved scalability in comparison to the baseline ELASP$_{\text{BASE}}$, are reported in Section 8. Prior to that, the design of the appropriate penalty function for the relaxed ELASP problem is discussed in the following section.

Algorithm 4: ELASP$_{\text{LAZY}}$ – Proposed solver with Lazy Evaluation of Safe Trajectories

Input: $\Xi = \{\xi_1, \ldots, \xi_m\}$ – Set of the landing sites
Input: $\mathcal{T}_{\text{alt}}$ – Altitude of the terrain (or obstacles)
Input: $S = \{s_1, s_2, \ldots, s_n\}$ – Set of target locations
Input: k – Number of heading samples
Output: $\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_n\}$ – Sequence of visits
Output: $\mathcal{R} = \{\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_n\}$ – Final trajectory

```
G, A ← SafeLandingMap(Ξ, T_alt) // Algorithm 1
W ← SampleHeading(S, k)
R_all ← ∅
forall the q_i^a ∈ W do
    forall the q_j^b ∈ W do
        if i = j then
            continue
        R_all ← R_all ∪ TerrainT(q_i^a, q_j^b, G, A)
do
    Σ, Σ', R' ← SolveGATSP(R_all)
    forall the R_i ∈ R' do
        if IsTerrain(R_i) then
            R_all ← R_all \ R_i
            R_all ← R_all ∪ SafeT(q_{σ_i}^{σ'_i}, q_{σ_{i+1}}^{σ'_{i+1}}, G, A)
while R_all updated
R ← AssurePitchLimits(R')
```

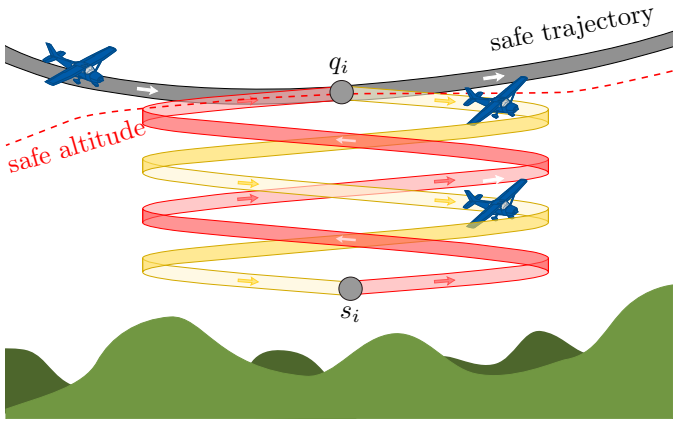


Figure 7: Possible maneuver to visit the target location s_i at the not safe altitude that minimizes the vehicle time at unsafe altitudes. First, the vehicle reaches the visiting configuration q_i at the safe altitude. Then, it descends using the spiral maneuver to reach s_i and continues with the climbing spiral to reach the safe altitude at q_i. For a relaxed visitation of s_i, the vehicle just pass the visiting configuration q_i at the safe altitude.

7. Penalty function

In the studied ELASP problem, it is requested to visit the given set of target locations such that it is guaranteed there is always an emergency landing trajectory for any point of the surveillance trajectory. However, the prescribed altitude of a target location might not allow a safe emergency landing. Therefore, visiting the location can be risky, and we can imagine a situation that a pilot takes the risk and descends to the prescribed altitude. Alternatively, the precise visitation of the target location can be relaxed, and the pilot keeps the vehicle at the minimum safe altitude.

For the former case, the risk can be minimized by minimizing the time spent at an unsafe altitude that can be minimized by the spiral type maneuver visualized in Fig. 7. For the latter case, the precise visitation of the target locations is relaxed, and a location is considered visited if the vehicle passes the configuration at the safe altitude above the target location.

A penalty function is introduced in Problem 3.1 to allow a trade-off between the distance of the relaxed trajectory from the target locations and its length. A particular value

of the penalty depends on the distance of the target location and its corresponding visiting configuration, which is always at a safe altitude. However, such a value cannot be directly added into the distance matrix utilized in the solution of GATSP instances because the altitude of two connected trajectories may differ, and a final altitude adjustment is applied. Therefore, half of the penalty value is considered for each endpoint of the trajectory, and it can be utilized in the ELASP algorithm as follows.

Let considers a safe trajectory from the configuration q_i to q_j that is generated by the `SafeT`() procedure. The safe trajectory starts at q_i^{S} which may be above q_i, and thus $[q_i^{\mathrm{S}}]^{xy} = [s_i]^{xy}$ and $[q_i^{\mathrm{S}}]^{z} \geq [s_i]^{z}$. The terminal endpoint of the safe trajectory is q_j^{S} with the analogous properties. Then, the distances utilized in the GATSP instance is computed as a half of the corresponding penalty value $\mathcal{P}$ applied at both endpoints

$$\mathcal{L}_{\mathrm{S}}(q_i, q_j) = \mathcal{L}(\texttt{SafeT}(q_i, q_j)) + \frac{\mathcal{P}(\Delta_i^{\mathrm{S}})}{2} + \frac{\mathcal{P}(\Delta_j^{\mathrm{S}})}{2}, \quad (15)$$

where the altitude differences are defined separately for both configurations

$$\Delta_i^{\mathrm{S}} = [q_i^{\mathrm{S}}]^z - [s_i]^z, \quad \Delta_j^{\mathrm{S}} = [q_j^{\mathrm{S}}]^z - [s_j]^z. \quad (16)$$

Terrain trajectory is introduced in ELASP$_{\mathrm{LAZY}}$ that might not be safe, but all motion constraints of the vehicle are met. The terrain trajectory is computed by `TerrainT`() procedure and can be considered as a lower bound estimation of the safe trajectory. Hence, it might also be augmented by the penalty.

The terrain trajectory starts at the configuration q_i^{T} and it ends at q_j^{T}. The distance utilized in the GATSP solution is given by

$$\mathcal{L}_{\mathrm{T}}(q_i, q_j) = \mathcal{L}(\texttt{TerrainT}(q_i, q_j)) + \frac{\mathcal{P}(\Delta_i^{\mathrm{T}})}{2} + \frac{\mathcal{P}(\Delta_j^{\mathrm{T}})}{2}, \quad (17)$$

where the altitude differences are defined analogously as for the safe trajectory

$$\Delta_i^{\mathrm{T}} = [q_i^{\mathrm{T}}]^z - [s_i]^z, \quad \Delta_j^{\mathrm{T}} = [q_j^{\mathrm{T}}]^z - [s_j]^z. \quad (18)$$

Terrain trajectories are utilized as the lower bound estimation of the safe trajectories that are easier to compute, and thus reduce the overall computational burden of ELASP$_{\mathrm{LAZY}}$. The necessary condition for $\mathcal{L}_{\mathrm{T}}$ to be a lower bound of $\mathcal{L}_{\mathrm{S}}$ is

$$\mathcal{L}_{\mathrm{T}}(q_i, q_j) \leq \mathcal{L}_{\mathrm{S}}(q_i, q_j), \quad (19)$$

that must hold between any two configurations q_i and q_j.

The particular value of the penalty multiplier β introduced in (4) has to ensure $\mathcal{L}_{\mathrm{T}}(q_i, q_j)$ is still a lower bound of $\mathcal{L}_{\mathrm{S}}(q_i, q_j)$. The lower bound property (19) can be expanded using (15) and (17). Then, the penalty function (4) can be applied to get

$$\begin{aligned} &\mathcal{L}(\texttt{TerrainT}(q_i, q_j)) \leq \\ &\leq \mathcal{L}(\texttt{SafeT}(q_i, q_j)) + \frac{\beta}{2}\left(\Delta_i^{\mathrm{S}} - \Delta_i^{\mathrm{T}} + \Delta_j^{\mathrm{S}} - \Delta_j^{\mathrm{T}}\right). \end{aligned} \quad (20)$$

Hence, the minimum value of the multiplier β can be determined to ensure the lower bound constraint.

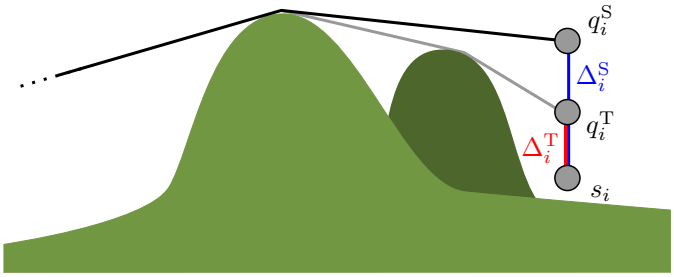


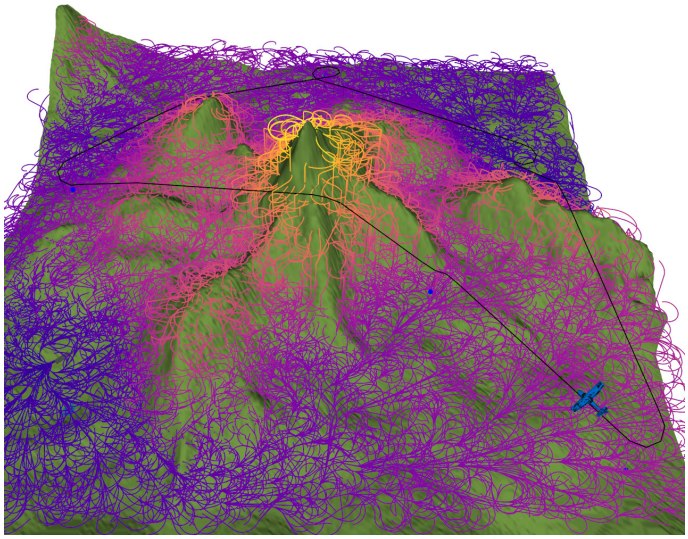
Figure 8: Penalty for safe and terrain trajectories. The terrain trajectory (gray line) may be restricted by pitch constraints of the aircraft, which results in the visiting configuration q_i^{T} above the target location s_i, and thus being penalized by $\mathcal{P}\left(\Delta_i^{\mathrm{T}}\right)$. The safe trajectory (black line) may be restricted furthermore by the safe altitude leading to the visiting configuration q_i^{S} being higher than q_i^{T}, and penalized by $\mathcal{P}\left(\Delta_i^{\mathrm{S}}\right)$.

Let the final configurations of the terrain and safe trajectories be at different altitudes, as depicted in Fig. 8. The lower bound condition (19) is met for the multiplier $\beta \geq 2$ because of the triangular inequality. The value of β influences the performance of ELASP$_{\mathrm{LAZY}}$ because, for higher values, the algorithm needs to test more sequences until it converges to a solution without contained lower bound estimations. That is why $\beta = 2$ is utilized for all the results reported in this paper.

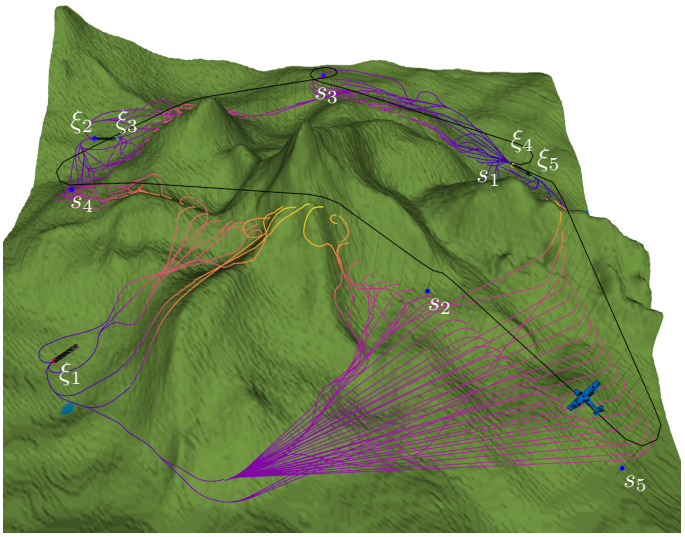
8. Results

The proposed solution to the introduced ELASP is empirically evaluated in several scenarios to demonstrate the feasibility of the proposed ELASP$_{\mathrm{BASE}}$ approach and computational benefits of the improved ELASP$_{\mathrm{LAZY}}$ algorithm in solving large problem instances. The evaluation surveillance scenarios are motivated by search-and-rescue missions or border patrol deployments in difficult terrains such as mountains. Therefore, a simulated real-life scenario with a mountain terrain is created for the evaluation, and an example of the terrain is visualized in Fig. 9b. The mission area is 5 km × 5 km large, and the terrain altitude ranges from $[0, 1500]$ meters above the sea level. Two bidirectional and one unidirectional landing sites are placed at the lower part of the terrain. Thus, emergency landing trajectories are selected from five possible landing sites, and thus $m = 5$. In all evaluation results, the computational time to create the roadmap G by Algorithm 2 is set to thirty seconds, $t_{plan} = 30$ s.

Multiple instances of the mountain scenario are assumed that consists of n randomly placed target locations $n \in \{5, 10, 20, \ldots, 100\}$ that are at 100 m above the terrain. The number of heading angles k per each location is selected from the set $k \in \{4, 8, 16\}$. The trajectory safety is examined for uniformly sampled decision points with the step size d_{step} selected from the set



(a) Pre-generated graph of possible landing trajectories



(b) Emergency landing trajectories for the final trajectory

Figure 9: (left) Visualization of the roadmap of possible emergency landing trajectories that are colored based on the altitude. (right) The final trajectory generated by the developed ELASP$_{\text{LAZY}}$ algorithm (in the black) connects the locations $s_i \in S$ with the selected landing trajectories to the landing sites ξ_j. The results were obtained for the setup with $k = 4$, $d_{\text{step}} = 100\,\text{m}$, and $t_{\text{plan}} = 30\,\text{s}$.

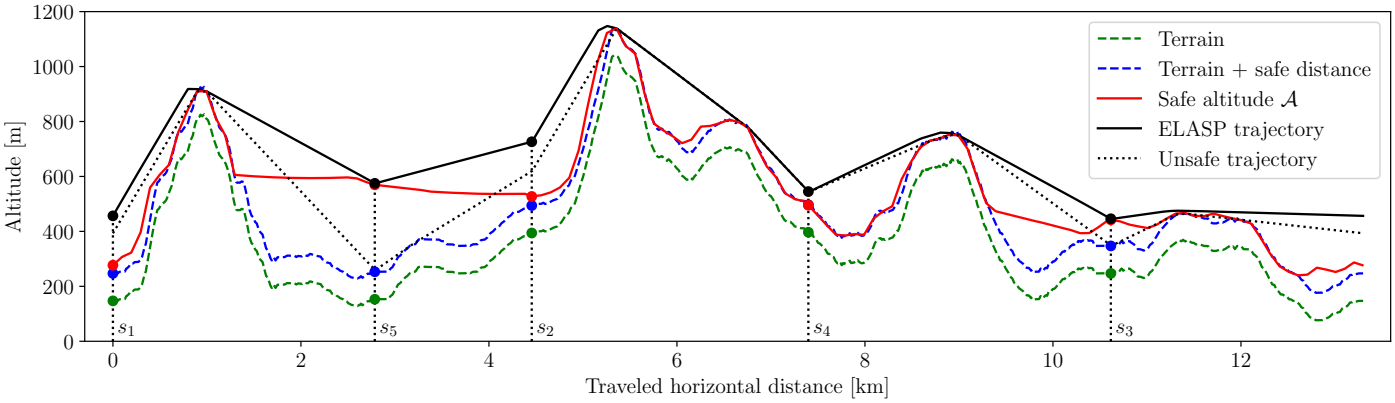


Figure 10: Altitude profiles generated by theELASP$_{\text{LAZY}}$ algorithm, where the *Unsafe trajectory* considers only the pitch angle limits and the concavity constraint. The highlighted points correspond to the locations $s_i \in S$, and the minimum altitude for safety landing is shown in the red. The results were obtained from the same run as in Fig. 9.

$d_{\text{step}} \in \{10, 30, 50, 70, 90, 100\}$ m. The emergency landing trajectory is found at each decision point, and the trajectory influences the altitude of the final trajectory. Further, the necessary altitude for a safe landing is increased by 100 m as the selected safety distance above the terrain.

The particular aircraft model is based on Cessna 172 adopted from [36], including its complex model of the altitude loss. The utilized aircraft model parameters are depicted in Table 1. The maximum allowed bank angle of the aircraft is 60° leading to the minimum turning radius of $\rho = 65.7\,\text{m}$ for the assumed constant forward velocity $v = 33.4\,\text{m}\,\text{s}^{-1}$. The assumed pitch angle limitations are $\psi_{\max} = 30°$ and $\psi_{\min} = -20°$. The altitude loss model uses the optimal glide speed v, and the altitude loss depends on the utilized turning radius. The minimum angle of emergency descent is 4.9°, which is achieved during a straightforward glide.

Table 1: Used technical parameters of Cessna 172 aircraft.

Parameter	Symbol	Value
Vehicle mass	m	1000 kg
Wing area	S	16.2 m^2
Wing span	b	11 m
Span efficiency factor	ϵ	0.8
Coefficient of geometric drag	C_{D0}	0.0341
Optimal glide airspeed	v	33.4 m s^{-1}
Lift-induced drag coef.	L	0.053
Maximum roll angle	$\lvert\varphi_{\max}\rvert$	60°
Minimum turning radius	ρ	65.7 m

The proposed ELASP algorithms have been implemented in C++ and executed on a single core of the Intel

Xeon Scalable Gold 6146 running at 3.2 GHz. Therefore, the computational performance of both ELASP$_{\text{BASE}}$ and ELASP$_{\text{LAZY}}$ can be directly compared. Each problem instance has been solved ten times, and the reported values are medians accompanied by the 60 % non-parametric confidence intervals visualized as an area around the average values in the presented plots. The studied computational performance is overviewed in Figs. 11 to 14 and detailed computational performance is listed in Tables 2 and 3.

An example of the landing trajectories is shown in Fig. 9a, and the final planned trajectory with its safe emergency landing trajectories are shown in Fig. 9b. The altitudes for the final found trajectory are further visualized in Fig. 10 to demonstrate the behavior of the improved ELASP$_{\text{LAZY}}$ algorithm. The highest points of the terrain significantly influence an altitude of the final trajectory. The constraints on the pitch angle are applied, and the maximum slope around the highest points is limited. It causes that the locations s_2 cannot be visited in the requested altitude, and the altitude of the final trajectory is increased. The requirement for the safe landing further increases the altitude at s_2 because the highest point of the terrain needs to be reached from an even higher altitude to provide the option to over-fly the mountain safely if there is not enough space to turn back. The final altitude adjustment by the emergency landing planning is mostly influenced at the location s_5 (bottom right in Fig. 9b). Here, the altitude is not affected by the pitch angle constraints nor the terrain, but mostly by the distance to the closest landing site. A similar case can be seen even for the location s_3 with a much lower altitude increase.

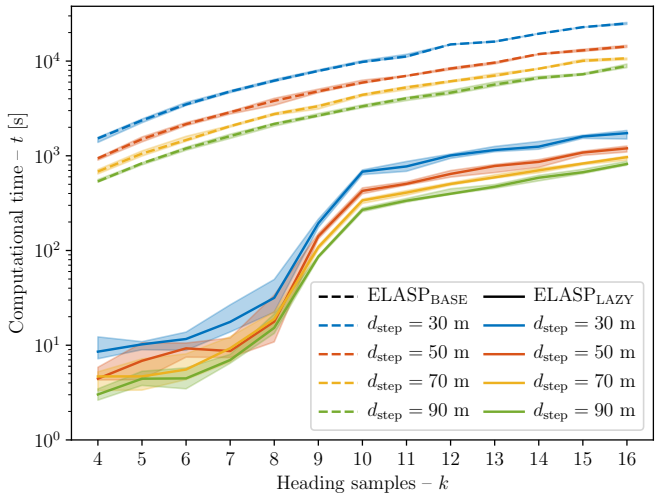


Figure 11: Influence of the number of heading samples k to the required computational times needed to solve ELASP instance with $n = 10$ target locations.

The computational benefits of ELASP$_{\text{LAZY}}$ are evident from the overview in Fig. 11 where the influence of the number of heading samples can be studied. More samples increase the computational burden, but

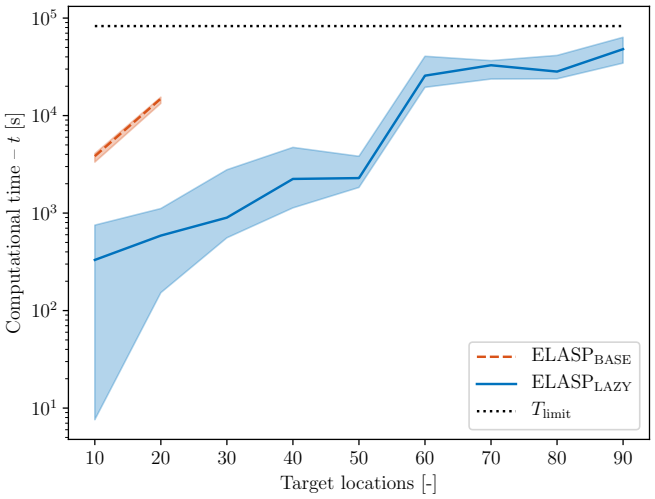


Figure 12: Influence of the increasing number of the target locations n to the scalability of the ELASP algorithms. The shown results are for $k = 8$ heading samples and $d_{\text{step}} = 50$ m.

ELASP$_{\text{LAZY}}$ is still about one order of magnitude faster than ELASP$_{\text{BASE}}$. The scalability of the algorithm for an increasing number of target locations n is further visualized in Fig. 12. It is evident from the results that ELASP$_{\text{BASE}}$ does not scale with n, and a solution for instances with more than 20 target locations are found in 24 hours, which is considered impractically high.

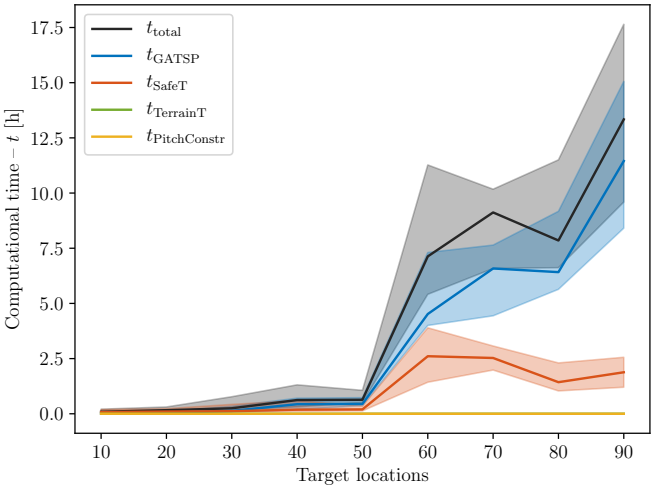


Figure 13: Computational performance of the particular steps of the ELASP$_{\text{LAZY}}$ algorithm for increasing the number of target locations n. The results are obtained for five random instances, each solved ten times, and $k = 8$, and $d_{\text{step}} = 30$ m.

The computational performance of particular steps of ELASP$_{\text{LAZY}}$ is shown in Fig. 13 indicating that the most computationally demanding part is a solution of the GATSP, and generation of safe trajectories is the most demanding part only for small instances. The benefit of lazy evaluation of safe trajectories in ELASP$_{\text{LAZY}}$ is visualized in Fig. 14 for the reduced number of examined configuration for safe emergency landing. The number of

Table 2: Average computational times of ELASP algorithms' steps for instances with the sampling step $d_{\text{step}} = 50$ m.

ELASP Step	$n = 5,\ k = 4$		$n = 5,\ k = 8$		$n = 20,\ k = 8$	
	$\text{ELASP}_{\text{BASE}}$	$\text{ELASP}_{\text{LAZY}}$	$\text{ELASP}_{\text{BASE}}$	$\text{ELASP}_{\text{LAZY}}$	$\text{ELASP}_{\text{BASE}}$	$\text{ELASP}_{\text{LAZY}}$
Roadmap generation	30 s	30 s	30 s	30 s	30 s	30 s
Safe trajectories	279 s	111 s	1477 s	497 s	15013 s	268 s
Solving the GATSP	12 ms	514 ms	<1 s	6 s	<1 s	87 s
Limits the pitch angle	<1 ms	<1 ms	<1 ms	<1 ms	<1 ms	<1 ms
Total	309 s	142 s	1507 s	533 s	15044 s	385 s

Table 3: Average numbers of calls (executions) of ELASP algorithms' steps for instances with sampling step $d_{\text{step}} = 50$ m.

ELASP Step	$n = 5,\ k = 4$		$n = 5,\ k = 8$		$n = 20,\ k = 8$	
	$\text{ELASP}_{\text{BASE}}$	$\text{ELASP}_{\text{LAZY}}$	$\text{ELASP}_{\text{BASE}}$	$\text{ELASP}_{\text{LAZY}}$	$\text{ELASP}_{\text{BASE}}$	$\text{ELASP}_{\text{LAZY}}$
Roadmap expansions	7458	7420	5566	7286	7591	7541
Safe trajectories queries	20.8k	8.1k	83.3k	34.9k	1.2M	21.1k
Number of GATSP solutions	1	33	1	134	1	145
Calls of pitch angle limit	1	1	1	1	1	1

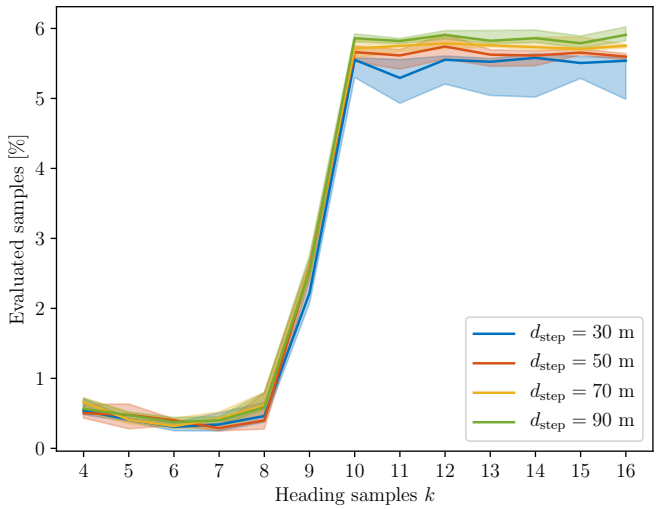


Figure 14: Percentage of the relative number of samples evaluated for safe landing by $\text{ELASP}_{\text{LAZY}}$ to the number of all evaluations of $\text{ELASP}_{\text{BASE}}$.

examinations increases with the number of heading samples k because more updates on the trajectory estimates are needed to exploit more options on how to connect two target locations using different heading angles. However, only a fraction of queries on safe trajectories is needed in comparison with the former $\text{ELASP}_{\text{BASE}}$.

Detail computational times in Table 2 further support that the lazy evaluation vastly reduces the number of queried samples. On the other hand, there are multiple calls of the LKH solver to solve the GATSP with the updated trajectory costs repeatedly, see Table 3. Overall, it is less demanding to heuristically solve the GATSP instances multiple times than determine all safe trajectories connecting all pairs of target locations while getting the same quality of solutions.

The influence of the sampling step d_{step}, and the number of heading samples k to the solution quality is depicted in Fig. 15. Both parameters have a negligible impact on the final trajectory length as the median differs only in tens of meters for tens of kilometers long trajectory, which is less than 0.2 %. Also, both algorithm variants provide similar results, albeit there is a difference in the stability of the solutions that is supposed to be related to the utilized heuristic LKH [38]. Although the differences are tiny, there is a noticeable trend of a slightly better solution for longer d_{step}, which might be bit surprising. This phenomenon is discussed in the following section.

8.1. Discussion

The $\text{ELASP}_{\text{LAZY}}$ algorithm provides a significant speed-up over $\text{ELASP}_{\text{BASE}}$ in the studied scenarios, while solutions of the same quality are found. The speed-up is achieved by the reduction of computationally demanding determination of safe trajectories despite the fact that more calls of the GATSP solver are required. The utilized heuristic solver LKH [38] has reported empirically measured asymptotic complexity about $\mathcal{O}((nk)^{2.2})$ [30]. On the other hand, the asymptotic complexity of safe trajectory generation can be bounded by $\mathcal{O}(n^2k^2)$. Nevertheless, the practical results indicate that it is much faster to call LKH multiple times than to compute safe trajectories between all target locations, as in $\text{ELASP}_{\text{BASE}}$.

The computational performance of $\text{ELASP}_{\text{LAZY}}$ depends on the number of heading angles k, where for a high number of k, samples are too close, and thus more

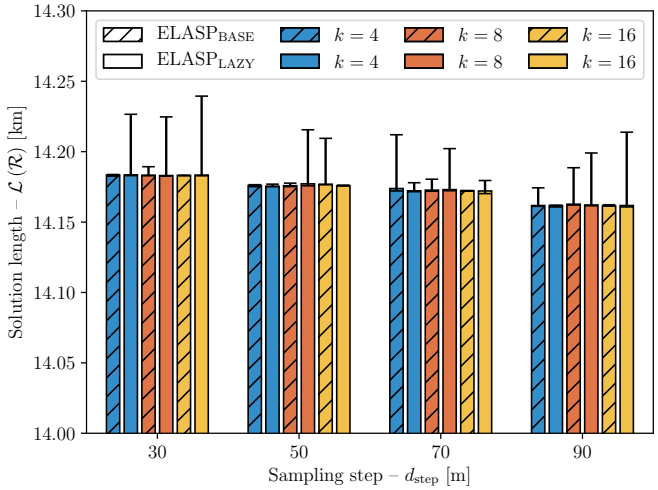


Figure 15: Influence of the sampling step d_{step} and the number of heading samples k to the length of the final trajectory. The depicted results for the both algorithms ELASP$_{\text{LAZY}}$ and ELASP$_{\text{BASE}}$ have been computed on the same instance with $n = 10$ target locations.

solutions of the GATSP are necessary to refine all estimations of the possible trajectories. Regarding the solution quality, it seems that even low values of k are sufficient. Therefore, a suitable number of heading samples seems to be eight as for a higher number, the number of evaluated trajectories starts to grow, see Figs. 11 and 14.

The solution of the ELASP problem seems to be insensitive to the selection of the sampling step d_{step}, which is mostly because of small altitude loss between two decision points for the considered range of $d_{\text{step}} \in [30, 90]$ in Fig. 15. However, the sampling step is utilized for the detection of the terrain altitude and queries on the safe altitude that is probably the reason why the total length slightly decreases for longer sampling steps in Fig. 15 as a result of exploiting the safe distance above the terrain. Thus, the trend will unlikely continue for very long sampling steps, because in such a case, it would be necessary to introduce an individual sampling for collisions with the terrain to guarantee the trajectories are collision-free and sufficiently above the terrain.

9. Conclusion

In this paper, we propose a solution of the *Emergency Landing Aware Surveillance Planning* (ELASP) problem that combines finding a feasible multi-goal trajectory to visit a set of target locations with the guarantee of safe emergency landing from any point of the trajectory in a case of the loss of thrust. The proposed solution leverages on the pre-computed roadmap of possible landing trajectories within the surveillance mission area that is utilized during the multi-goal planning together with the lazy determination of safe trajectories to reduce the computational burden. The proposed algorithmic solution has been empirically evaluated in the mountain scenario to demonstrate the difficulty of ELASP, the effect of the pre-computed landing trajectories, and the importance of using the lazy evaluation technique to improve the scalability of the solution. Possible future research directions are in the improvement and generalization of the vehicle model to consider more complex trajectories, and thus support further optimization of the found trajectories.

Acknowledgements

The presented work has been supported by the Czech Science Foundation (GAČR) under research project No. 19-20238S. The access to the computational infrastructure of the OP VVV funded project CZ.02.1.01/0.0/0.0/16_019/0000765 “Research Center for Informatics” is also gratefully acknowledged.

The early results on the herein presented method have been presented at the European Conference on Mobile Robots (ECMR) 2019 [8].

References

[1] A. Tahir, J. Böling, M.-H. Haghbayan, H. T. Toivonen, J. Plosila, Swarms of Unmanned Aerial Vehicles — A Survey, Journal of Industrial Information Integration 16 (2019) 100106. doi:10.1016/j.jii.2019.100106.

[2] K. Kanistras, G. Martins, M. J. Rutherford, K. P. Valavanis, Survey of unmanned aerial vehicles (uavs) for traffic monitoring, in: K. P. Valavanis, G. J. Vachtsevanos (Eds.), Handbook of Unmanned Aerial Vehicles, Springer Netherlands, Dordrecht, 2015, pp. 2643–2666. doi:10.1007/978-90-481-9707-1_122.

[3] M. Dunbabin, L. Marques, Robots for Environmental Monitoring: Significant Advancements and Applications, IEEE Robotics & Automation Magazine 19 (1) (2012) 24–39. doi:10.1109/MRA.2011.2181683.

[4] K. Savla, E. Frazzoli, F. Bullo, On the point-to-point and traveling salesperson problems for dubins' vehicle, in: American Control Conference, IEEE, 2005, pp. 786–791. doi:10.1109/ACC.2005.1470055.

[5] L. E. Dubins, On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents, American Journal of Mathematics (1957) 497–516doi:10.2307/2372560.

[6] J. D. Kenny, 26th Joseph T. Nall Report, Richard G. McSpadden, JR., 2017.

[7] Loss of Thrust in Both Engines After Encountering a Flock of Birds and Subsequent Ditching on the Hudson River, National Transportation Safety Board, 2009.

[8] P. Váňa, J. Faigl, J. Sláma, Emergency landing aware surveillance planning for fixed-wing planes, in: European Conference on Mobile Robots (ECMR), IEEE, 2019, pp. 1–6. doi:10.1109/ECMR.2019.8870933.

[9] D. G. Macharet, M. F. M. Campos, A survey on routing problems and robotic systems, Robotica 36 (12) (2018) 1781–1803. doi:10.1017/S0263574718000735.

[10] P. Oberlin, S. Rathinam, S. Darbha, Today's Traveling Salesman Problem, IEEE Robotics & Automation Magazine 17 (4) (2010) 70–77. doi:10.1109/MRA.2010.938844.

[11] J. Faigl, P. Váňa, R. Pěnička, M. Saska, Unsupervised learning-based flexible framework for surveillance planning with aerial vehicles, Journal of Field Robotics 36 (1) (2019) 270–301. doi:10.1002/rob.21823.

[12] J. Faigl, P. Váňa, M. Saska, T. Báča, V. Spurný, On solution of the dubins touring problem, in: European Conference on Mobile Robots (ECMR), 2017, pp. 1–6. doi:10.1109/ECMR.2017.8098685.

[13] J. Faigl, P. Váňa, Surveillance planning with bézier curves, IEEE Robotics and Automation Letters 3 (2) (2018) 750–757. doi:10.1109/LRA.2018.2789844.

[14] H. Chitsaz, S. M. LaValle, Time-optimal paths for a dubins airplane, in: IEEE Conference on Decision and Control, 2007, pp. 2379–2384. doi:10.1109/CDC.2007.4434966.

[15] M. Owen, R. W. Beard, T. W. McLain, Implementing dubins airplane paths on fixed-wing uavs, in: Handbook of Unmanned Aerial Vehicles, Springer, 2015, pp. 1677–1701. doi:10.1007/978-90-481-9707-1_120.

[16] P. Váňa, J. Sláma, J. Faigl, The dubins traveling salesman problem with neighborhoods in the three-dimensional space, in: IEEE International Conference on Robotics and Automation (ICRA), 2018, pp. 374–379. doi:10.1109/ICRA.2018.8460957.

[17] S. Paul, F. Hole, A. Zytek, C. A. Varela, Flight Trajectory Planning for Fixed-Wing Aircraft in Loss of Thrust Emergencies, arXiv preprint arXiv:1711.00716.

[18] S. Paul, F. Hole, A. Zytek, C. A. Varela, Wind-aware trajectory planning for fixed-wing aircraft in loss of thrust emergencies, in: IEEE/AIAA 37th Digital Avionics Systems Conference (DASC), IEEE, 2018, pp. 1–10. doi:10.1109/DASC.2018.8569842.

[19] M. Saha, T. Roughgarden, J.-C. Latombe, G. Sánchez-Ante, Planning Tours of Robotic Arms among Partitioned Goals, International Journal of Robotics Research 25 (3) (2006) 207–223. doi:10.1177/0278364906061705.

[20] V. Darbari, S. Gupta, O. P. Verma, Dynamic motion planning for aerial surveillance on a fixed-wing uav, in: International Conference on Unmanned Aircraft Systems (ICUAS), IEEE, 2017, pp. 488–497. doi:10.1109/ICUAS.2017.7991463.

[21] H. J. Sussmann, Shortest 3-dimensional paths with a prescribed curvature bound, in: IEEE Conference on Decision and Control, Vol. 4, 1995, pp. 3306–3312. doi:10.1109/CDC.1995.478997.

[22] G. Ambrosino, M. Ariola, U. Ciniglio, F. Corraro, A. Pironti, M. Virgilio, Algorithms for 3d uav path generation and tracking, in: IEEE Conference on Decision and Control, 2006, pp. 5275–5280. doi:10.1109/CDC.2006.377555.

[23] P. Váňa, J. Faigl, J. Sláma, R. Pěnička, Data collection planning with dubins airplane model and limited travel budget, in: European Conference on Mobile Robots (ECMR), 2017, pp. 1–6. doi:10.1109/ECMR.2017.8098715.

[24] P. Váňa, A. A. Neto, J. Faigl, D. G. Macharet, Minimal 3D Dubins Path with Bounded Curvature and Pitch Angle, in: IEEE International Conference on Robotics and Automation (ICRA), 2020, pp. 8497–8503.

[25] P. F. Di Donato, E. M. Atkins, Three-dimensional dubins path generation and following for a uas glider, in: 2017 International Conference on Unmanned Aircraft Systems (ICUAS), IEEE, 2017, pp. 294–303.

[26] A. A. Neto, D. G. Macharet, M. F. Campos, 3d path planning with continuous bounded curvature and pitch angle profiles using 7th order curves, in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2015, pp. 4923–4928. doi:10.1109/IROS.2015.7354069.

[27] K. Obermeyer, P. Oberlin, S. Darbha, Sampling-based roadmap methods for a visual reconnaissance uav, in: AIAA Guidance, Navigation, and Control Conference, 2010, p. 7568. doi:10.2514/6.2010-7568.

[28] C. Noon, J. Bean, An Efficient Transformation of the Generalized Traveling Salesman Problem, Tech. Rep. 91–26, Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI, USA (1991). doi:10.1080/03155986.1993.11732212.

[29] D. Applegate, R. Bixby, V. Chvátal, W. Cook, D. Espioza, M. Goycoolea, K. Helsgaun, Concorde TSP Solver, [last accessed on 2 May 2019] (2003).
URL https://www.tsp.gatech.edu/concorde.html

[30] K. Helsgaun, An Effective Implementation of the Lin-Kernighan Traveling Salesman Heuristic, European Journal of Operational Research 126 (1) (2000) 106–130. doi:10.1016/S0377-2217(99)00284-2.

[31] X. Yu, J. Hung, A genetic algorithm for the dubins traveling salesman problem, in: IEEE International Symposium on Industrial Electronics, 2012, pp. 1256–1261. doi:10.1109/ISIE.2012.6237270.

[32] X. Zhang, J. Chen, B. Xin, Z. Peng, A memetic algorithm for path planning of curvature-constrained uavs performing surveillance of multiple ground targets, Chinese Journal of Aeronautics 27 (3) (2014) 622–633. doi:10.1016/j.cja.2014.04.024.

[33] J. Faigl, P. Váňa, Self-organizing map for the curvature-constrained traveling salesman problem, in: International Conference on Artificial Neural Networks (ICANN), 2016, pp. 497–505. doi:10.1007/978-3-319-44781-0_59.

[34] N. Meuleau, C. Plaunt, D. Smith, T. Smith, Emergency landing planning for damaged aircraft, in: Innovative Applications of Artificial Intelligence Conference, 2009, pp. 3247–3259.

[35] P. Eng, Path planning, guidance and control for a uav forced landing, Ph.D. thesis, Queensland University of Technology (2011). doi:10.1109/MRA.2010.936949.

[36] P. Váňa, J. Sláma, J. Faigl, P. Pačes, Any-time trajectory planning for safe emergency landing, in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018, pp. 5691–5696. doi:10.1109/IROS.2018.8594225.

[37] B. Ayhan, C. Kwan, B. Budavari, J. Larkin, D. Gribben, Preflight contingency planning approach for fixed wing uavs with engine failure in the presence of winds, Sensors 19 (2) (2019) 227. doi:10.3390/s19020227.

[38] K. Helsgaun, LKH solver 2.0.9, [last accessed on 26 Jan 2020] (2018).
URL http://www.akira.ruc.dk/~keld/research/LKH/