Bio-inspired Algorithm for Designing an Adaptive Cyclic Corridor for UAS Traffic Management

Matthieu Verdoucq*, Rodolphe Fremond*, Zeynep Bilgin‡, Murat Bronz*†
*ENAC Airbus Sopra Steria Drones and UTM Research Chair
†Fédération ENAC ISAE-SUPAERO ONERA, Université de Toulouse, Toulouse, France
‡Chair of Rotorcraft and Vertical Flight, Technical University of Munich, Munich, Germany
{matthieu.verdoucq, rodolphe.fremond, murat.bronz}@enac.fr
zeynep.bilgin@tum.de

***Abstract*—This paper presents a novel bio-inspired method for designing cyclic airspace corridors to support high-throughput, bidirectional flows of Unmanned Aircraft Systems at Very Low Level. We adapt the Slime-Mold Algorithm to generate corridors by modeling vertiports as attractors, no-fly zones as repulsors, and using a discretized pheromone field to guide agent behavior. Agents operate in three roles: explorers, exploiters, and trail-thickeners, and interact solely via pheromone diffusion and decay. A food timer mechanism regulates exploration based on the average spacing between vertiports. From the resulting pheromone distribution, a Catmull–Rom spline is fitted to extract a smooth, continuous corridor. The design is evaluated using five criteria: vertiport proximity, no-fly zone avoidance, self-intersection, curvature smoothness, and total length. We evaluate our approach on three large-scale urban scenarios with up to 15 vertiports and 10 no-fly zones. Results show convergence within 200 iterations, rapid adaptation to topological changes, and the potential to support more complex corridor structures giving promises to the method's suitability for real-time, dynamic airspace design in UAS traffic management.**

***Keywords*—Slime-Mold Algorithm; Urban Air Mobility; Airspace Design; Cyclic Corridor; Unmanned Aircraft System Traffic Management.**

I. Introduction

In recent years, Urban Air Mobility (UAM) has been emerged as a potential solution to address increasing urban on-ground congestion, extending a new dimension of mobility through aerial routes.

As demand for UAM grows, managing the Very Low Level (VLL) airspace in urban environments becomes increasingly complex and calls for modernized Air Traffic Management (ATM) solutions [1].

Current urban airspace management approaches fall into two main categories. The first group advocates for less structured airspace and proposes separation by technology level. This approach is safe and efficient with vehicles with high autonomy level [2]–[4]. The second approach is having a more structured system with the introduction of new airspace classes overseen by Air Traffic Control (ATC) centers [5].

This research is funded by ENAC - Airbus - Sopra Steria, Drones and UTM Research Chair.

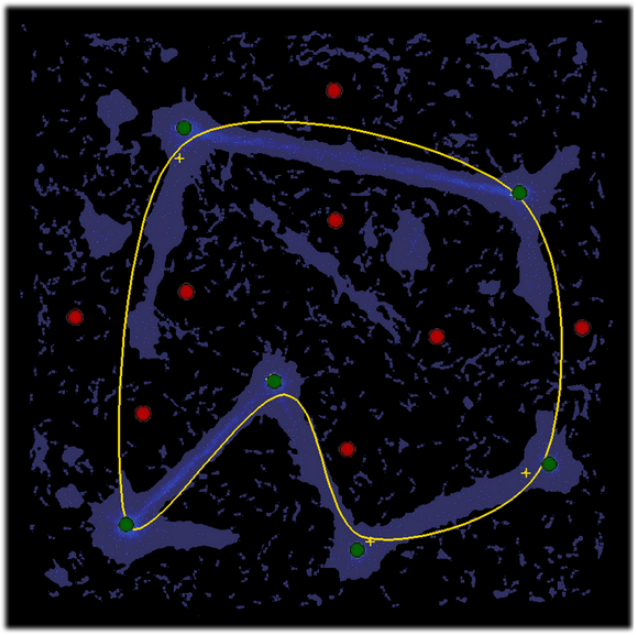


Fig. 1. Slime Mold Algorithm used for generating a corridor connecting vertiports (in green) while avoiding no fly zones (in red). The blue distribution is the pheromone grid, the media used by all the slime agents to communicate. They deposit pheromone to make a global distribution they can follow to reach all interest points.

In densely populated areas, increased structure is necessary [6]–[8]. Introducing lanes, tubes, or corridors provides a more structured airspace, and also utilizes an already-established traffic-management system, enabling smoother integration of urban air mobility [9]–[11].

One such emerging concept is cyclic corridors, structured and looped aerial routes that allow for continuous, multi vehicle usage [11], [12]. These corridors offer benefits such as predictable flow, fair access, and simplified deconfliction strategies. However, unlike traditional static corridors, cyclic corridors require dynamic adaptability to accommodate real time changes such as shifting demand, weather conditions, and temporary airspace restrictions.

Conventional airspace management methods often rely heavily on static and predefined pathways or zones, which fail to effectively accommodate rapid environmental changes and vehicle dynamic [13]. These methods typically do not adjust flexibly in real time, leading to suboptimal routing to keep the highest operational safety [13], [14]. The cyclic corridor concept, while common in ground transportation networks such as ring roads or roundabouts, has yet to be fully explored in aerial traffic systems. Challenges inherent to cyclic corridors include ensuring robust safety protocols, managing dynamic variability, and optimizing throughput without compromising operational flexibility.

To address these limitations, this paper introduces a biologically inspired approach to cyclic corridor generation using the Slime Mold Algorithm (SMA) [15]–[18], a swarm based metaheuristic model derived from the adaptive foraging behavior of *Physarum polycephalum*. SMA agents interact indirectly via a pheromone medium, dynamically adapting to environmental feedback and updating path preferences. SMA balances exploration and exploitation without explicit global memory, making it especially well suited to environments with frequent variability. These properties allow SMA to generate cyclic corridors that not only respond rapidly to changes but also self optimize through reinforcement mechanisms. In these simulated environment, dynamically propagating and decaying information, the swarm particles (also named slime agents or slime throughout the paper) deposit information depending on their respective strategies and position. The overall information (described as a pheromone) can be used to draw a curvature path linking food points (in our case vertiports) while avoiding obstacles (no-fly zones, real obstacles, geofencing, ...).

This research introduces a novel bio-inspired method based on the SMA for UTM, and investigates its application to the design of cyclic airspace corridors. In this conceptual airspace, the generated corridors connect vertiport infrastructure at the city scale while safely avoiding no-fly zones. The main contributions are as follows:

- A swarm-based method for cyclic corridor design is proposed using SMA, involving three types of agents with distinct roles: vertiport exploiters, airspace explorers, and trail-thickeners that reinforce dominant paths.
- These decentralized agents interact indirectly by depositing pheromones in the environment, collectively shaping high-density regions from which the cyclic airspace corridor is extracted and discretized into usable waypoints.
- The method is validated through simulations under varying traffic and environmental constraints, demonstrating scalability, robustness, and performance across multiple urban scenarios.

II. Related Work

Our biological inspiration started within the family of models called Path Formation models [19], where a swarm systems modify and communicate indirectly through an environment, converging to global solutions created by all its agents. The most common model coming form this family is the Ant Colony Optimization (ACO) [20]–[23]: it relies on a pheromone matrix and an outer loop in which ants construct solutions, the colony evaluates them, and pheromone trails are reinforced or evaporated to bias the next iteration. The value update of the pheromone data at a time t and a position (x, y) is, thus, linked to global conditions and or score evaluation.

Conventional SMA models the foraging behavior of *Physarum polycephalum* [15], [16]. Search agents ("slimes") adjust their positions with adaptive positive/negative weights that mimic rhythmic contraction waves, thereby, balancing exploration and exploitation. No explicit pheromone is stored; instead, the model builds veins whose thickness reflects food quality.

Other bio-inspired metaheuristic models such as Particle Swarm Optimization [24], Pelican Optimization Algorithm [25], and more [26] aim at the same goal, while providing different score function, agent update dynamics, and overall solution convergence. There exist a lot of different models in this particular field of Swarm Intelligence based models, with the most known being genetic algorithms, and evolutionary algorithms.

Both the ACO and the SMA are the closest to what this work is trying to achieve, by using stigmergy, and forming paths/trails. By using the dynamic sensing and movement of the slime, and a pheromone grid used for indirect communication, our different agents generate a pheromone path that is updated frequently to match the decaying and diffusion of the information, while converging to a satisfying solution rapidly, and even more adaptively.

The problem of generating a path between interest points while avoiding designated areas could be addressed by every model mentioned in this section, and it relies on the complexity, speed, and robustness of each solution to define which is the best one for this specific purposes.

III. Methodology

Sec.III-A begins with a description of the airspace environment. Sec.III-B and Sec.III-C introduce the development of the SMA, respectively covering the biological agent model and the pheromone field design. Sec.III-D explains how airspace corridors are derived from agent behavior. Finally, Sec.III-E compiles the overall SMA workflow.

A. Airspace Environment Modeling

In this research, we model the airspace as a square environment that accommodates:

- A set of n_v **vertiports**: each vertiport $(v_j)_{j\in[1,\,n_v]}$ is modelled as a disk $V_j = (x_{V_j},\, y_{V_j},\, r_V)$, where $(x_{V_j},\, y_{V_j})$ represents its location and r_V denotes the radius from the center to the edge. r_V is fixed and common to all vertiports.
- A set of n_o **no-fly zones**: each obstacle $(o_j)_{j\in[1,\,n_o]}$ is also modelled as a disk $O_j = (x_{O_j},\, y_{O_j},\, r_{O_j})$, following

the same representation as vertiports, except that each obstacle o_i may have a different radius r_{O_i}.
- A cyclic **airspace corridor**: represented by N_c control points C_i, where $i \in [1,\, N]$. The corridor's geometry is optimized based on the position of these control points, which are used to generate Catmull-Rom splines as detailed later in Sec.III-D2.

To optimize the cyclic corridor using the bio-inspired SMA algorithm, we scale the physical airspace into a normalized workspace box of side length L, denoted by $\tilde{\Omega} = [0,\, L] \times [0,\, L]$, and discretized into $N \times N$ cells. The scaling function is defined in Eq.1.

$$\phi : (x, y) \in [0, L]^2 \;\longrightarrow\; (\tilde{x}, \tilde{y}) = \left(2\,\frac{x}{L} - 1,\; 2\,\frac{y}{L} - 1\right) \in \tilde{\Omega} \qquad (1)$$

The inverse mapping ϕ^{-1} converts any point from the normalized airspace back to continuous Euclidean coordinates, enabling large-scale representation in our experiments.

In our methodology, we assume a working space $\tilde{\Omega} = [-1,\, 1] \times [-1,\, 1]$. Choosing a finer lattice (i.e., larger N) increases spatial fidelity, at the cost of higher memory and computational demand in the bio-inspired approach detailed in the next sections. While still slightly out of context at this stage, it is important to note that the SMA is characterized by a population of biological agents, described in Sec.III-B1, which move continuously within the normalized airspace. Although the agents navigate in continuous space, their interactions with the environment, such as pheromone deposition, as detailed in Sec.III-C, are mapped onto the underlying discrete lattice.

B. Biological Agent Modeling

This section introduces one of the two aspects of our research on SMA to airspace design, focusing here on the modeling of agents. It is structured as follows: Agent definitions are provided in Sec.III-B1, their sensing and motion mechanisms are described in Sec.III-B2, and the management of their memories is detailed in Sec.III-B3.

1) Definitions : The proposed SMA models a swarm of autonomous, biologically-inspired "slime" agents that evolve within the continuous two-dimensional airspace environment introduced in Sec.III-A.

These agents are purely fictitious and are not directly correlated with any specific drone or air traffic entity.

From the perspective of the agents, both vertiports and no-fly zones are assimilated to food sources: vertiports are interpreted as attractive zone (i.e., food), while no-fly zones are treated as repulsive zone.

The airspace corridor is generated in response to the movement and flow of the agents, specifically, how frequently they visit certain regions and the intensity of their presence.

To achieve this, agents interact with a discretized version of the airspace, which is perceived as a pheromone field from the agents' point of view, as detailed later in Sec.III-C. Each cell in this field contains a pheromone concentration. Agents follow a score-driven mechanism based on local pheromone values and environmental cues. In addition to reacting to the pheromone field, agents also contribute to its evolution by depositing pheromones along their trajectories, reinforcing frequently traveled paths.

A "swarm" is defined here as a group of decentralized agents that interact locally with one another and with the environment, collectively exhibiting emergent behavior. The swarm consists of three distinct agent types:
- **Exploiters**: greedy agents that pursue attractive food as much as possible while avoiding repulsive food.
- **Explorers**: entropic agents that tend to explore previously unvisited regions of the airspace. They are curious and not guided by food in the short term.
- **Trail Thickeners**: follower agents that preferentially move toward regions with high pheromone concentration.

2) Sensing and Motion Mechanism : The decision-making of each agent is based on its local sensing mechanism, which allows it to perceive the surrounding environment. The sensing architecture is defined by three key parameters:
- Sensor distance d_s, representing how far the agent can detect pheromone, food, or threats (i.e., repulsive food).
- Sensor angle θ_s, defining the total angular range covered by the sensors.
- N_{angle}, the number of sensor orientations on each side of the agent's heading, resulting in a total of $2N_{\text{angle}}+1$ sensing directions (including the forward-facing direction), as illustrated in Fig. 2.

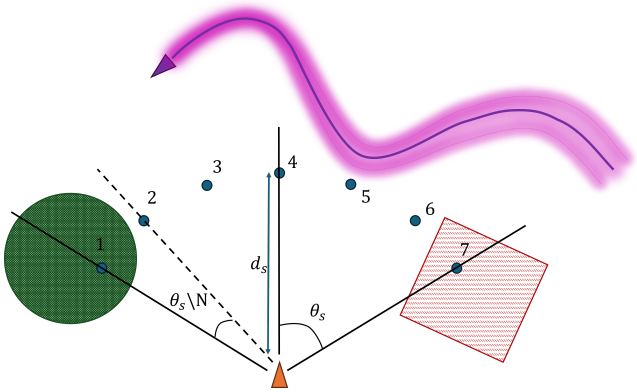


Fig. 2. Agent sensing mechanism: sensors distributed across a defined angle θ_s and distance d_s. The agent evaluates discrete points to determine the optimal movement direction. Based on the slime type, the attraction and repulsion to different quantities are defined in Eq. (3).

At each iteration, an agent i located at position (x_i, y_i) computes the coordinates of each sensing point (x_k, y_k), for $k \in [\![1,\, 2N_{\text{angle}} + 1]\!]$, using Eq.2.

$$\begin{aligned} &(x_k, y_k) = (x_i, y_i) + d_s \cdot (\sin\theta_k,\, \cos\theta_k) \\ &\text{with } \theta_k = \left(\frac{k-1}{N_{\text{angle}}} - 1\right) \cdot \frac{\theta_s}{2} \end{aligned} \qquad (2)$$

We define the following field functions and parameters:

- $P(x, y)$: pheromone intensity at location (x, y).
- $F(x, y)$: food attraction value at location (x, y).
- $T(x, y)$: threat repulsion value (e.g., from no-fly zones) at location (x, y).
- w_p, w_f, w_t: behavior-specific weights associated with pheromone, food, and threat terms respectively. Each weight from Eq.3 is related to the type of the agent as detailed later in Table II.
- ϵ: small random noise to promote exploration.

At each step, the agent computes a score $S_{i,k}$ for every sensing point as described in Eq.3.

$$S_{i,k} = w_{p_i} \cdot P(x_k, y_k) + w_{f_i} \cdot F(x_k, y_k) - w_{t_i} \cdot T(x_k, y_k) + \epsilon \quad (3)$$

The agent then selects the direction corresponding to the highest score as described in Eq.4.

$$\begin{aligned} k^* &= \arg\max_k S_{i,k} \\ (x_i, y_i)(t+1) &= (x_i, y_i)(t) + v \cdot (\sin\theta_{k^*}, \cos\theta_{k^*}) \end{aligned} \quad (4)$$

Here, v denotes the motion step size between two iterations (not a continuous speed). If an agent detects that its next position would exceed the airspace boundaries, the direction is reversed or randomly adjusted to keep it within bounds.

3) Memory Management : From the previous sensing mechanism, a memory model is introduced to prevent premature convergence of exploiters toward a single vertiport. Such convergence results in highly localized swarm behavior and inhibits the generation of meaningful traffic flows for airspace corridor design.

Each exploiter i maintains a memory dictionary $\mathcal{M}_i = \{v_j \mapsto m_{i,j}\}$, where each vertiport v_j, as defined in Sec.III-A, is associated with a cooldown timer $m_{i,j} \in [0, M_{\max}]$. The memory $\mathcal{M}_i$ is initially empty and stores a vertiport v_j when it is visited. Once in memory, a vertiport cannot be revisited by the same agent until its cooldown reaches zero. At every simulation iteration, the values $m_{i,j}$ are decremented by one. When a cooldown timer reaches zero (i.e., $m_{i,j} = 0$), the corresponding entry is removed (i.e., $\mathcal{M}_i \leftarrow \mathcal{M}_i \setminus \{v_j\}$).

At simulation start-up, a global memory limit $M_{\max}$ is computed based on the average pairwise distance between vertiports and the agent's motion step size v, as described in Algorithm 1. From this value, a per-agent food timer $\tau_{\max}$ is derived to represent the average expected time between vertiport visits as depicted in Eq.5.

$$\tau_{\max} = \left\lceil \frac{M_{\max}}{n_v} \right\rceil, \qquad \tau_i(0) = \tau_{\max} \quad (5)$$

At each iteration, if agent i (whether explorer or exploiter) reaches a new vertiport, its timer is reset: $\tau_i \leftarrow \tau_{\max}$. Otherwise, the timer decays: $\tau_i \leftarrow \max(0, \tau_i - 1)$.

Algorithm 1 Global memory limit and food timer calculation

Require: Set of vertiport positions $\{v_j\}_{j=1}^{n_v}$, motion step size v

```
if n_v < 2 then
    M_max ← 100, τ_max ← 100
else
    d_mean ← 2/(n_v(n_v−1)) Σ_{j<l} ‖v_j − v_l‖
    t_trav ← d_mean / v
    M_max ← max(100, ⌊t_trav · n_v⌋)
    τ_max ← ⌈M_max / n_v⌉
end if
return M_max, τ_max
```

C. Pheromone Field Design

This section introduces the pheromone field that coordinates the behaviour of the different agent types. Sec.III-C1 presents the pheromone deposition mechanism, while Sec.III-C2 describes the diffusion and decay processes.

1) Pheromone Deposition : Let $P(x, y, t)$ denote the pheromone intensity at time t on the discretized airspace cell located at coordinates (x, y). The agents in the swarm collectively influence this field through pheromone deposition, using role-specific rules that depend on (i) their respective food timer $\tau_i(t)$, detailed in Sec.III-B3, and (ii) the current local pheromone concentration. Deposition occurs on the agent's current cell as well as its adjacent square cells, according to the deposit size parameter Δ_{deposit}. The constant α_0 represents the base deposit amount for all agents, while $P_{\max}$ and $P_{\min}$ define the global upper and lower bounds of the pheromone field as set by the environment.

- **Trail Thickener:** Deposits only where an existing trail is already "significant" ($P > 0.15\,P_{\max}$) and reinforces it proportionally:

$$P(x,\, y,\, t+1) = \min\left(P_{\max},\; P(x,y,t) + \alpha_0(1 + \frac{P(x,y,t)}{P_{\max}})\right) \quad (6)$$

- **Exploiter:** Adds pheromone in a 5×5 neighbourhood, scaled only by the remaining food timer:

$$P(x,\, y,\, t+1) = P(x,y,t) + \max\left(\alpha_0 \frac{\tau_a(t)}{\tau_{\max}},\; P_{\min}\right) \quad (7)$$

- **Explorer:** Uses the same timer-weighted rule as above. A memory factor $m_a(t) = \sum_f M_f \,/\, (n_v M_{\max})$ is computed in code for diagnostics but is not applied in the current release; therefore, its influence is omitted here.

All agents cease depositing when $\tau_a(t) = 0$, forcing them to refresh their timers by reaching another vertiport before contributing again.

2) Diffusion and Decay : After deposition, pheromone values evolve through a diffusion and a decay process, in this order.

- **Diffusion** Pheromone diffuses to neighboring cells using a uniform convolution kernel:

$$P(x,\, y,\, t+1) = \frac{1}{(2d+1)^2} \sum_{i=-d}^{d} \sum_{j=-d}^{d} P(x+i, y+j, t) \quad (8)$$

where d is the diffusion radius, determined by system parameters.

- **Decay** At each iteration the pheromone value decays exponentially:

$$P(x, y, t+1) = \delta\, P(x, y, t), \qquad \delta \in (0, 1) \quad (9)$$

D. Path Extraction

This section bridges the gap between SMA agents and the generation of the airspace corridor. Sec.III-D1 begins with the clustering of high-pheromone cells, followed by the principles of Catmull–Rom spline generation in Sec.III-D2.

1) Clustering of High Pheromone Cells : In the discretized pheromone field, all lattice points with an intensity above a fixed threshold $\tau_P = 0.7 P_{\max}$ are collected in $\mathcal{S}_{\text{high}}(t)$. The reduced cloud is then clustered with *k-means* where $k = \max\big(4,\ n_v\big)$

The vector of centroids $C = \{c_i\}_{i=1}^{k}$ is ordered by a nearest-neighbour heuristic, starting from the closest point to the center. This produces a cyclic sequence well suited for spline interpolation and works even when vertiports and no-fly zones are interlaced.

2) Catmull–Rom Spline Generation : After the k centroid points $\{C_1, \ldots, C_k\}$ have been ordered into a loop, the corridor is generated with a *Catmull–Rom spline* [27]. Catmull–Rom curves are particularly attractive for U-space design for the following reasons:

- They interpolate every control point, guaranteeing that each vertiport lies exactly on the reference path;
- They offer local support, moving one control point affects only its two adjacent segments, which simplifies real-time updates;
- They are C^1-continuous, preventing heading jumps and abrupt turn-rate changes.

Given four consecutive control points $P_{i-1}, P_i, P_{i+1}, P_{i+2}$, the location on the segment $[P_i, P_{i+1}]$ at the normalized parameter $t \in [0, 1]$ is defined through Eq.10 with $T(t) = [\, t^3\ t^2\ t\ 1\,]$. The reference matrix $M(\tau)$ and the local geometry matrix P are defined in Eq.11 where τ is the tension parameter, fixed to 0.5 to avoid overshooting.

$$S(t) = T(t)\, M(\tau)\, P, \quad (10)$$

$$M(\tau) = \begin{bmatrix} -\tau & 2-\tau & \tau-2 & \tau \\ 2\tau & \tau-3 & 3-2\tau & -\tau \\ -\tau & 0 & \tau & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, P = \begin{bmatrix} P_{i-1} \\ P_i \\ P_{i+1} \\ P_{i+2} \end{bmatrix} \quad (11)$$

Wrapping the control-point list as $\{C_{k-1}, C_k, C_1, C_2, \ldots, C_k, C_1\}$ give each segment the four required neighbours in Eq.10 at every index, so the resulting curve forms a closed C^1-continuous close circular corridor.

E. Algorithm Overview

This brief section completes the methodology specific to SMA process through a joined workflow as summarize in Algorithm 2.

Algorithm 2 SMA Based Adaptive UAV Corridor Generation

Initialize: Agents, Pheromone field P, vertiports $\{v_j\}_{j=1}^{n_v}$, no-fly zones $\{o_j\}_{j=1}^{n_o}$, others parameters SMA compiled in Table I.
for each iteration t **do**
  for each agent i **in** Agents **do**
    Select movement direction based on S_{i,k^*} (Eq.3)
    Move agent and check for boundary collisions
    Deposit pheromones according to agent type
    Update agent memory (decay, food detection)
  end for
  Apply global pheromone diffusion and decay
  Update spline extraction from pheromone trails
end for
Output: Smoothed cyclic UAV corridors

IV. Experiments

The experiment starts with the setup in Sec.IV-A and is followed by the results in Sec.IV-B.

A. Setup

This section is divided into two parts: first, the parametrization, specification, and evaluation of the SMA algorithm, presented in Sec.IV-A1; second, the description of the operational metrics and the large-scale use case scenarios, are respectively presented in Sec.IV-A2 and Sec.IV-A3.

1) Slime Mold Algorithms : To monitor the evolution of the SMA throughout the iterations, Eq.12 introduces a global score R_{global} composed of five simplified metrics which are subsequently defined. While less realistic than the metrics defined after in Sec.IV-A2, they are easy to compute when SMA generation can be intensive.

We denote $p_{\mathcal{C}_{wpt}}$, the position of a waypoint wpt on the discretized airspace corridor $\mathcal{C}$ generated with SMA (composed of n waypoints). A such airspace corridor is illustrated in Fig.3 in complement of the comprehension of both the vertiport proximity and obstacle incursion scores detailed below.

$$\begin{aligned} R_{\text{global}} &= W_V R_V + W_O R_O + W_{\text{inter}} R_{\text{inter}} \\ &\quad + W_{\text{curv}} R_{\text{curv}} + W_{\text{length}} R_{\text{length}} \\ \sum_i W_i &= 1 \end{aligned} \quad (12)$$

Each component is normalized within the range $[0, 1]$ to ensure consistency across scenarios and allow intuitive interpretation and easy tuning of the weight between the different objective.

- **Vertiport Proximity** score, denoted R_V, is detailed in Eq.13. It encourages the corridor to remain close to vertiports within a given threshold distance r_v.
- **No-fly Zone Incursion** score, denoted R_O is detailed in Eq.14. It penalises violations where the corridor enters restricted (no-fly) zones.
- **Self-Intersection** score, denoted R_{inter} is detailed in Eq.15. It penalises topologies where the corridor intersects itself.
- **Smoothness** score, denoted R_{curv} is detailed in Eq.16. It rewards corridors with lower curvature, promoting smooth and flyable paths.
- **Traffic Length** score, denoted R_{length} is detailed in Eq.17. It penalises excessive total arc length of the corridor to encourage efficiency.

$$R_V = \frac{1}{n_v}\sum_{i=1}^{n_v}\frac{d_i}{d_{\max}}, \quad d_i = \max_{wpt}(0, \|p_{\mathcal{C}_{wpt}} - v_i\| - r_v) \tag{13}$$

$$R_O = -\frac{1}{N_O}\sum_{j=1}^{N_O}\mathbb{1}_{\{\min_{wpt}(\|p_{\mathcal{C}_{wpt}} - o_j\| \le r_{o_j}\}} \tag{14}$$

$$R_{\text{inter}} = \begin{cases} -1 & \text{if the spline is self-intersecting} \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

$$R_{\text{curv}} = \frac{1}{\Delta}\sum_{i=1}^{N-1}(K_{i+1} - K_i)^2 \tag{16}$$

$$R_{\text{length}} = -\frac{d_{\mathcal{C}}}{d_{memory}} \tag{17}$$

Finally, the specifications of the SMA features seen throughout the methodology in Sec.III are provided below: Table I provides the parametrization fixed for the environment and the swarm agents regardless of their types, and Table II provides the weights that define the behavior of the different agents in the swarm.

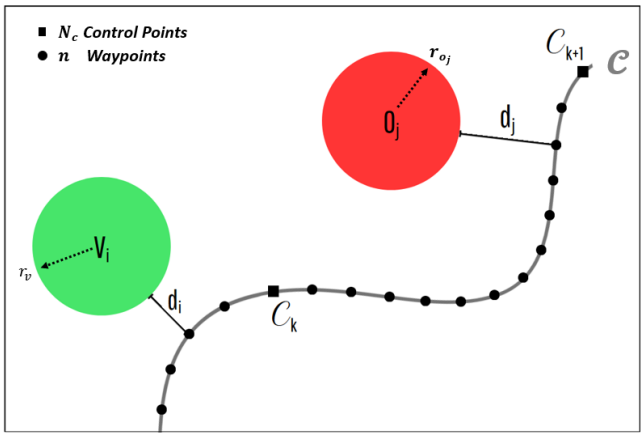


Fig. 3. A Corridor $\mathcal{C}$ is generated from Catmull-Rom spline defined by its N_c control points C_k. The figure shows a portion of the discretized corridor $\mathcal{C}$ through n waypoints and illustrates the main distance features seen in the score assignment related to vertiports and no-fly zones.

TABLE I. Specifications of the slime mold environment

Symbol	Description	Value	Unit
N^2	Discretized airspace environment (Square)	10^6	cell
n	Discretised airspace corridor (Waypoints)	500	#
N_C	Corridor control points	$n_v + n_o$	#
r_V	Vertiport radius	300	ft
T_{wall}	SMA iteration wall-time	2,000	iteration
$P_{\min}$	Minimum pheromone intensity	0.25	#
$P_{\max}$	Maximum pheromone intensity	5.0	#
$\Delta_{deposit}$	Deposit influence size	5	cell
N_{angle}	Number of angular sensor offsets per side	2	#
d_s	Agent sensor distance	45	cell
d	Agent diffusion radius	2	cell
θ_s	Agent total sensing angle	37.46	degree
v	Agent forward gap	5.0	cell
α_0	Agent base deposit amount	3.80	#
δ	Agent global decay factor	0.938	#

The parameters not defined in Table I are specific to the scenarios and assigned in Sec.IV-A3 related to these use cases.

TABLE II. Weight specifications for score assignment of agents' types in the swarm

Symbol	Description	Exploiter	Explorer	Trail Thickeners
w_p	Pheromone Weight	1.50	0.25	1.25
w_f	Food Weight	0.25	1.25	0.10
w_t	Threat Weight	-2.00	-1.50	-0.45
ϵ	Random Factor	0.50*	—— $\mathcal{U}(0.00, 0.15)$** ——	

* *Deterministic food alignment factor. It increases as the agent's heading aligns closely with the direction of the nearest unvisited food source.*
** *Uniform Distribution.*

2) Operational Metrics: We evaluate the quality of the corridors generated by the SMA using a set of operational metrics that reflect the practical consequences of such designs.

To do so, we define $\mathcal{C}$ as the corridor generated by the SMA, and $P2P$ as a point-to-point (direct) reference baseline, which connects each pair of vertiports with the shortest possible straight-line path. This P2P model represents the idealistic cost-efficient routing scenario.

The metrics used in our analysis primarily focus on the operational cost and efficiency of using $\mathcal{C}$ compared to the P2P baseline:

- $d_{P2P}(v_j, v_l)$ is the euclidean distance between two vertiports (i.e., $\|v_j - v_l\|^2$), regardless of no-fly zones or airspace structure.
- $\mathcal{C}_{\text{length}}$: is the total arc length of the corridor $\mathcal{C}$, measured along its full closed path.
- $d_{\mathcal{C}_{\text{out}}}(v_j)$ is the distance from vertiport v_j to the closest point on the corridor. This represents an operational transition distance (i.e., entry or exit).
- $d_{\mathcal{C}_{\text{in}}}(v_j, v_l)$ is the distance flown within the corridor between the nearest points to v_j and v_l.
- $d_{\mathcal{C}}(v_j, v_l)$ is the total distance of an operation from v_j to v_l via the corridor, including entry and exit transitions as described in Eq.18.

$$d_{\mathcal{C}}(v_j, v_l) = d_{\mathcal{C},\text{out}}(v_j) + d_{\mathcal{C},\text{in}}(v_j, v_l) + d_{\mathcal{C},\text{out}}(v_l) \quad (18)$$

- The extra flown distance $d_{\mathcal{C}/P2P}(v_j, v_l)$ measures the excess distance compared to the point-to-point baseline as described in Eq.19.

$$d_{\mathcal{C}/P2P}(v_j, v_l) = d_{\mathcal{C}}(v_j, v_l) - d_{P2P}(v_j, v_l) \quad (19)$$

- Inefficiency ratio $\eta_{\mathcal{C}/P2P}(v_j, v_l)$: Compares the SMA corridor cost with the ideal baseline:

$$\eta_{\mathcal{C}/P2P}(v_j, v_l) = \frac{d_{\mathcal{C}}(v_j, v_l)}{d_{P2P}(v_j, v_l)} \quad (20)$$

Additional global metrics are considered such as the cumulative and the average SMA flight distance over all vertiport pairs denoted $D_{\mathcal{C}}$ and $\hat{d}_{\mathcal{C}}$ respectively, in addition to $\max d_{\mathcal{C}}$; the maximum SMA operation distance among all pairs. These are also computed for entry/exit transitions, respectively denoted $D_{\mathcal{C}_{\text{out}}}$, $\hat{d}_{\mathcal{C}_{\text{out}}}$ and $\max d_{\mathcal{C}_{\text{out}}}$.

Other metrics, such as the "corridor usage rate" (i.e., proportion of the corridor used across all operations) and the "number of no-fly zone incursions" made by $\mathcal{C}$, are also tracked as operational constraints.

3) Large Scale Scenarios : Our approach to pass from SMA to an airspace corridor consist of keeping track of the best spline results with regard to the score function detailed in Sec.IV-A1, during a maximum number of T_{wall} iterations.

When generating a corridor $\mathcal{C}$, we duplicate it into two vertically separated corridors to accommodate clockwise and counterclockwise traffic at 300 ft and 450 ft respectively, thereby capping the VLL airspace. As the focus of this work is on airspace design and its applicability, we leave safety implications, such as tactical separation, to complementary solutions to be explored in future studies.

Fig.4 presents the three large-scale scenarios used to demonstrate the SMA approach. Scenario I in Fig.4a introduces a baseline case with a random distribution of vertiports and no-fly zones. Scenario II in Fig.4b is a densified version of Scenario I, featuring a higher concentration of vertiports and obstacles. Scenario III in Fig.4c proposes a uniformly distributed set of vertiports over a larger airspace, with the area increasing from 10 km^2 to 15 km^2. The proportion of no-fly zones in each scenario is 1.96%, 3.62%, and 1.48% respectively.

For convenience, we denote $\mathcal{C}_I$, $\mathcal{C}_{II}$, and $\mathcal{C}_{III}$ as the corridors generated by the SMA for the three respective scenarios.

B. Results

1) SMA Stability and Convergence : Fig.5 depicts the evolution of the global swarm score R_{global} regarding the respective scenarios, in addition to the score of the Scenario II when initialized with the Scenario I in purple color.

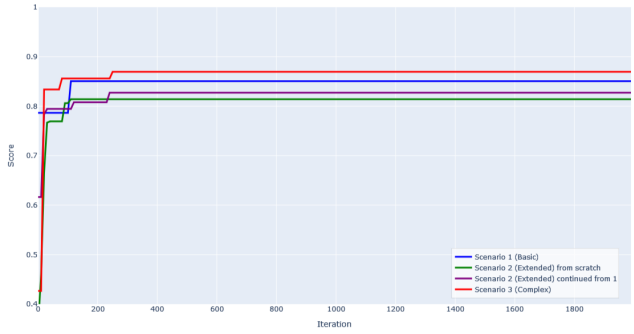
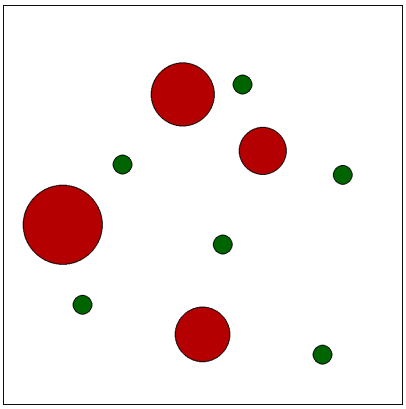
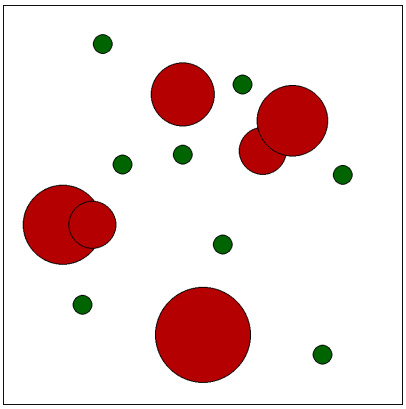


Fig. 5. Evolution of the global score R_{global} of the SMA generating $\mathcal{C}_I$, $\mathcal{C}_{II}$, and $\mathcal{C}_{III}$ for the respective scenarios.

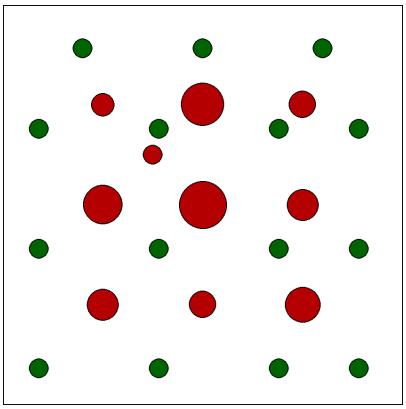
At the exception of the previous extended Scenario II, all other scenarios have been initiated with an empty



(a) Scenario I — Sparse layout considering an airspace of 10 km^2. ($n_v = 6$, $n_o = 4$)



(b) Scenario II — Extended layout considering a denser airspace of 10 km^2. ($n_v = 8$, $n_o = 6$)



(c) Scenario III — Cluttered airspace of 15 km^2. ($n_v = 15$, $n_o = 10$)

Fig. 4. Configurations of the three large-scale use case airspace scenarios. These are designed to challenge the adaptivity of the SMA: Scenario I to Scenario II tests rapid recovery from topological changes, while Scenario III evaluates convergence behavior in a homogeneous environment.

pheromone distribution.

Overall, the score function demonstrates rapid convergence, reaching values above 0.8 within approximately 300 iterations. Notably, the score of $\mathcal{C}_{III}$ outperforms the others, followed by $\mathcal{C}_I$, which shows a sharp increase around iteration 100. Scenario II presents the most significant challenge showing a slower and more gradual improvement of the swarm behavior to generate $\mathcal{C}_{II}$. Fortunately, a moderate boost in performance is observed when the SMA is initialized with $\mathcal{C}_I$.

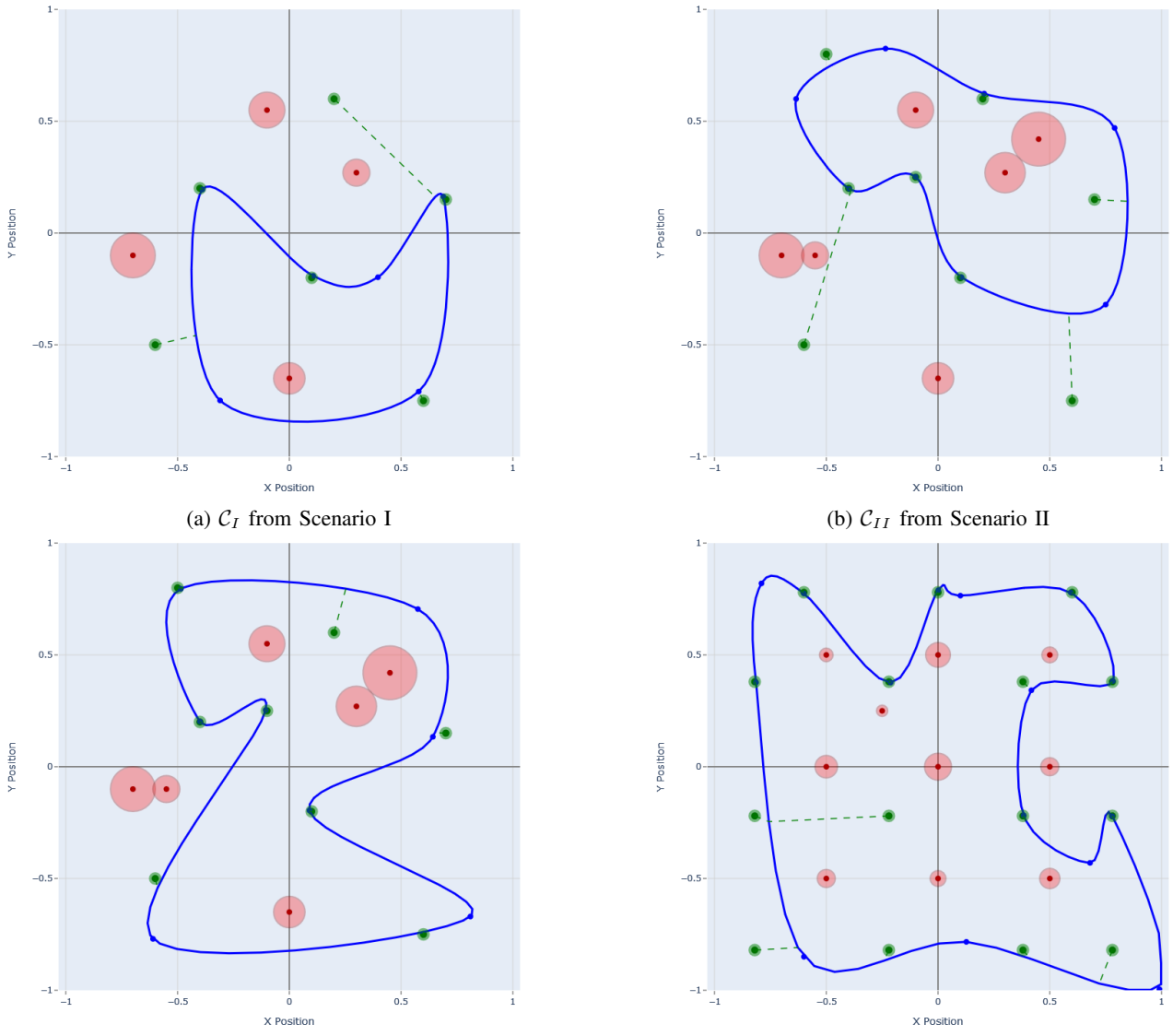
2) Large Scale Scenarios : Fig.6 compiles the airspace corridors generated from SMA swarm agents after the iterations described in Sec.IV-B1 for the respective scenarios.

These observations are further supported by the comparative analysis presented in Table III. The generated corridors exhibit no visible incursions into no-fly zones, neither along the main corridor paths nor within the operational transitions between vertiports and corridors. Likewise, no self-intersections were detected in any of the generated corridors.

Across scenarios, the agents tend to exploit the airspace corridor by steering as close as possible to the vertiports, which results in longer routes and sharper turns. Fortunately, the resulting trajectories remain operationally realistic.

Some vertiports, however, remain relatively isolated from the main corridor paths such as the northern vertiport in Fig.6a, two southern vertiports in Fig.6c, and several central vertiports in Fig. 6d, where the generated corridor $\mathcal{C}_{III}$ follows a more peripheral route.

From an operational point of view, Scenario II demonstrates notable cost-efficiency, with 50.00% of operations



(a) $\mathcal{C}_I$ from Scenario I

(b) $\mathcal{C}_{II}$ from Scenario II

(c) $\mathcal{C}'_{II}$ from Extended Scenario II

(d) $\mathcal{C}_{III}$ from Scenario III

Fig. 6. Airspace corridors generated by the SMA for each scenario, corresponding to the highest scoring iterations discussed in Sec.IV-B1.

TABLE III. Comparative Analysis on Operational Aspects of the Scenarios.

Metric	Scenario		
	I	II	III
Vertiports n_v	6	8	15
No-fly zones n_o	4	6	10
No-fly zones incursions	0	0	0
$\mathcal{C}$ usage rate	100.00%	100.00%	100.00%
$\mathcal{C}_{length}$	21.30 km	22.00 km	61.27 km
$\hat{d}_{\mathcal{C}_{\text{out}}}$	1.26 km	1.38 km	0.81 km
$\hat{d}_{\mathcal{C}}$	3.59 km	3.44 km	8.35 km
$\max d_{\mathcal{C}_{\text{out}}}$	2.89 km	4.25 km	3.40 km
$\max d_{\mathcal{C}}$	12.22 km	14.05 km	34.06 km
$D_{\mathcal{C}_{\text{out}}}$	7.59 km	11.08 km	12.13 km
$D_{\mathcal{C}}$	25.88 km	44.06 km	250.68 km
$\hat{\eta}_{\mathcal{C}/P2P}$	0.125	0.078	0.084
$1.00 < \eta_{\mathcal{C}/P2P}$	20.00%	17.90%	49.50%
$0.65 < \eta_{\mathcal{C}/P2P} \leq 1.00$	40.00%	17.90%	14.30%
$0.50 < \eta_{\mathcal{C}/P2P} \leq 0.65$	13.30%	14.30%	3.80%
$0.30 < \eta_{\mathcal{C}/P2P} \leq 0.50$	13.30%	39.30%	13.30%
$\eta_{\mathcal{C}/P2P} \leq 0.30$	13.30%	10.70%	19.00%

exhibiting an inefficiency below that 0.50, compared to 26.60% in Scenario I and 32.30% in Scenario III. However, Scenario II also presents the highest average transition distance $\hat{d}_{\mathcal{C}_{\text{out}}}$, suggesting that the corridor tends to pass closer to certain vertiports, thereby increasing access distance for others. $\mathcal{C}_{III}$ exhibits a significant drop in efficiency for certain vertiport connections, with flight distances reaching up to 34.00 km, compared to a maximum of 12.22 km for $\mathcal{C}_I$ and 14.05 km for $\mathcal{C}_{II}$, respectively. Assuming the corridor accommodates operations between all possible vertiport pairs, every waypoint along the corridor would be visited at least once, as confirmed by the high usage rate.

V. Discussion

The SMA-based approach benefits from a strong initial convergence due to the high pheromone concentration around vertiports, which acts as a strong attractor for agents early in the process. This results in promising initial scores. The scoring system is designed to prioritize connectivity and obstacle avoidance. Connectivity is evaluated using the distance to the vertiport radius, not just the centroid, introducing a stricter criterion. While one could argue this contradicts the agents' tendency to seek vertiport centers, we maintain this formulation to ensure a common and rigorous evaluation ground across scenarios.

One key strength of the approach is its adaptivity. When Scenario II is initialized with the pheromone distribution from Scenario I (simulating a change in topology), convergence to a high-quality corridor is both faster but more connected with better corridor shapes. This suggests strong potential for real-time reactivity in dynamic environments. Furthermore, increasing the number of spline control points N_C to 15 does not hinder convergence, as shown in Scenario III. High global scores (> 0.8) are reached in roughly the same number of iterations across scenarios which suggest potential scalability and robustness in using SMA for airspace design.

The proposed metrics serve both as a benchmarking tool and as a proxy for practical performance in real deployments. Even though the SMA does not explicitly optimize for shortest paths, the extra-flown distance and inefficiency scores remain within acceptable margins. If higher flight speeds are authorized within the corridor, as is expected in real-world implementations, the longer distances may be compensated by improved throughput and regulatory compliance.

Comparative analysis across the three scenarios reveals interesting trade-offs. Scenario I, being sparse with minimal topological complexity, allows rapid convergence and efficient generation of $\mathcal{C}_I$. The best solution, found after 112 iterations, achieves a global score of 0.82, with a compact spline defined by just four control points. However, connectivity is occasionally limited, especially for more isolated vertiports.

Scenario II increases the complexity with a denser distribution of vertiports and no-fly zones. When initialized from scratch, convergence is slower, and the corridor is less optimal. Yet, when bootstrapped from Scenario I's pheromone map, results significantly improve, supporting the claim that our SMA approach is reactive and reconfigurable. The corridor shape is cleaner, vertiport access is improved, and fewer iterations are needed. This adaptability reinforces the SMA's suitability for environments with evolving constraints.

Scenario III challenges the algorithm further by extending the airspace and distributing vertiports uniformly. Despite the increased clutter and tighter margins, the resulting corridor (found in 228 iterations) avoids obstacles effectively and connects nearly all vertiports. The corridor length is significantly higher, around 61 km (scaled to a 15 km area), compared to approximately 22 km in Scenarios I and II. This is expected due to the more complex layout. However, it is notable that no portion of the corridor remains unused (i.e., all waypoints are visited), and safety margins are preserved.

Overall, Scenarios I and II demonstrate the highest efficiency in terms of extra flown distance, suggesting they are more cost-effective for operational planning. Scenario III, while less efficient, offers better coverage and no-fly zones avoidance in constrained environments. It provides a safer and more comprehensive solution, assuming onboard or systemic deconfliction mechanisms within the corridor.

The SMA method exhibits strong adaptability, robustness across scales, and favourable operational characteristics, making it a viable candidate for real-world airspace corridor planning in both sparse and dense urban environments.

On the limitation aspect, this work focuses on a simplified environment with static vertiport and no-fly zone locations, and does not account for dynamic factors such as evolving risk areas. Although the pheromone-based agents converge quickly, the scalability to dense urban settings with many vertiports remains to be assessed. The

method also relies on empirically tuned heuristic weights, which may not generalize well across scenarios. Automated calibration could improve robustness. Finally, while real-time applicability is claimed, execution time was not systematically evaluated. Future work should include runtime profiling and comparisons with airspace design methods to assess suitability for flight planning authorisation.

Ongoing research focuses on deploying operations within a mixed-reality environment to serve as a proof of concept for the cyclic airspace corridor, while progressively integrating additional service provisions. Although the current study considers cyclic corridors, the SMA approach naturally enables more complex structures, including intersecting and branching segments, which can be further investigated to refine airspace efficiency and make the design more competitive with the state-fo-the-art. Future work will explore the dynamic nature of the airspace environment and evaluate the responsiveness of the SMA in such contexts. Additionally, Reinforcement Learning will be integrated with the SMA to fine-tune its parametrization and stress-test the resilience of the approach in more demanding operational scenarios.

VI. Conclusion

This work introduces a bio-inspired approach for generating cyclic airspace corridors in simulated urban environments, with an emphasis on adaptivity to evolving vertiport networks and no-fly zones. Building on the Slime Mold Algorithm, our method exploits the natural path-forming behavior of biological swarms, where agents interact through a discretized pheromone field. By clustering high pheromone concentrations and fitting a Catmull–Rom spline, we generate smooth, continuous corridors that ensure connectivity between vertiports while avoiding obstacles.

Through multiple urban configurations with varying numbers of vertiports and restricted areas, we demonstrate that the resulting corridors offer strong performance in terms of connectivity, operational feasibility, and design simplicity, highlighting their potential as lightweight alternatives to current UAM infrastructure.

The proposed method shows fast convergence, robustness to environmental changes, and extensibility toward hybrid architectures. Future work will involve validating this approach in mixed-reality or real-world experiments, integrating real-time traffic dynamics and operational constraints for UTM applications.

References

[1] R. J. Hansman and P. Vascik, "Constraint Identification In On-Demand Mobility For Aviation Through An Exploratory Case Study Of Los Angeles," *ICAT Report*, 2018.

[2] Amazon Prime Air, "Revising the Airspace Model for the Safe Integration of Small Unmanned Aircraft Systems," 2015.

[3] D. Geister and B. Korn, "Concept for Urban Airspace Integration, *DLR Blueprint*," 2017.

[4] B. Lascara, A. Lacher, M. DeGarmo, D. Maroney, R. Niles, and L. Vempati, "Urban air mobility airspace integration concepts," *The Mitre Corporation*, 2019.

[5] SESAR Joint Undertaking, "European ATM Master Plan," Publications office of the European Union, Luxembourg, 2025.

[6] National Aeronautics and Space Administration, "UTM: Air Traffic Management for Low-Altitude Drones," 2015.

[7] K. Balakrishnan, J. Polastre, J. Mooberry, R. Golding, and P. Sachs, "Blueprint for the sky: The roadmap for the safe integration of autonomous aircraft," *Airbus UTM, San Fransisco, CA*, 2018.

[8] K. Low, L. Gan, and S. Mao, "Preliminary Study in Managing Safe and Efficient Low-Altitude UAV Operations," *Air Traffic Management Research Institute, School of Mechanical and Aerospace Engineering Nanyang Technological University*, 2014.

[9] D.-S. Jang, C. A. Ippolito, S. Sankararaman, and V. Stepanyan, "Concepts of Airspace Structures and System Analysis for UAS Traffic flows for Urban Areas," AIAA SciTech Forum, 2017.

[10] C. L. Tallec, P. L. Blaye, and M. Kasbari, "Low Level RPAS Traffic Management (LLRTM) Concept of Operation," 17^{th} AIAA Aviation Technology, Integration, and Operations Conference, 2017.

[11] Federal Aviation Administration, "Urban Air Mobility, Concept of Operations," Release, Version 2, 2023.

[12] SESAR Joint Undertaking, "U-space ConOps and architecture, Edition 4," 2023.

[13] R. J. Bauranov A, "Designing airspace for urban air mobility: A review of concepts and approaches," *Progress in Aerospace Sciences, vol. 123*, 2021.

[14] E. Sunil, J. Hoekstra, J. Ellerbroek, F. Bussink, D. Nieuwenhuisen, A. Vidosavljevic, and S. Kern, "Metropolis: Relating Airspace Structure and Capacity for Extreme Traffic Densities," *11^{th} USA/EUROPE Air Traffic Management R&D Seminar, FAA & Eurocontrol*, 2015.

[15] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: A new method for stochastic optimization," *Future Generation Computer Systems*, vol. 111, pp. 300–323, 2020.

[16] Y. Ma, Z. Zhang, M. Yao, and G. Fan, "A Self-Adaptive Improved Slime Mold Algorithm for Multi-UAV Path Planning," *Drones*, vol. 9, no. 3, p. 219, Mar. 2025.

[17] J. Huang, Y. Chen, A. Heidari, L. Liu, H. Chen, and G. Liang, "Enhancing slime mould algorithm for engineering optimization: leveraging covariance matrix adaptation and best position management," *Journal of Computational Design and Engineering*, vol. 11, no. 4, pp. 151–183.

[18] L. Gao and H. Ren, "Multipath planning based on multimodal optimization slime mold algorithm," *International Conference on Culture-Oriented Science & Technology (CoST), IEEE Publisher*, pp. 50–54, 2024.

[19] F. Schweitzer, "Tracks and Trail Formation in Biological Systems," *Springer Series in Synergetics, Brownian Agents and Active Particles*, pp. 203–245, 2003.

[20] W. Wei, M. Cao, F. Ma, P. Ji, and X. Wang, "Research on path planning for UAV based on improved ant colony algorithm," in *Proceedings of the 2023 6th International Conference on Signal Processing and Machine Learning*. Association for Computing Machinery, 2023, pp. 194–200.

[21] C. Zhang and Q. Feng, "Research on UAV Path Planning Combined with Ant Colony and A*," in *Advances in Guidance, Navigation and Control*. Springer Nature, 2023, pp. 1228–1236.

[22] T. Stützle and M. Dorigo, "Ant Colony Optimization," *Technische Universität Darmstadt, Ph.D. Thesis, Habilitationsschrift, Teil 2*.

[23] M. Dorigo, E. Bonabeau, and G. Theraulaz, "Ant algorithms and stigmergy," *Future Generation Computer Systems*, vol. 16, no. 8, pp. 851–871, 2000.

[24] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, pp. 1942–1948, 1995.

[25] P. Trojovský and M. Dehghani, "Pelican Optimization Algorithm: A Novel Nature-Inspired Algorithm for Engineering Applications," *Sensors, MDPI*, vol. 22, no. 3, p. 855, 2022.

[26] X. Fan, W. Sayers, S.-j. Zhang, Z. Han, L. Ren, and H. Chizari, "Review and Classification of Bio-inspired Algorithms and Their Applications," *Journal of Bionic Engineering*, vol. 17, pp. 611–631, 2020.

[27] C. Yuksel, S. Schaefer, and J. Keyser, "Parameterization and applications of Catmull–Rom curves," *Computer-Aided Design*, vol. 43, no. 7, pp. 747–755, 2011.