In: 2022 International Conference on Unmanned Aircraft Systems (ICUAS), 21-24 June 2022, Dubrovnik, Croatia
DOI: 10.1109/ICUAS54217.2022.9836139

Strategic Conflict Management for Performance-based Urban Air Mobility Operations with Multi-agent Reinforcement Learning

Cheng Huang[1], Ivan Petrunin[1] and Antonios Tsourdos[1]

***Abstract*— With the urban air mobility (UAM) quickly evolving, the great demand for public airborne transit and deliveries, besides creating a big market, will result in a series of technical, operational, and safety problems. This paper addresses the strategic conflict issue in low-altitude UAM operations with multi-agent reinforcement learning (MARL). Considering the difference in flight characteristics, the aircraft performance is fully integrated into the design process of strategic deconfliction components. With this concept, the multi-resolution structure for the low-altitude airspace organization, Gaussian Mixture Model (GMM) for the speed profile generation, and dynamic separation minima enable efficient UAM operations. To resolve the demand and capacity balancing (DCB) issue and the separation conflict at the strategic stage, the multi-agent asynchronous advantage actor-critic (MAA3C) framework is built with mask recurrent neural networks (RNNs). Meanwhile, variable agent number, dynamic environments, heterogeneous aircraft performance, and action selection between speed adjustment and ground delay can be well handled. Experiments conducted on a developed prototype and various scenarios indicate the obvious advantages of the constructed MAA3C in minimizing the delay cost and refining speed profiles. And the effectiveness, scalability, and stabilization of the MARL solution are ultimately demonstrated.**

I. INTRODUCTION

Urban Air Mobility (UAM) is an evolving air transport system for the transit of passengers and delivery of goods in dense urban areas and comes up with a beautiful blueprint for the envisioning of the smart city. With the demand growing, the traffic flow in metropolitan low-altitude airspace will lead to a fundamental challenge for safety and efficiency. To ensure safe and secure flights in metropolitan airspace, it is critical to efficiently manage all trajectories in case of any conflicts with other users.

Following the similar architecture in current Air Traffic Management (ATM) and Unmanned Aircraft System (UAS) Traffic Management (UTM), where ATM is assisted by the human air traffic controller and UTM is supported by UAS Service Supplier (USS) [1] [2] [3], the network of Provider of Services to UAM (PSU) in UAM provides the necessary services (separation, communications, information exchange, etc) under rules and regulations established by the authority [4].

Prior to the operation in the UAM Operations Environment (UOE), PSU must process the submitted operation plans of all aircraft from fleet operators. The aircraft type, expected flight path, departure time, arrival time, and other information about communication are elements in the plan for PSU to provide efficient and safe operation suggestions to the fleet operators or aircraft [4].

Conventionally, to improve the efficiency of urban air traffic flow, conflict-free traffic management is mainly achieved by the three-layer solution: strategic conflict management, separation provision, and collision avoidance [5]. The structure is also fit for UAM with some transitions. Because of the high-dynamic demand, not like a half year or several months in advance in ATM, the strategic conflict management for UTM or UAM begins several days or weeks before the execution of the operations [6]. Strategic conflict management constitutes airspace organization and management, demand and capacity balancing (DCB), and traffic synchronization [5]. The airspace organization and management provide efficient, dynamic, and flexible airspace resources and services for users. And the DCB component in the cycle solves the issue that demand exceeds the capacity and eliminates potential conflicts. Finally, the traffic synchronization part cooperates with the other components to improve the efficiency of the traffic flow, reduce the risk of conflicts and relieve the stress of the pre-tactical and tactical phases.

As the element in strategic conflict management, the airspace structure in UAM is different from the current high-altitude airspace architecture. Considering the high freedom and flexibility rules, the stacked grids [7] construct typical airspace structures for UAM, such as the AirMatrix [8] distributed over the entire low-altitude space of the city, and the corridor networks over buildings or roads. Since the size of stacked blocks can affect the airspace complexity and operation safety, the AirMatrix can be divided into different resolutions at different altitudes and areas around the buildings and populated regions [9]. On the basis of the grid structure, many efficient methods such as A* are used to generate conflict-free trajectories [10].

And here two options for conflict management are provided. One is considering the effect of conflict and generating the conflict-free trajectories directly; the other one is following the traditional way to iterate with initial conflicted trajectories to get the optimal result. However, the potential issue for the first measure is that the UAM operation environment (UOE) is always dynamic, and it is hard to guarantee the efficiency to re-generate trajectories for high-frequency requests. In contrast, when employing the second method, it can still provide the feasibility of fine-tuning some parts of the trajectory.

The heart of the fine-tuning process is to resolve the DCB issue and the separation conflict. For the DCB issue

[1]Cheng Huang, Ivan Petrunin and Antonios Tsourdos are with the School of Aerospace, Transport and Manufacturing, Cranfield University, Cranfield, MK43 0AL, UK {cheng-huang.huang, i.petrunin, a.tsourdos}@cranfield.ac.uk

in the conventional ATM field, it is usually formulated with 0-1 integer programming [11] or Eulerian-Lagrangian [12] models to obtain optimal ground delay, rerouting or airborne holding actions. One obvious shortcoming is the lack of ability to cope with flexible environments. To solve this issue, multi-agent reinforcement learning (MARL) is widely studied in ATM to allow intelligent agents to interact with the dynamic environment to resolve the DCB issue. The system can be constructed with the network structure, in which agents with interactions are defined as "peers" and connected for information propagation. With this definition, the edge-based and agent-based reinforcement learning leverage the coordination graph to solve the DCB issue [13]. And to enable collaboration among multiple agents, the hierarchical reinforcement learning formulates state-action abstraction and temporal action abstraction to resolve the congestion issue [14]. Even more, unsupervised learning and supervised learning are integrated with MARL to improve the cooperation of agents [15].

Behind these studies, the trajectory-based operation (TBO) allows different strategies to manage the trajectories effectively. Whereas these strategies are not applicable for UAM, because of the different airspace structures and operation requirements. The congestion problem caused by manned or unmanned aircraft in metropolitan regions, which is critical for air or ground operation efficiency and safety, is seldom investigated. There is only one effective attempt till now to apply reinforcement learning, specifically, a deep Q-Learning network (DQN) with a genetic algorithm (GA) to the UAM system and generate a feasible solution for the congestion problem [16].

As for the separation conflict problem, the studies tend to change the speed direction straightforwardly, according to the converging and diverging status. To achieve this, the traditional mathematical methods, such as velocity obstacles [17] and Mixed Integer Linear Programming (MILP) [18], even reinforcement learning methods, e.g. those using single-agent deep deterministic policy gradient (DDPG) for pairwise aircraft [19] and multiple aircraft [20], are utilized to learn when and the specific values for aircraft to change headings. It is reasonable for those flights to perform a heading turn in broad airspace. But in dense low-altitude UOE, it is risky to provide only rough information of change heading or altitude, even if the command can be sent in high-frequency. Due to this, a detailed and elaborate plan is necessary for the UAM.

In this paper, multi-resolution airspace structure organization and performance-based operation are developed for strategic conflict resolution along with the MARL. Assuming that each basic air block can be occupied by only one aircraft at each time, we then merge the DCB issue and separation conflict into one meta-problem. All components for strategic conflict management consider the heterogeneous features of aircraft performance, for example, the aircraft size, max flight speed and max hovering time, etc. To improve the realistic operation and mitigate the risk, the environment for MARL is fully adapted for UAM, where both speed adjustment and ground delay are regarded as effective actions for conflict resolution. The MARL is designed with variable agents and mask recurrent structure for better action selection, as a result, improving the ability to handle the dynamic environment in UAM.

The contributions of this paper are summarized as follows:

- Multi-resolution structure is built for efficient low-altitude airspace organization according to the various size of small aircraft.
- A Gaussian Mixture Model (GMM) based approach for extraction of detailed speed information is proposed and offers the possibility for speed adjustment through refining GMM parameters.
- The Multi-Agent Asynchronous Advantage Actor-Critic (MAA3C) framework with mask recurrent structure is implemented and evaluated for the first time to select an appropriate action between ground delay and speed change at each step.
- The way of enabling performance-based operation in MARL is proposed and implemented by taking the aircraft performance into consideration in the learning process.

The rest of the paper is organized as follows: Section II introduces the fundamental components in strategic conflict management and defines the conflict for UAM. The proposed MAA3C framework is illustrated in Section III. Section IV develops a small prototype and analyses the results of some study cases. Section V concludes the paper.

II. PROBLEM STATEMENT

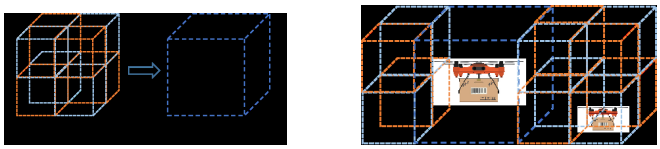
In this section, basic components in strategic conflict management are modelled, in which the low-altitude airspace construction and trajectory generation contribute to the definition of conflict.

A. Low-altitude Urban Airspace Organization

The first kernel aspect in strategic conflict management is managing the airspace. The adaptive structure such as AirMatrix [8] and grid-based solution [21] are good practices for low-altitude urban airspace discretization. As the operational aircraft varies in shape and size, instead of constantly using basic blocks, the blocks will merge into the actual operational block for larger aircraft, as in Fig. 1a, to achieve the condition that each block is only occupied by one aircraft per second. In this way, as displayed in Fig. 1b, the practical operation structure is composed of operation blocks with multiple resolutions and thus enables performance-based operation. The discrete grids are constructed as graph structure $G(V,E)$, where V is the set of all blocks, and E represents the connection of neighbor blocks.

B. Trajectory Formulation

Trajectory based operation (TBO) is an efficient method for strategic planning. Traditionally, the trajectory of a flight is formulated by sequence of positions and air speeds, which can be written as: $traj_f = \{(t_i, node_i, v_i)\}_{i=0}^{N_f-1}$, where $node_i \in \mathbb{R}^3$, $v_i \in \mathbb{R}_+$ and N_f is the number of traversed blocks of



(a) Blocks merging (b) Multiple resolutions

Fig. 1: Multi-resolution blocks.

flight f. Each flight has its operation plan represented by a spatio-temporal trajectory.

1) Spatial nodes: The spatial information of a trajectory is encoded with discrete nodes (center of blocks). To generate the list of traversed nodes with different resolutions, the aircraft performance database $AC = \{ac_i\}_{i \in N_a}$ must be built for retrieval. The shortest paths between two desired vertiports are filtered with the assigned flight level (FL). To be precise, the spatial traversed blocks of a trajectory are generated by Algorithm 1 as below.

Algorithm 1: Trajectory Generation

1 Select one type of aircraft ac_i from the aircraft performance database $AC = \{ac_i\}_{i \in N_a}$;
2 Random select the origin vertiport Ori and destination vertiport Des from vertiport database VT;
3 Merge elementary blocks based on the size of ac_i and get new airspace graph $G'(V', E')$;
4 Assign a cruise level FL ;
5 Remove nodes not in this flight level FL from airspace graph $G(V, E)$ and obtain filtered graph $G(V', E')$;
6 Generate shortest path between two vertiports: $\{node_i\}_{i=0}^{N_f-1} = Dijkstra(Ori, Des, G(V', E'))$;

2) Temporal profile: Not like the wing aircraft, the UAV or electric vertical take-off and landing (eVTOL) aircraft have the ability to hover in mid-air while necessary. Considering the aircraft performance characteristics, the Gaussian Mixture Model (GMM) is utilized to model the speed profile of small aircraft, and the probabilistic density function (PDF) of GMM is denoted as:

$$\begin{cases} p(x) = \sum_{k=1}^{K} \phi_k \mathcal{N}(x \mid \mu_k, \sigma_k) \\ \mathcal{N}(x \mid \mu_k, \sigma_k) = \frac{1}{\sigma_k \sqrt{2\pi}} exp\left(-\frac{(x-\mu_k)^2}{2\sigma_k^2}\right) \\ \sum_{k=1}^{K} \phi_k = 1 \end{cases} \tag{1}$$

where the GMM is composed of K individual normal distribution, μ_k and σ_k are the mean and variance of the Kth component. The continuous profile, as displayed in Fig. 2, is then discretized along with traversed block number N_f, providing that the speed is constant inside each block. The time and speed information are stored in each counterpart. Finally, the speed profile $\{v_i\}_{i=0}^{N_f-1}$ is obtained from the GMM model.

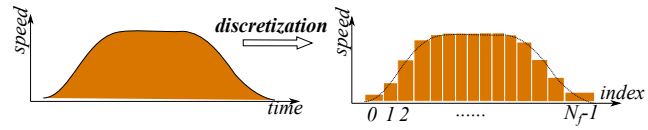


Fig. 2: Discretize time-speed graph.

With all traversing blocks $\{node_i\}_{i=0}^{N_f-1}$, speed profile $\{v_i\}_{i=0}^{N_f-1}$ and initial departure time t_0, the entry time into each block can be calculated by:

$$t_i = t_{i-1} + dim_{block}/v_i \ \ (1 \le i \le N_f - 1) \tag{2}$$

Where dim_{block} is the dimension of the block.

C. Strategic Conflict Problem

The strategic conflict problem comprises the DCB issue and the separation conflict. For the loss of separation, the dynamic separation minima [6] is defined with the nearest points and the max relative distance of the interacted trajectories $node_j$ and $node_k$:

$$dist(node_j - node_k) < (\delta v_{max} \times \tau) \tag{3}$$

where δv_{max} is determined by the max relative speed of aircraft, and τ is the temporal margin.

For the DCB issue, we define that each block can only be taken by one aircraft, and in other words, the capacity of every block is 1. To calculate the demand, we first divide the operation time horizon (H hours) into SN pieces. During each split period $sn \in \{0, 1, \cdots, SN-1\}$, we count the number of flight whose entry time t_i $(i \in [0, N_f - 1])$ locates in this period and get its block index i. All related blocks will correspondingly record how many flights are traversing during this period and denoted by $d_{node_i,sn}$. From the other view, the DCB issue here can be presented by the loss of separation of any pairwise flight. The DCB function is that:

$$node_j == node_k \tag{4}$$

Where $node_j$ and $node_k$ are traversing blocks of two different flights during the same period. This equation can also be rewritten as:

$$dist(node_j - node_k) < dim_{block} \tag{5}$$

We always set $(v_{max} \times \tau) >> dim_{block}$, and can notice from Eq. (3) and Eq. (5) that the separation conflict (C) event and DCB event (D) subject to $D \subseteq C$. It means that if conflict can be resolved, the DCB issue is eliminated at the same time. Therefore, the remaining content views the DCB issue and conflict as the same problem.

To further illustrate the steps for strategic conflict resolution, two sample trajectories are presented: $traj_1 = \{\ldots, (101\ s, B10, 4\ m/s), (110\ s, B15, 2\ m/s), \ldots\}$ and $traj_2 = \{\ldots, (102\ s, B10, 3\ m/s), (108\ s, B16, 2\ m/s), \ldots\}$. We can observe that during a 5-second period from 100 s to 105 s, both trajectories will enter the block $B10$. The demand of this block during this period $d_{B10,20} = 2 > 1$. To resolve this issue, effective measures are supposed to be

taken. In this paper, we consider leveraging the flexibility of UAVs' flight performance to fine-tune the airborne speed profile as action and performing ground delay as another option. As depicted in Fig. 3, the speed change revises parameters in GMM, and in the meantime, ground delay can directly shift the entire trajectory in the time dimension in the event of a potential conflict. In some circumstances, both actions can be taken to resolve conflicts and this process is repeated until all conflicts are eliminated. In practical applications, the continuous curves are all discretized as in Fig. 2.

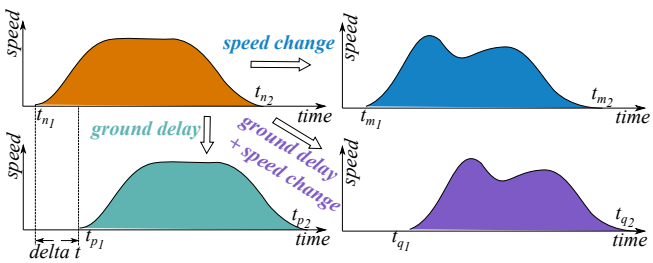


Fig. 3: Time-speed graphs based on GMM.

III. MULTI-AGENT REINFORCEMENT LEARNING FRAMEWORK

The strategic deconfliction problem is formulated as a Partially Observable Markov Decision Process (POMDP) in this section, in which the involved components and performance-based factors are explained in detail. To a great extent, the multi-agent asynchronous advantage actor-critic (MAA3C) framework is implemented with the mask recurrent structure.

A. POMDP

The POMDP is described as: an agent i $(i \in \mathcal{N})$ in agent set $\mathcal{N} = \{1, \cdots, N_a\}$ observes the local information $\mathcal{O}^i$ from global environment state $\mathcal{S}$ according to the observation equation $\mathcal{S} \times \mathcal{N} \longrightarrow \mathcal{O}$. This function reveals the kernel feature of partial observations. The agent then samples its action from the action space $\mathcal{A}^i$, therefore the environment is transitioned based on the probability: $\mathcal{P} : \mathcal{S} \times \mathcal{A} \longrightarrow \mathcal{S}$, where $\mathcal{A} = \mathcal{A}^1 \times \cdots \times \mathcal{A}^N$ is the joint action of all agents. After performing this actions, the action of individual agent i is assessed by the reward $\mathcal{R}^i : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \longrightarrow \mathcal{R}$. Therefore, the tuple $\left(\mathcal{N}, \mathcal{S}, \{\mathcal{A}^i\}_{i\in\mathcal{N}}, \mathcal{P}, \{\mathcal{R}^i\}_{i\in\mathcal{N}}, \gamma, \{\mathcal{O}^i\}_{i\in\mathcal{N}}\right)$ concludes the necessary parameters in POMDP [22]. The detailed components that forms the whole process of POMDP are illustrated as follows:

1) Agent: Each flight has its unique trajectory that contains the traversed blocks, entry time and flight speed. But not all trajectories are included in the agent set, only flight trajectories that are involved in conflicts are regarded as agent. The agent number varies with the steps evolving, since the speed refinement or ground delay might change the conflict number.

2) Observation: Because of the partial observation attribute, the agent is not able to access the global state $\mathcal{S}$. Only local information about its traversed blocks, flight information, and conflict status can be observed. The agent's observation might be influenced by other agents through the variables in shared blocks or conflict. In detail, three indicators including the index list $\{node_i\}_{i=0}^{N_f-1}$ and conflict status list $\{C_i\}_{i=0}^{N_f-1}$ of all traversed blocks as well as entry time list $\{t_i\}_{i=0}^{N_f-1}$ consists of the observations of the agent, and can be given by a matrix:

$$O_i = \begin{bmatrix} node_0 & c_0 & t_0 \\ node_1 & c_1 & t_1 \\ \vdots & \vdots & \vdots \\ node_{N_f-1} & c_{N_f-1} & t_{N_f-1} \end{bmatrix}_{(N_f \times 3)} \tag{6}$$

3) Action: At each learning step st, every agent samples an action a_{st}^i from the action set $\mathcal{A}^i$. Specifically, the $\delta\mu_k, \delta\sigma_k$ $(k \in [1, K])$ in GMM model and the ground δt are candidate actions. At each step, only one type of actions is able to be performed.

Combined with Fig. 3, if speed change is selected as the action, the generated result will be used to generate new GMM parameters with $\mu_i' = \mu_i + \delta\mu_i$ and $\sigma_i' = \sigma_i + \delta\sigma_i$. As the consequence, the speed profile as well as entry time can be updated with Eq. (2). Finally, the trajectory $traj_f' = \{(t_i', node_i, v_i')\}_{i=0}^{N_f-1}$ is refreshed.

While the ground delay is chosen as the action, the whole trajectory is temporally shifting by δt, and the new trajectory $traj_f' = \{(t_i + \delta t, node_i, v_i)\}_{i=0}^{N_f-1}$ can be obtained.

Besides that, the specific mechanism needs to be determined to select the action.

4) Reward Shaping: The performance of an action is evaluated by the return value. All agents collaborate to minimize the total cost, and in the meantime, resolve all conflicts. To efficiently managing the trajectory and resolve conflicts, the return function is formulated in Eq. (7) with aircraft performance taken into consideration.

$$\mathcal{R}_{st}^i\left(\{o_{st}^i\}, \{a_{st}^i\}, \{o_{st+1}^i\}\right) = r_1 + r_2 + r_3 + r_4 + r_5 \tag{7}$$

where,

- $r_1 = \Delta t_f = t_{f2} - t_{f1}$ is the increased en-route flight time between initial flight time t_{f1} and revised flight time t_{f2};
- $r_2 = \Delta t_a$ is the increased arrival time;
- $r_3 = \left|\partial \{v_i\}_{i=0}^{N_f-1} / \partial t - dv_{max}\right|$ regulate the speed gradient and avoid sharp acceleration;
- $r_4 = C_{st} - C_{st-1}$ shows the change of the conflict number;
- $r_5 = t_{max}^f - (t_{N_f-1} - t_0)$ requests the en-route flight time should be less than the max flight time that the battery can support.

In the meanwhile, agents cooperates to reduce the loss by sharing the mean return value.

B. MAA3C

With the defined POMDP, the agent is able to take a local observation. But the environment would be non-stationary if all agents perform their actions simultaneously [23]. To deal with the non-stationarity, the centralized critic architecture is a great option to access all agents' observations while updating the policy parameters. Especially, the single-agent asynchronous advantage actor-critic (A3C) [24] takes advantage of the actor-critic architecture and multi-thread learning, in which the actor aims at optimizing the policy and the critic assesses the performance of the policy, to train models efficiently. To utilize this framework for the stationary cooperation of multiple agents, we extend the single-agent A3C to multi-agent A3C (MAA3C).

The advantage function of A3C, which indicates the better action than the average, is described as $A^{\pi}(\mathcal{S}_{st}, a_{st}) = Q^{\pi}(\mathcal{S}, a) - V^{\pi}(\mathcal{S}) = \sum_{j=0}^{k-1} \gamma^j \mathcal{R}^j_{st+k} + \gamma^k V^{\pi}(\mathcal{S}_{st+k}) - V^{\pi}(\mathcal{S}_{st})$. For the multi-agent extension, all agents share the same function to generate their advantage values, which are then averaged to indicate the whole advantage value of the multi-agent system by $A^{\pi}_{avg} = \left\{\sum_{n=0}^{N_a} A^{\pi}_n(\mathcal{S}^n_{st}, a^n_{st})\right\}$. To update parameters of the policy network ϕ and the critic network ϕ_v for the multi-agent system, the policy loss $J(\pi_{\phi'})$ and critic loss $L(\phi_v)$ are replaced with mean values of all agents:

$$J(\pi_{\phi'}) = \mathbb{E}\left[\sum_{st=0}^{T} \left\{log\pi_{\phi'}(a^n_{st} \mid \mathcal{S}^n_{st})\right\}_{avg} \cdot A^{\pi}_{avg}\right] \tag{8}$$

$$L(\phi_v) = \mathbb{E}\left[\frac{1}{2}\sum_{st=0}^{T}\left(\left\{U - V\left(\mathcal{S}^n_{st}; \phi'_v\right)\right\}_{avg}\right)^2\right] \tag{9}$$

Where $\left\{log\pi_{\phi'}(a^n_{st} \mid \mathcal{S}^n_{st})\right\}_{avg}$ presents the average log value of all agents' policies $\pi_{\phi'}$ at current actions a^n_{st}. And $\{U - V\left(\mathcal{S}^n_{st}; \phi'_v\right)\}_{avg}$ equals to average advantage A^{π}_{avg}, U is the accumulative returns and V is the critic value for state $\mathcal{S}^n_{st}$. $\mathbb{E}[\cdot]$ denotes the expected value.

In addition to adjusting the general framework, some localized features introduced for strategic conflict resolution are also presented. We can easily find from Eq. (6) that the traversed blocks number N_f for every aircraft is different, the quantity of concerned agents and therefore the dimension of $\{\mathcal{O}^i\}_{i\in\mathcal{N}}$ amend over steps. To address this challenging problem, the mask recurrent architecture is built. The recurrent neural network has the ability to deal with variable-length input, and in this way, as depicted in Fig. 4, the first layer of the structure is able to compress the information of all traversed blocks. The RNNs in the second layer are responsible for different functions, including generating speed modification parameters $[\delta\mu_k, \delta\sigma_k]$ $(k \in [1, K], K = 2)$, ground delay time δt, the mask and the critic value $V^{\pi}(\mathcal{S})$. We expect that only one choice is made between speed change and ground delay, we propose a two-value mask component, where $mask = 0$ or 1, to accomplish this task. In detail, the mask value is predicted at each step, and the actual action value is concatenated by Eq. (10):

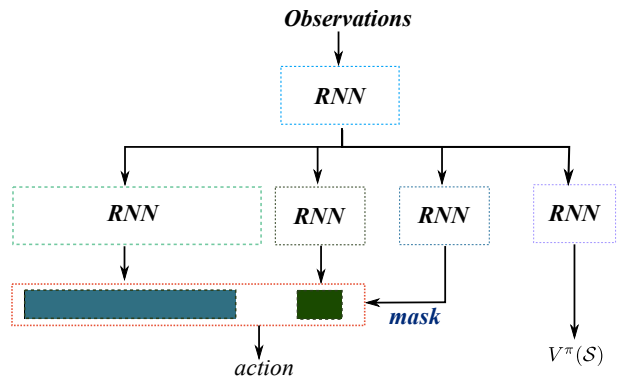


Fig. 4: Mask recurrent neural networks.

Algorithm 2: MAA3C Algorithm

Initialize parameters ϕ and ϕ_v for global thread, ϕ' and ϕ'_v for child threads, $st \longleftarrow 1$ for step counter;
repeat
  $d\phi \longleftarrow 0$ and $d\phi_v \longleftarrow 0$;
  Synchronize parameters from child thread: $\phi' = \phi$ and $\phi'_v = \phi_v$;
  $st_{start} = st$;
  Determine the valid agent list from the up-to-date schedule based on entry count ;
  Get all agents' observations $\{\mathcal{O}^i\}_{i\in\mathcal{N}}$;
  repeat
    Determine the effective action between speed change and ground delay with Eq.(10) for each agent i;
    Execute action $a_{st} = \{a^i_{st}\}_{i\in N}$ according to policy π;
    Get reward $\mathcal{R}_{st}$ and update the schedule for new state s_{st+1};
    $st \longleftarrow st + 1$;
  until *terminal* s_{st} *or* $st - st_{start} == st_{max}$;
  U=$\begin{cases} 0 & \text{for terminal } s_{st} \\ V(s_{st}, \phi'_v) & \text{for non-terminal } s_{st} \end{cases}$;
  for $i \in \{st-1, \cdots, st_{start}\}$ **do**
    $U \longleftarrow \mathcal{R}_i + \gamma U$;
    Accumulate gradients wrt ϕ': $d\phi \longleftarrow d\phi + \nabla_{\phi'}\left\{log\pi\left(a_i \mid s_i; \phi'\right)\right\}_{avg} \cdot \left\{U - V\left(s_i; \phi'_v\right)\right\}_{avg}$;
    Accumulate gradients wrt ϕ'_v: $d\phi_v \longleftarrow d\phi_v + \partial\left(\left\{U - V\left(s_i; \phi'_v\right)\right\}_{avg}\right)^2 / \partial\phi'_v$;
  Perform asynchronous update of ϕ using $d\phi$ and of ϕ_v using $d\phi_v$;
until *MaxSteps*;

$$a_{st}^{i} = \left[mask \times [\delta\mu_k, \delta\sigma_k], \quad (1-mask) \times \delta t\right] \tag{10}$$

We can observe that there should only be one type of action left and, conversely, the other action is set to 0.

The RNNs in the second layer take the agent number as batch size, as a result, no matter how many agents exist at each step, the network can process it flexibly. Combined with the property that the first layer can handle varying numbers of blocks, the scalability of the framework can be promised. Ultimately, the entire process of the extended MAA3C revised from single-agent A3C [24] is described in Algorithm 2.

IV. USE CASES AND ANALYSIS

To evaluate the performance of multi-agent framework in UAM, a small-scale prototype is developed for the last-mile delivery task. And elementary modules including low-altitude airspace, aircraft performance data and operation plan generation are explained exhaustively. The training and test cases are simulated to demonstrate the effectiveness of the model.

A. Simulation Environment

1) Air Corridor: The scenario is simulated based on the layout and specifications of the Multi-User Environment for Autonomous Vehicle Innovation (MUEAVI) road built at Cranfield University. The environment is then copied into the Carla Simulator [25] for airspace or trajectory visualization. As stated in the Section II-A, the low-altitude urban airspace needs to be built at first, and we prefer to construct the dense grid-based air corridor over existing ground routes, as ground infrastructures sensors such as cameras, lidar and radar can provide better assistance for surveillance in urban low-altitude GPS-degraded or GPS-denied environments.

Initially, blocks are stacked along the road to formulate the 5-layer corridor visualized in Fig. 5a, where the dimension of each basic block is $4m \times 4m \times 4m$. The connection of all blocks are modeled as the graph $G(V,E)$, where V is the set of all blocks/nodes and here, $V \in \{0, \cdots, 1900\}$. And 5 vertiports for small UAVs taking-off or landing are indexed and labeled in Fig. 5b.

2) Performance Database: To analyze the efficiency of the multi-agent system for performance-based operation, the aircraft performance data of some small aircraft is firstly collected as the input of simulation experiments, providing that those UAVs are able to accomplish similar task like the last-mile delivery.

In TABLE I, the aircraft size serves as the maximum length in all directions of its body reference framework and is used to merge basic blocks for larger aircraft, as a result, generating adaptive resolution corridors. The maximum hovering time, no matter with full or empty payload demonstrates the ideal maximum flight time that its battery can supply, and is also a constraint for the multi-agent system as displayed in Eq. (7). The speed range limits the updated speed profile generated by GMM and prevents the flight capacity from being overflowed.

3) Operation Plan Generation: With the purpose of training an effective model and resolving the conflicts strategically, an operation plan with 300 trajectories is generated with the established air corridor, aircraft performance data in TABLE I and trajectory generation algorithm in Algorithm 1. In the operation plan, the operation time starts from 06 : 00 to 18 : 00, and the snapshot time is set to 5s, hence the operation period can be divided into 8640 snapshots. Fig. 6 exhibits the spatio-temporal information in the operation plan. Each consecutive curve is an individual trajectory, and we can observe some cross trajectories which indicate the direct conflicts. The block index larger than 1800 is the block for vertiports and is displayed in the area below the horizontal line.

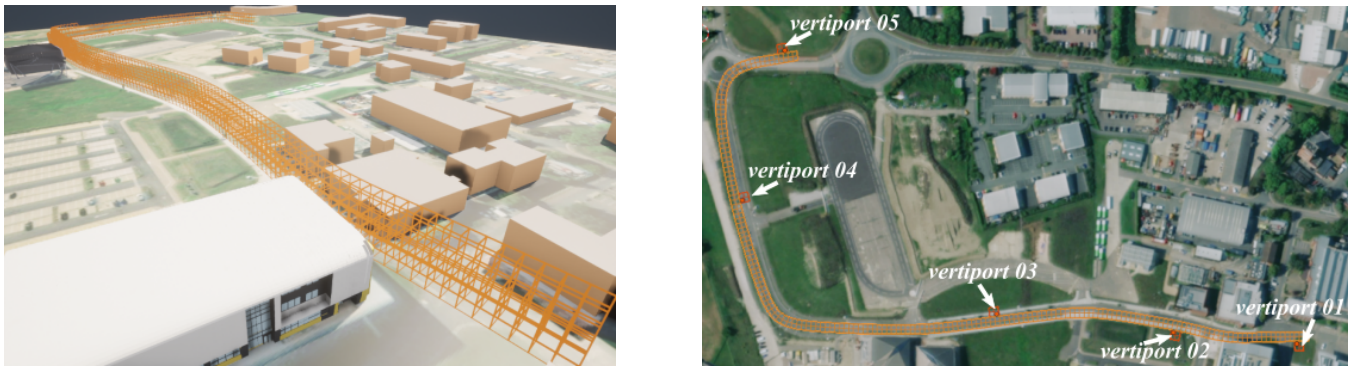
To visualize the traffic flow density, the 2D information is then mapped to the 3D corridor structure as in Fig. 7a. It is evident that the crowded blocks are mainly assembled in the vicinity of vertiports. To further check if all conflicts also cluster around the vertiport, the conflict heat map for the initial operation plan is viewed in Fig. 7b. There are 42 conflicts in total, and we can notice that the conflicts are not only close to the ground vertiports, but also nearby some blocks over the vertiports.

B. Model Training

To resolve the 42 conflicts during the 12-hour operation period, the proposed MAA3C framework is applied. The purpose of the system is to maximize the value of each

TABLE I: Small Aircraft Performance Data.

Aircraft Type	**Size (m)**	**Max Hovering Time (min)** [*Full Payload*]	**Max Hovering Time (min)** [*Empty Payload*]	**Speed Range (m/s)**
DJI Matrice 600	1.668	16	35	[-18,18]
DJI Matrice 100	0.65	13	22	[-22, 22]
DJI Matrice 200	0.887	27	38	[-23, 23]
DJI Mavic pro	0.887	24	24	[-18, 18]
Horsefly Gen 5.3	1.1	25	50	[-22, 22]
DJI AGRAS MG-1S	1.47	10	22	[-15, 15]
DJI AGRAS T30	2.858	7.5	20.5	[-10, 10]
DJI AGRAS T16	2.509	10	18	[-10, 10]



(a) Perspective view

(b) Top view

Fig. 5: Air corridor above MUEAVI road.

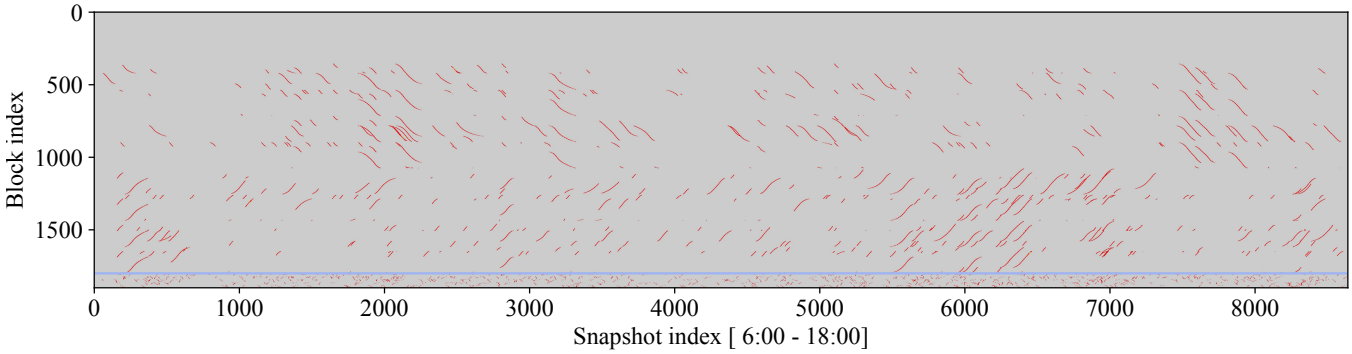
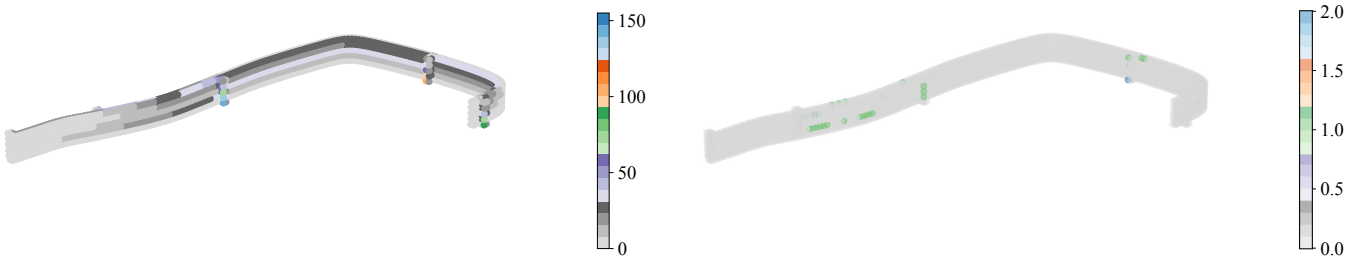


Fig. 6: Spatio-temporal trajectory information associated with traversed sectors.



(a) 3D traffic flow heat map

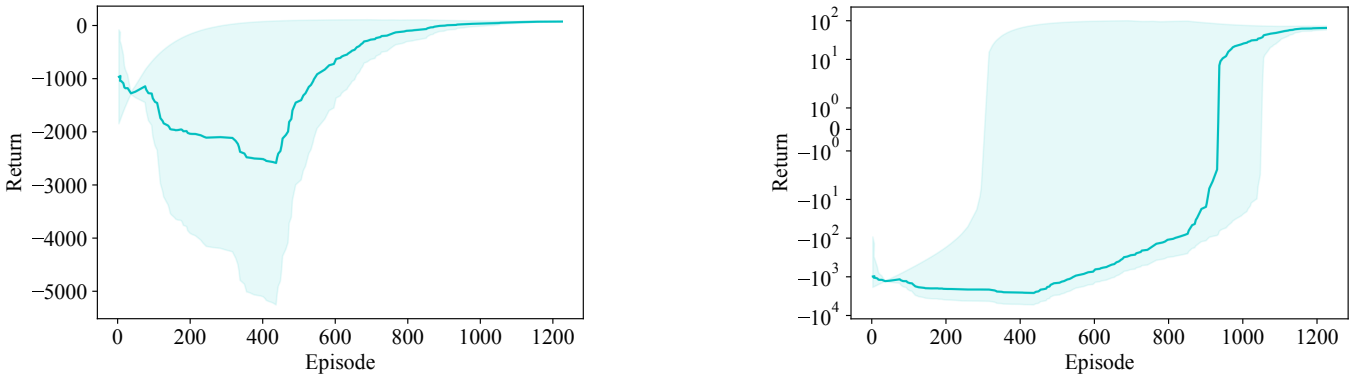
(b) 3D conflict heat map

Fig. 7: Traffic flow and conflict heat maps.

component in the return function and consequently, the total return value. The result is drawn in Fig. 8a after training with multiple trials, the average return value converges after 1000 episodes. And to observe the result in detail, the return values are scaled with log_{10} as drawn in Fig. 8b. In this *case*, the real range of the learning process is much clearer, where the multi-agent system can improve the reward value from -10^{-3} to 10^{2}. In the meanwhile, the conflicts can be finally resolved with the learned parameters as depicted in Fig. 9.

C. Model Tests

It is not sufficient to establish the confidence for the multi-agent solution only based on training results. The solution is supposed to be scalable and adaptable for other different cases. From these considerations, three test scenarios are produced with 200, 300 and 400 trajectories, respectively. And as a comparison, RANDOM, of which elements are random, is employed to replace the network parameters with the continuous uniform distribution, the cumulative distribution function of which is formulated in Eq. (11). And



(a) Training curve.

(b) Training curve (Log scale).

Fig. 8: Return curves of training process.

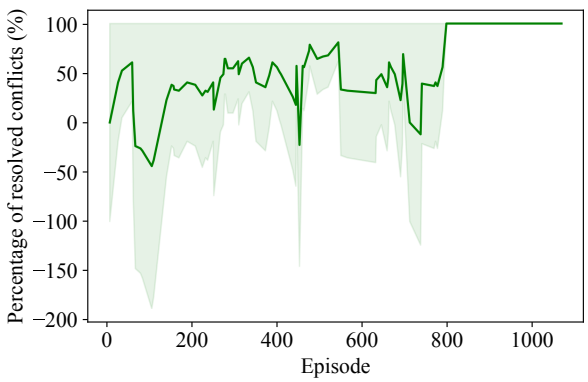


Fig. 9: Conflict resolution curve of training process.

therefore, the random action value can be generated by Eq. (12).

$$f(x) = \begin{cases} \frac{1}{b-a} & a \le x \le b, \\ 0 & x < a \ or \ x > b \end{cases} \quad (a = 0 \ , b = 1) \tag{11}$$

$$[\delta\mu_k, \delta\sigma_k, \delta t] = f(x), \ x \in [0,1] \tag{12}$$

The mask component in RANDOM is exchanged with Bernoulli distribution $p(mask = 0) = p(mask = 1) = 0.5$.

In this instance, there are 6 cases for comparing as each of the 3 scenarios has two choices between MAA3C and RANDOM. Running with the same trained model, the results of all test cases are listed in the TABLE II below.

Case 1 and 2 share the same test scenario with 200 aircraft trajectories. We can know that the conflicts are resolved by both MAA3C and RANDOM but combined with Fig. 10a, MAA3C still has advantages in the return value and average delay.

TABLE II: Results of Test Cases.

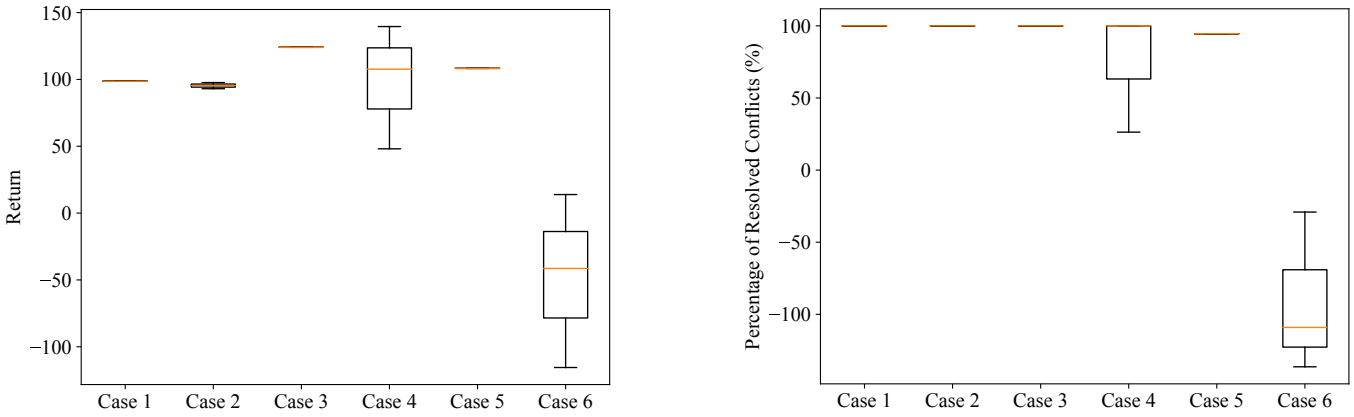
	Method	Flights Num	Conflicts Num	Resolved Ratio	Num for Action	Average Delay (min)
Case 1	MAA3C	200	29	100%	19/200	1.23
Case 2	RANDOM	200	29	100%	20/200	1.57
Case 3	MAA3C	300	57	100%	56/300	5.59
Case 4	RANDOM	300	57	75.4%	58/300	7.32
Case 5	MAA3C	400	55	94.5%	96/400	3.61
Case 6	RANDOM	400	55	0%	149/400	10.94

Cases 3 and 4 face a relatively complex scenario where the conflict number increases to 57. Under such circumstances, the MAA3C reveals its superiority in that 100% conflicts is eliminated in comparison to the 75.4% of RANDOM. At the same time, the variance of multiple trails of Case 4 in Fig. 10b is large. The number of flights that needs to take delay or speed change actions is less than RANDOM, in addition to the less average delay time. The return values in Fig. 10a further indicate the stability of the MAA3C model because of the much larger variance created by RANDOM.

Cases 5 and 6 cope with a more complicated situation where the quantity of trajectories reaches 400 and is larger than the training data. Although the MAA3C cannot eliminate all 55 conflicts, it still maintains a high level of 94.5%, in contrast to the 0% of RANDOM which reversely increases the conflict number. And with an average delay of 3.61 minutes, Case 5 is better than Case 6. More advanced, the return values in Fig. 10a and the percentage of resolved conflicts in Fig. 10b show the stable mean values and small variances in multiple test trials. Unlike Cases 1-4, the divergence between Cases 5 and 6 is much more apparent and demonstrates the robustness of MAA3C.

As opposed to the dramatically decreased performance of Cases 2, 4, and 6 with the increasing number of flights and conflicts, Cases 1, 3, and 5 convince the kernel features of MAA3C for low-altitude UAM operation, including scalability, adaptation, and effectiveness.

Finally, the normalized speed profiles of some agents, which are initially generated by GMM models and revised by MAA3C and RANDOM, are visualized in Fig. 11, 12 and 13. In each plot, the speed value created by parameters $[\mu_1, \sigma_1, \mu_2, \sigma_2]$ changes with the various number of traversed blocks. From Fig. 11, we can notice that the revision results of MAA3C and RANDOM are similar when the study cases are simple. But according to Fig. 12 and Fig. 13, overlapping domains of curves decrease and the differences between MAA3C and RANDOM become obvious in complicated situations. There is no rigorous metric to judge the difference



(a) Return value comparison of test cases.

(b) Ratio value comparison of resolved conflicts.

Fig. 10: Results of test cases.

in GMM revision results alone between MAA3C and RANDOM. But combined with small delay values in TABLE II, we can remark the great fine-tuning ability of MAA3C as it can make the suitable adjustment to avoid potential conflicts in cooperation with the ground delay.

V. CONCLUSIONS

In this paper, the strategic conflict management problem is reconstructed with multi-resolution model of airspace and performance-based operations for UAM, and resolved by the proposed MAA3C method. It is critical to emphasize the importance of performance-based operation, since all parts in strategic conflict management are affected by it, for instance, the efficient low-altitude airspace organization, speed profile generation with GMM, dynamic separation and reward shaping for MARL, etc. With these elaborated designs, the mask recurrent neural networks enables MAA3C to deal with complex situations where hundreds of and various types of small aircraft are planning to accomplish their tasks, and in the meanwhile, provides the feasibility for hybrid operation of scheduled service and on-demand service.

This paper explores to transfer the operative architectures in ATM/UTM to UAM for localization and integrates many frames in current UAM ConOps. But it still stays in the initial stage of UAM development and is a little far from the expected mature UAM operations. More necessary functions (for instance, resilient airspace operations in reaction to unintended disruptions and urban weather prediction, etc.) should be implemented and involved to make the system more robust and applicable.

ACKNOWLEDGMENT

This research was partially supported by grants from the Funds of China Scholarship Council (202008420248).

REFERENCES

[1] P. Kopardekar, J. Rios, T. Prevot, M. Johnson, J. Jung, and J. E. Robinson, "Unmanned aircraft system traffic management (utm) concept of operations," in *16th AIAA Aviation Technology, Integration, and Operations Conference*. AIAA, 2016, pp. 1–16.

[2] "Unmanned aircraft system (uas) traffic management (utm) - concept of operations v2.0," Federal Aviation Administration, Tech. Rep. 1-68, 2020.

[3] "Urban air mobility (uam) - concept of operations v1.0," Federal Aviation Administration, Tech. Rep. 1-37, 2020.

[4] B. P. Hill, D. DeCarme, M. Metcalfe, C. Griffin, S. Wiggins, C. Metts, B. Bastedo, M. D. Patterson, and N. L. Mendonca, "Uam vision concept of operations (conops) uam maturity level (uml) 4," 2020.

[5] O. de l'aviation civile internationale, *Global Air Traffic Management Operational Concept*. ICAO, 2005.

[6] "Drone dcb concept and process," SESAR, Tech. Rep. 1-261, 2021.

[7] J. J. Acevedo, Á. R. Castaño, J. L. Andrade-Pineda, and A. Ollero, "A 4d grid based approach for efficient conflict detection in large-scale multi-uav scenarios," in *2019 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED UAS)*. IEEE, 2019, pp. 18–23.

[8] M. F. B. Mohamed Salleh, C. Wanchao, Z. Wang, S. Huang, D. Y. Tan, T. Huang, and K. H. Low, "Preliminary concept of adaptive urban airspace management for unmanned aircraft operations," in *2018 AIAA Information Systems-AIAA Infotech@ Aerospace*, 2018, p. 2260.

[9] B. Pang, W. Dai, T. Ra, and K. H. Low, "A concept of airspace configuration and operational rules for uas in current airspace," in *2020 AIAA/IEEE 39th Digital Avionics Systems Conference (DASC)*. IEEE, 2020, pp. 1–9.

[10] W. Dai, B. Pang, and K. H. Low, "Conflict-free four-dimensional path planning for urban air mobility considering airspace occupancy," *Aerospace Science and Technology*, p. 107154, 2021.

[11] D. Bertsimas and S. S. Patterson, "The air traffic flow management problem with enroute capacities," *Operations research*, vol. 46, no. 3, pp. 406–422, 1998.

[12] Y. Zhang, R. Su, Q. Li, C. G. Cassandras, and L. Xie, "Distributed flight routing and scheduling for air traffic flow management," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 10, pp. 2681–2692, 2017.

[13] C. Spatharis, T. Kravaris, G. A. Vouros, K. Blekas, G. Chalkiadakis, J. M. C. Garcia, and E. C. Fernandez, "Multiagent reinforcement learning methods to resolve demand capacity balance problems," in *Proceedings of the 10th Hellenic Conference on Artificial Intelligence*, 2018, pp. 1–9.

[14] C. Spatharis, A. Bastas, T. Kravaris, K. Blekas, G. A. Vouros, and J. M. Cordero, "Hierarchical multiagent reinforcement learning schemes for air traffic management," *Neural Computing and Applications*, pp. 1–13, 2021.

[15] C. Huang and Y. Xu, "Integrated frameworks of unsupervised, supervised and reinforcement learning for solving air traffic flow management problem," in *2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC)*. IEEE, 2021, pp. 1–10.

[16] Y. Xie, A. Gardi, and R. Sabatini, "Reinforcement learning-based flow management techniques for urban air mobility and dense low-altitude air traffic operations," in *2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC)*. IEEE, 2021, pp. 1–10.

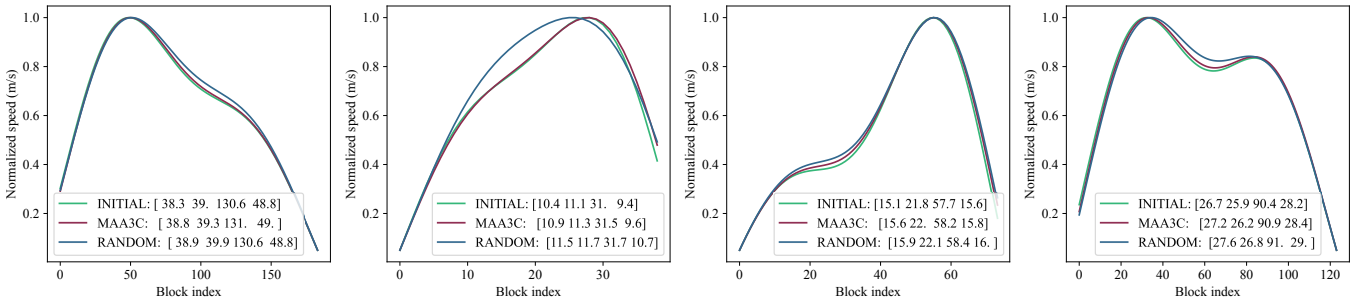


Fig. 11: Speed profiles of selected agents (Case 1 and 2).

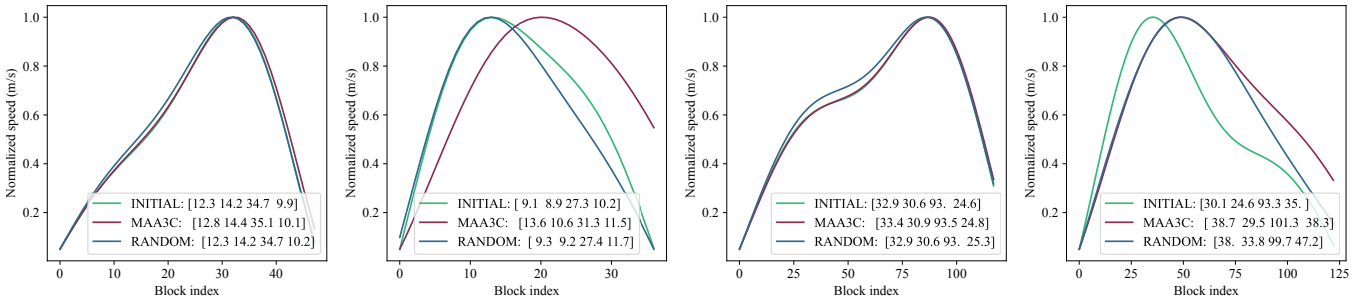


Fig. 12: Speed profiles of selected agents (Case 3 and 4).

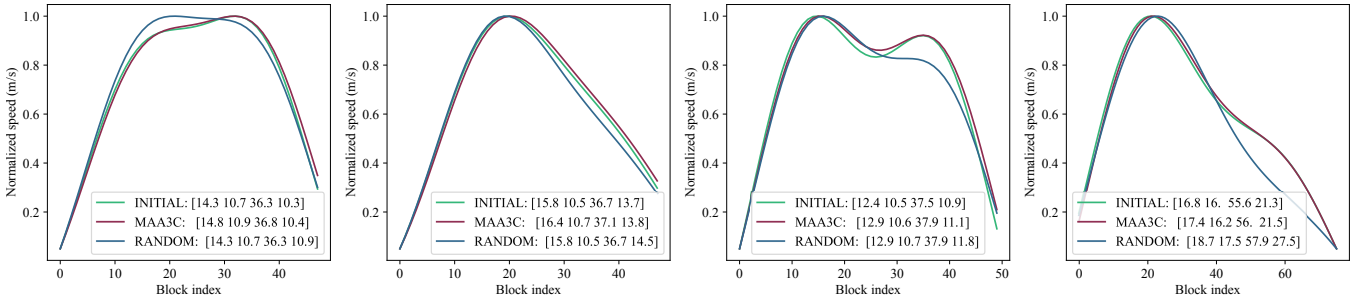


Fig. 13: Speed profiles of selected agents (Case 5 and 6).

[17] S. Balasooriyan, "Multi-aircraft conflict resolution using velocity obstacles," 2017.

[18] M. Pelegrín, R. Delmas, Y. Hamadi *et al.*, "Urban air mobility: From complex tactical conflict resolution to network design and fairness insights," 2021.

[19] D.-T. Pham, N. P. Tran, S. K. Goh, S. Alam, and V. Duong, "Reinforcement learning for two-aircraft conflict resolution in the presence of uncertainty," in *2019 IEEE-RIVF International Conference on Computing and Communication Technologies (RIVF)*. IEEE, 2019, pp. 1–6.

[20] P. N. Tran, D.-T. Pham, S. K. Goh, S. Alam, and V. Duong, "An interactive conflict solver for learning air traffic conflict resolutions," *Journal of Aerospace Information Systems*, vol. 17, no. 6, pp. 271–277, 2020.

[21] W. Wang, Y. Liu, R. Srikant, and L. Ying, "3m-rl: Multi-resolution, multi-agent, mean-field reinforcement learning for autonomous uav routing," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[22] K. Zhang, Z. Yang, and T. Başar, "Multi-agent reinforcement learning: A selective overview of theories and algorithms," *Handbook of Reinforcement Learning and Control*, pp. 321–384, 2021.

[23] G. Papoudakis, F. Christianos, A. Rahman, and S. V. Albrecht, "Dealing with non-stationarity in multi-agent deep reinforcement learning," *arXiv preprint arXiv:1906.04737*, 2019.

[24] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International conference on machine learning*. PMLR, 2016, pp. 1928–1937.

[25] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, pp. 1–16.

2022-07-26

Strategic conflict management for performance-based urban air mobility operations with multi-agent reinforcement learning

Huang, Cheng

IEEE

Huang C, Petrunin I, Tsourdos A. (2022) Strategic conflict management for performance-based urban air mobility operations with multi-agent reinforcement learning. In: 2022 International Conference on Unmanned Aircraft Systems (ICUAS), 21-24 June 2022, Dubrovnik, Croatia

https://doi.org/10.1109/ICUAS54217.2022.9836139