Trajectory Coordination based on Distributed Constraint Optimization Techniques in Unmanned Air Traffic Management

Gauthier Picard

Trajectory Coordination based on Distributed Constraint Optimization Techniques in Unmanned Air Traffic Management

Gauthier Picard
ONERA/DTIS, Université de Toulouse
Toulouse, France
gauthier.picard@onera.fr

ABSTRACT

In this paper, we explore a multiagent-based deconfliction strategy in Unmanned Air Traffic Management (UTM). We equip autonomous unmanned aerial vehicles (UAV) with decision capabilities to update their trajectories (4D contracts) when facing unpredictable events or when priority trajectories are added to the airspace. Current initiatives for UTM envision a multi-layered management system where the lowest one is composed of either autonomous UAV or human-operated one, that may have to react to incidents and emergencies, while sticking to the contracted 4D trajectories. We propose an agent-based approach, where UAVs, being aware of close potential conflicts provided by the UTM communication layer, can coordinate directly, without central UTM control, to update trajectories and solve resulting conflicts in a decentralized manner. This reduces the access to a central decision bottleneck and permits reactive deconfliction. We propose both uncoordinated and DCOP-based coordinated behaviors, that we experimentally evaluate on dense scenarios within a limited area with numerous 4D contracts, potential incidents and emergency procedures.

KEYWORDS

Unmanned Air Traffic Management; Coordination; Distributed Optimization; DCOP; 4D Contracts

ACM Reference Format:
Gauthier Picard. 2022. Trajectory Coordination based on Distributed Constraint Optimization Techniques in Unmanned Air Traffic Management. In *Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), Online, May 9–13, 2022*, IFAAMAS, 9 pages.

1 INTRODUCTION

In 2018, the NextGen office of the US Federal Aviation Administration (FAA) published a first global concept of operations (ConOps) for unmanned aircraft system traffic management (UTM) that presents a vision and describes the associated operational and technical requirements for developing a supporting architecture and operating in a UTM ecosystem [9], depicted in Figure 1. This is a community-based traffic management system in which operators and entities providing operations support services are responsible for the coordination, execution, and management of operations, with rules of conduct established by the FAA. This federated set of services enables cooperative operations management among UAS (unmanned aircraft systems) operators, facilitated by third-party service providers (USSP) through networked information

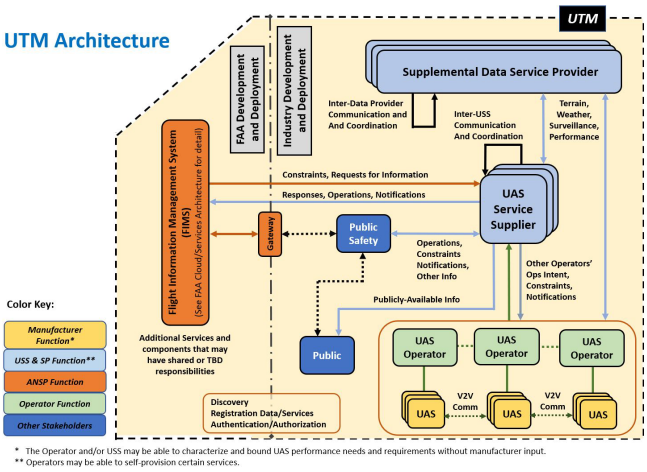

Figure 1: Candidate architecture for Unmanned Traffic Management, as proposed by FAA [9]

exchanges. UTM is designed to support the demand and expectations for a broad spectrum of operations with ever-increasing complexity and risk through an innovative and competitive open market of service providers. The services provided by USSP are interoperable to enable the UTM ecosystem to meet the needs of the UAS operator community. This working document also proposes a set of scenarios raising strong coordination issues, whose resolution proves to be real scientific challenges as illustrated in [12], and which also allow to define scenarios for the evaluation of candidate solutions.

On the European side, the operational model envisioned by the Single European Sky Air Traffic Management Research (SESAR), called U-Space, also identifies the needs for coordination between actors, and the distribution of decisions [25], and supported by the CORUS concept of operations [24]. This service-oriented vision is also promoted by manufacturers and companies as Airbus and Boeing [1], and opens doors for various implementations, as in [5] for instance. Although they display notable differences (classification of UAVs, sectorization of airspace, level of security, responsibilities of the USS) as identified in [15, 18], these ConOps converge on many points (service orientation, modularity, confidentiality, high-level actors, separation into strategic, tactical and reactive levels), particularly concerning the coordination of USSPs and UAS.

In this paper, we propose an adaptive coordination mechanism enabling unmanned aircraft systems (autonomous or operated) evolving in a urban free route airspace to handle trajectory conflicts due to delays, incidents (e.g. the UAV has to stay in position for

capturing some image of a monitored area) or the appearance of new high priority trajectories (e.g. emergency helicopter evacuation). While there exist some adaptation strategies (postponing trajectories or elevating trajectories) to avoid future trajectory conflicts, we propose here to make UAVs coordinate using a direct messaging service (either direct vehicle-to-vehicle or using platform-wide communication) to solve conflicts, and to choose the best actions wrt. multiple objectives. The coordination mechanism is based on distributed constraint optimization, as to ensure UAVs will collectively choose deconfliction actions, while minimizing some operational indicators. The subject of our study being the interaction between UAVs and USSPs, the proposed results aim be generic enough to be applicable to the different UTM and U-Space concepts of operations. Our approach addresses the online 4D contract repair in a distributed fashion. Multi-agent-wise, as far as we know, there is no multi-agent approach to this problem, which is a novel topic following these new concepts of operation in urban air spaces.

This paper is structured as follows. In Section 2, we present an illustrative scenario emphasizing the need for coordination in multi-UAV settings. Section 3 expounds the operational model we use, based on recent concepts of operation [9, 24], and particularly focus on the 4D trajectory model the UAV will commit to. It also presents three uncoordinated deconfliction behaviors (namely, *postpone*, *elevate*, and *skip*), while Section 4 details the coordinated and adaptive approach based on distributed constraint optimization. We experimentally evaluate these behaviors, and multiple DCOP solution methods, on scenarios with randomly generated trajectories, incidents and emergency trajectories. Section 6 finally concludes the paper with some perspectives.

2 ILLUSTRATIVE SCENARIO

Let's use an illustrative scenario to highlight the needs for coordination and adaptation in unmanned Urban Air Traffic settings, depicted in Figure 2. We mainly reuse the scenario proposed in [9], focused on a free route airspace. We consider a 2km by 2km urban area where several UAVs operators have requested some 4D trajectories to perform different tasks, as delivery, surveillance or ground image capturing. These trajectories have been requested to the UTM, which in return sends the 4D way-points (3D space and time) the UAVs have to follow to perform their tasks. UTM also provides some tolerance margins to diverge from the initial trajectory, represented as safety tubes encapsulating the trajectories. The trajectories (and their respective safety tubes) provided by the UTM are ensured to be spatio-temporally conflict-free but require to be requested some hours before the operations. In short, as to flight in this free route area, operators have to inform UTM in advance. The resulting 4D trajectories are thus contracted (and then called *4D contracts*), and UAVs commit to these contracts.

However, in some cases, UAVs may not be able to stick to their trajectory. For instance, unpredictable wind may drastically change the trajectories, especially on small and light UAVs [21]. Moreover, some tasks may require an UAV to stay in position to implement some surveillance routine (e.g. a monitoring UAV detecting a hazard on its surveillance area has to capture the scene on different angles). Finally, some high priority trajectories may be added to the airspace, due to emergency reasons. For instance, following a road accident,

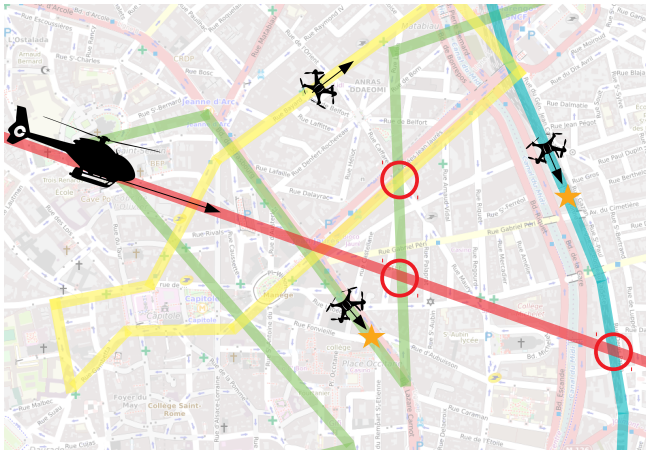

Figure 2: A sample scenario with 3 UAVs following their trajectories (green, blue and yellow) handling some incidents (orange stars), a Medevac helicopter on its emergency trajectory (red), and some identified conflicts (red circles).

a Medevac Helicopter may fly to the accident destination, evacuate the injured people, and then fly back to the hospital, thus potentially generating several conflicts with current trajectories. 4D contracts do not handle such conditional and non deterministic trajectories, and thus must be updated when such events occur.

In our scenario and experiments, we will consider UAVs having some surveillance trajectories, and that may have to implement some incident surveillance routine (with stationary flight), thus suspending their current trajectory. The occurrence of such incidents is not known in advance, and is assumed fully random. We will also consider some emergency events, with destination not known in advance too, that are high priority and cannot be negotiated. In Figure 2, 3 UAVs are following their trajectories (green, blue and yellow) and handling some incidents (orange stars), which will delay their planned trajectories, and thus generate a conflict (red circle at yellow-green crossing) at a position that was spatio-temporally safe before. Even more conflicting, a Medevac helicopter has to cross the airspace in emergency, thus leading so two other conflicts (red circles at green-red and blue-red crossings).

We position the work at the UAS level of Figure 1, where UAVs can directly exchange information via V2V communication, they can receive commands and orders from operators, and obtain some real-time info (e.g. about trajectories, conflicts, emergency procedures) from some USSPs. We envision the proposed tactical and reactive coordination mechanisms to take place between several (semi-)autonomous UAS or between several UAS operators, depending on the autonomy of the vehicles. We notably focus on small UAVs able to perform stationary flight and operating at low altitude (between 0m and 300m).

3 CONCEPTS AND PROBLEM MODEL

This section presents the core models used in this study, namely the 4D trajectories, the UAVs and the UTM services required to implement deconfliction actions between the UAVs.

3.1 4D Trajectories

A trajectory is a set $W \subset \mathbb{R}^4$ of *4D points* $w = (x, y, z, t)$ where x and y are coordinates on the 2D plane (or GPS coordinates), z is the altitude, and t the time. These points define way-points UAVs have to flight over. We call a *segment* $l = (w_s, w_e)$ a line between two timely consecutive points. We will only consider horizontal and vertical segments. UAVs evolves on planes or move to another plane using a vertical segment. Thus the airspace is divided into several planes separated by a constant height, noted Z_{sep}. On each plane, the routes are free, which means there is no predefined flight network. A trajectory is also defined by safety margins, as to prevent UAVs from collision, letting time and space for reactive avoidance routines to subsume navigation orders. Classically, these safety tubes are defined horizontally (for x and y dimensions), vertically (for z) and timely (for t). So a safety tube is defined by $\tau = (h, v, t)$. A UAV is diverging when outside of the safety tube of its current flight segment. A UAV is violating a trajectory if its position is inside the safety tube of another UAV. Formally, A point w_0 is inside a safety tube (h, v, t) for segment $l = (w_1, w_2)$ if it satisfies all three conditions (1) and (2) and (3):

$$\frac{|(x_2 - x_1)(y_1 - y_0) - (x_1 - x_0)(y_2 - y_1)|}{d_{xy}(w_1, w_2)} \leq h \tag{1}$$

$$\min(z_1, z_2) - v \leq z_0 \leq \max(z_1, z_2) + v \tag{2}$$

$$|t_0 - (t_1 + \frac{d_{xy}(w_1, p_{xy}(w_0, l))}{d_{xy}(w_1, w_2)} d_t(w_1, w_2))| \leq t \tag{3}$$

with d_{xy} the 2D Euclidean distance, d_t the 1D time distance, and $p_{xy}(w, l)$ the 2D projection of w on l. Two segments are in *conflict* if they are on the same plane (either horizontally or vertically) and they intersect at the same time (at the tolerance t).

Building 4D trajectory is a classical operational problem with an extensive literature. It has been very well studied in the context of aircraft traffic management [6, 7]. Building conflict-free trajectories is a hard optimization problem, often solved using metaheuristics as simulated annealing [13] or evolutionary algorithms [26]. In the presence of small UAVs able to change direction and speed in a more flexible way than classical aircrafts, the problem remains a hard one, but other techniques as PSO [2] or even multi-agent based ones [29] have been considered. We study here *unstructured*, free route airspace, i.e. there is no predefined air traffic route network the trajectories are constrained to, contrary to usual ATM operational concepts [20]. In our case, we will consider the initial trajectories to be conflict-free, obtained from any trajectory generator (incremental in our experiments).

We focus on the repair procedure; not the generation of the initial set of trajectories. We aim at optimizing some criteria related to the quality of the repair. We consider here: (i) *minimizing the number of conflicts* generated by the trajectory adaptation, as to ensure flight safety; (ii) *minimizing the number of missed way-points*, as to ensure the quality of the trajectories, especially in urban context, where trajectories are defined to fulfill some missions between and at these way-points; (iii) *minimizing the overall delay* induced by the adaptation. Such a problem is non-trivial and may require some trade-off; e.g. skipping a conflicting segments improves safety but reduces quality of service.

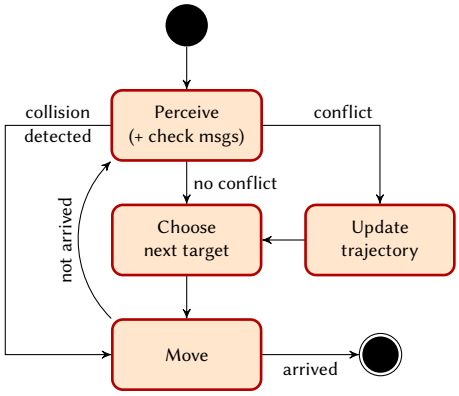

Figure 3: The UAV generic behavior.

3.2 UAVs

Our scenario considers a set U of UAVs, $u = (p, s, d, T)$, where $p = (x, y, z, t) \in \mathbb{R}^4$ is its position, $s = (h, v, a) \leq (h_{max}, v_{max}, a_{max}) \in \mathbb{R}^3$ represents its current horizontal (in $\text{m}.s^{-1}$), vertical (in $\text{m}.s^{-1}$) and angular ($\text{rad}.s^{-1}$) speeds. $d \in [0, 2\Pi]$ defines its current direction, and T is its 4D trajectory/contract. Note that these speed values can be equal to zero (e.g. no motion). In nominal conditions, UAVs change their speed and direction as to reach their next way-point on time, or to enter back into their safety tubes if they have been forced to exit (e.g. due to strong wind or obstacle avoidance). In case, there is an incident close to its position, a UAV will stay at the same position (they keep a way-point at their current position) to observe the situation for a given time. Then it will switch back to its nominal trajectory. We model this target-following behavior as a simple motion model. When u is at position p, aiming to reach position o at angle $\angle(p, o) \in [-\Pi, \Pi]$, it will apply the following variations to its three degrees of freedom:

$$\Delta h = \min(\frac{d_{xy}(p, o)}{d_t(p, o)}, \Delta h_{max}) \tag{4}$$

$$\Delta v = \min(\frac{d_z(p, o)}{d_t(p, o)}, \Delta v_{max}) \tag{5}$$

$$\Delta a = \text{sgn}(\angle(p, o)) \cdot (\min(\angle(p, o), \Delta a_{max}) \tag{6}$$

leading to the following position variation:

$$\Delta p = (h \cos(d), h \sin(d), v, a) \tag{7}$$

UAV nominal behavior depicted in Figure 3 consists first in perceiving the environment and checking incoming messages. Then if no conflict, nor close obstacle is detected, the UAV chooses a target position (either a way-point, an incident location, or its contracted position if it diverged), and then move by adjusting its speeds using (4), (5) and (6) to reach a new position $p + \Delta p$ bringing it closer to its target position. UAVs communicate with UTM services (e.g. geolocalization, conflict detection) and with their neighboring UAVs (either directly or via USSP). Such communication may warn about conflicts. In such case, the UAV will deconflict by possibly updating its trajectory. UAVs are also equipped with reactive collision avoidance routine if they detect a mobile too close to their position. In such case the UAV will perform a "sense and avoid" behavior [27], we do not detail in this paper.

3.3 UTM Services

UAV operators subscribe to services as to obtain information and provide information about their ongoing missions. Real-time information (about positions, conflicts or new inserted trajectories) are send by the services to the subscribed UAVs, while trajectory update requests and incident report are sent by the UAVs to the USSPs. We consider at least the following services:

- A *direct messaging service* (DMS), which enables UAVs to communicate in a P2P manner. UAVs will make use of this service to send coordination messages (related to the DCOP solution method, see Section 4).
- A *conflict detection service* (CDS), which sends to subscribers the conflicts to occur in the next few minutes, following a trajectory update or a new trajectory addition. A conflict information contains the intersecting segments, the respective UAVs, and the intersection point. We note $C(u)$ the set of conflicts in which u is involved. As to keep it reactive enough, the conflict detection is performed at a limited horizon: it only checks the k next segments for each trajectory, starting a the current time.
- A *trajectory update service* (TUS), that UAVs request to postpone their current trajectory or to add new way-points (e.g. to bypass an obstacle outside the safety tubes). In return, the service sends back the new trajectory and the new conflicts (if any). Indeed, if a UAV has to postpone its trajectory, it may produce conflicts on future segments.
- A *trajectory assessment service* (TAS) which is queried for evaluating the replacement of a trajectory by another. This service returns the new trajectory and some evaluation metrics (e.g. the number of generated conflicts and the resulting delay compared to the replaced trajectory).

3.4 Deconfliction Actions and Behaviors

After the conflict-free system starts, non predicted events (as incidents or emergencies) will eventually require the contracts to be broken. Because conflicts mostly consist in intersections on the same plan, and since the considered UAVs are able to perform stationary flight, three main options are opened for updating the contracts: *postpone*, *elevate* and *skip*.

Postpone. The principle of this update is simple: delaying all the way-points of the contract after a given date for a given time, as to be able to avoid a conflict or to have enough time to observe an incident. Postpone is automatically called when a UAV detect an incident close to its position. It will then request to the contract update service a delay for a time depending on the incident. All the way-points after its current spatio-temporal position are delayed by this amount of time. Postpone can also be called by UAVs as to avoid a future potential conflicts identified by the conflict detection service. In such a case, the UAVs will add a new way-point before the intersection and delay its route for an amount of time equals to the time required to extract from a safety tube. We will call postpone UAVs, UAVs only performing postpone deconfliction actions.

Elevate. Another approach to deconfliction with UAVs able of vertical flight if make the UAVs change their plane (either down or up), to bypass the conflict location, and then to flight back to is previous plane. This action is performed as to avoid obstacles or to prevent a future potential conflict. When requesting an elevate update, a UAV provides the conflict or the position to avoid, and the trajectory update service will send back an updated trajectory with a "bridge" over the position to avoid, i.e. the addition of 4 way-points to create a vertical segment, followed by an horizontal one, and finally another vertical one. A bridge is either upward or downward (depending on the altitude). We will call elevate UAVs, UAVs only performing elevate deconfliction actions.

Skip. Finally, one may just skip a conflicting segment, by removing its starting way-point. This the UAV will bypass the conflict by moving directly to the way-point after the conflicting segment. When requesting a skip, the UAV just provides the conflicts to the trajectory update service that will remove the related way-point, and send back the new trajectory. We will call skip UAVs, UAVs only performing skip deconfliction actions.

4 DCOP-BASED COORDINATION

This section defines a cooperative coordination between UAVs, so that they jointly decide which UAVs will trigger a specific deconfliction action for a given conflict. We model this collective decision problem as a DCOP.

4.1 DCOP Framework

Distributed Constraint Optimization Problem (DCOP) is the distributed variant of constrained optimization [19]. Here a group of agents must choose values for a set of variables in a decentralized way in order to minimize a cost function or to maximize a utility function. Formally, A DCOP is a tuple $\langle \mathcal{A}, \mathcal{X}, \mathcal{D}, C, \mu, f \rangle$, where: $\mathcal{A} = \{a_1, \ldots, a_{|A|}\}$ is a set of agents; $\mathcal{X} = \{x_1, \ldots, x_n\}$ are variables owned by the agents; $\mathcal{D} = \{\mathcal{D}_{x_1}, \ldots, \mathcal{D}_{x_n}\}$ is a set of finite domains, such that variable x_i takes values in $\mathcal{D}_{x_i} = \{v_1, \ldots, v_k\}$; $C = \{f_1, \ldots, f_m\}$ is a set of soft constraints, where each f_i defines a cost $\in \mathbb{R}^+ \cup \{+\infty\}$ for each combination of assignments to a subset of variables (a constraint is initially known only to the agents involved); $\alpha : \mathcal{X} \rightarrow \mathcal{A}$ maps variables to their associated agent; $f : \prod \mathcal{D}_{x_i} \rightarrow \mathbb{R}$ is an objective function, representing the global cost of a complete variable assignment. The optimization objective is represented by function f, which, in general, is considered as the sum of costs: $f = \sum_i f_i$. A *solution* to a DCOP P is a complete assignment to all variables. A solution is *optimal* if it minimizes f.

DCOP have been widely studied and applied to many areas [10], because: (i) they focus on decentralized approaches where agents coordinate a joint solution through local message exchanges; (ii) they exploit the domain structure (by encoding it in constraints) to address hard computational problems; (iii) there exists wide variety of solution methods ranging from exact methods to heuristic and approximate techniques. We refer the reader to Fioretto et al.'s survey for a recent catalog of algorithms [10]. DCOP solution methods implement direct message passing, which, in the context our UTM, are send via a DMS.

4.2 Solving Conflicts with DCOP

Let's now model our coordination problem into a DCOP to be instantiated each time some UAVs are aware of some conflict. The idea is twofold: (i) let UAVs choose between several deconfliction

actions when a conflict is detected; and (ii) make UAVs involved in the same conflict coordinate to choose which one is the best for performing the deconfliction.

The set of agents $\mathcal{A} \subseteq U$ is the set of UAVs that have been alerted by the *conflict detection service* for being part of some conflict. Each such UAV u is aware of its conflicts $C(u)$ and the other UAVs it is in conflict with, $U(C(u)) = \{v \in U \mid v \neq u, C(u) \cap C(v) \neq \emptyset\}$. For a given conflict c, a UAVs is able to perform some deconfliction actions; e.g. postpone$(c, 20)$, elevate$(c, -15)$, elevate$(c, +15)$ and skip(c). We note $\mathcal{I}$ this set of actions.

The decisions consist in choosing the deconfliction action to trigger for each known conflict. Let's note $x_{u,c,i} \in \{0, 1\}$ the decision variable stating whether UAV u decides to solve conflict c using action i. Actions are either postpone, elevate or skip. Thus, $\mathcal{X} = \{x_{u,c,i} \mid u \in \mathcal{A}, c \in C(u), i \in \mathcal{I}\}$, $\mathcal{D} = \{d_{x_{u,c,i}} = \{0, 1\} \mid u \in \mathcal{A}, c \in C(u), i \in \mathcal{I}\}$, and $\alpha : x_{u,c,i} \mapsto u$.

Constraints fall into two categories: unary costs (preferences for actions), and coordination ones preventing the same conflict to be solved more than once. Concerning unary costs, we model preferences for actions generating less conflicts without decreasing the quality of service (i.e. increasing the number of missed way-points), and delaying the mission. Agents should be able to valuate the actions using the *trajectory assessment service*. We note $v_{\text{conflict}}(i, c)$ (resp. $v_{\text{missed}}(i, c)$ and $v_{\text{delay}}(i, c)$) the number of conflicts (resp. the number of missed way-points and the delay) of the trajectory when performing action i to solve conflict c. We aggregate these multiple local objective valuation by linearization and prioritization, as follows, for all $u \in U$, $c \in C(u)$ and $i \in \mathcal{I}$:

$$f_{\text{pref}}(x_{u,c,i}) = \omega^2 \cdot v_{\text{conflict}}(i, c) + \omega \cdot v_{\text{missed}}(i, c) + v_{\text{delay}}(i, c) \quad (8)$$

with ω a sufficiently large number to assign preference to conflict resolution against missed way-points and delay reduction. This aggregated valuations model the multiple objectives discussed in Section 3. Concerning coordination constraints, we must ensure exactly one $x_{i,c,u}$ variable is set to 1 for the same conflict. Thus, for each such conflict c:

$$\sum_{u \in U} \sum_{i \in \mathcal{I}} x_{i,c,u} = 1 \quad (9)$$

In order to fit into the DCOP framework, this hard constraint is encoded into a soft constraint as follows:

$$f_{\text{exo}}(x_{u_1,c,i_1}, \ldots, x_{u_{|U|},c,i_{|\mathcal{I}|}}) = \begin{cases} 0, \text{ if (9)} \\ +\infty, \text{ otherwise} \end{cases} \quad (10)$$

Therefore, C is the set of constraints from (8) and (9). The objective function is the sum of all these constraints, thus aiming first at *solving existing conflicts* and then *reducing the overall delay*.

This DCOP can be distributively solved by the involved agents, using any DCOP solution method from the literature, each time agents require to coordinate. In the case an agent detects a conflict with a non-negotiable trajectory (e.g. an emergency), it will still try to solve the same mono-agent problem, thus deciding which alternative between skip, elevate and postpone is the better.

5 EXPERIMENTAL EVALUATION

We now evaluate the performance of the coordinated and adaptive behaviors on synthetic dense scenarios, where multiple UAVs adapt their trajectories following the occurrence of incidents and emergency trajectories.

Figure 4: A sample dense scenario with 20 UAVs flying over a 2km x 2km urban area. UAVs are represented as circles (with their current speed vector). 4D contracts (and their respective safety tubes) are represented in green. Current segments are represented in blue.

5.1 Experimental Setup

We will analyze the performances of the investigated deconfliction behaviors with a growing number of UAVs (and trajectories). They are coded in Java and executed on 20-core Intel(R) Xeon(R) CPU E5-2660 v3 @ 2.60GHz, 62GB RAM, Ubuntu 18.04.5 LTS, with an OpenJDK 11.0.9.1 JVM. We ran 30 instances of randomly generated sets of trajectories and incidents for each fleet size, and plot the average values, with [0.05, 0.95] confidence interval. The DCOP algorithms used here are the implementations from FRODO Library [14]. Randomly generated values are uniformly chosen within provided intervals. The reported computation time is the mono-CPU 20-core simulation time.

We consider an area of 2km by 2km, illustrated in Figure 4, with vertical airspace planes at 20m, 40m and 60m. We consider UAVs with the following characteristics: $h_{max} = 18m.s^{-1}$, $v_{max} = 6m.s^{-1}$, $a_{max} = \Pi/2\text{rad}.s^{-1}$, $\Delta h_{\max} = \Delta v_{\max} = 6m.s^{-2}$, $\Delta a_{\max} = \Pi/2\text{rad}.s^{-2}$. Initial speed is set to $(0, 0, 0)$. Initial UAV trajectories are randomly and incrementally generated as follows. The starting point is randomly positioned in the airspace, at altitude $z = 0$ and at time $t = 0$. The next points are randomly chosen within a distance between 200m and 1000m, so that there is no conflict with existing segments, by possibly increasing or decreasing the altitude. The time for the next point is also set using a default cruise speed, equals to $(12, 3, 0)$ (equiv. to 40km/h horizontally). UAVs ave

initially 60 way-points to their destination, then a last segment back to their origin, leading to a dense set of crossing trajectories. Safety tubes are defined by $(h, v, t) = (30, 15, 1)$, which means that UAVs must be separated by at least 30m horizontally, 15m vertically, with 1s temporal tolerance. To add unpredictable events, we generate 3 emergency trajectories, consisting in starting from a random position at the border of the area, moving to a random point at at least 1km distance from the initial point, then a waiting point for 360s (to simulate an emergency response), and finally a way back to the initial point. These emergency trajectories are not editable. Each simulated second there is also a 1% chance an incident occurs close to a randomly chosen UAV, thus having to capture the scene for a random duration between 30s and 120s.

We evaluate the following behaviors: (i) postpone characterizes UAVs that only perform a postpone$(c, 20)$ action when a conflict is identified (subscribing to CDS and TUS services); (ii) elevate characterizes UAVs that only perform elevate$(c, \pm 15)$ actions, depending on their current plane, when a conflict is identified (subscribing to CDS and TUS services); (iii) skip represents UAVs that only perform skip(c) action when a conflict is detected (subscribing to CDS and TUS services); (iv) afb [11], dpop [22], dsa (variant C) [28], mgm2 [16] represent UAVs which perform a coordinated and adaptive decision using a DCOP algorithm[1] to deconflict, that opt between postpone(c, d) with $d \in \{20, 40, 60\}$, elevate$(c, \pm 15)$ and skip(c) (subscribing to DMS, CDS, TAS and TUS services); and finally, (v) centralized is a tree-search based deconfliction algorithm computing the optimal sequence of repair actions (from the same set than DCOPs) to fully repair a conflicting set of trajectories, by optimizing objective derived from equation (8). All UAVs are equipped with collision avoidance routine and equipment. CDS Service is configured with an horizon $k = 10$, which in our case represents approx. 10min in the future.

Figure 5 presents performance metrics obtained by simulation, with an increasing number of UAVs (and trajectories); and Figure 6 shows the evolution of some metrics, with time, for a specific instance with 25 UAVs, 3 emergency procedures and 25 incidents.

5.2 Conflicts and Contracts

The first row in Figure 5 (5a-5d) focuses on conflicts and trajectories. While in classical ATM, with structured airspace, postponing at departure is used to reduce conflicts [3], in our free route scenario, postpone is generating numerous conflicts due to the accumulated delays over crossing trajectories, which are not structured as flow networks, as shown in Figure 5a. All the other behaviors still generate conflicts (around 200 on larger instances), since almost 70 trajectories are added to the airspace on larger instances.

Facing conflicts, UAVs update their contracts, which are no more guaranteed to be conflict-free compared to the initial ones. Thus, UAVs tend to request numerous contract updates, as illustrated in Figure 5b. This is especially true for postpone and elevate, which delays trajectories for some duration, which most of the time is not sufficient for breaking the conflicts, or which generates other conflicts. On larger instances, a trajectory update is requested every second, on average. These dynamics of trajectory updates make some UAVs violate some others' safety tubes, and often trigger the anti-collision routines, as illustrated in Figures 5c and 5d. Indeed, when a UAV requests for a contract update with a new segment very close to its current position (in space and time) and to another UAV's position, this may lead to such situations, before any of the 2 UAVs is able to update again its contract. Notably, elevate makes UAVs violate 4 times more safety tubes than the other behaviors. These numbers of contract updates and violations is high because we choose here to generate very dense trajectories, which represents some sort of worst scenario, where even safety margins are not enough to prevent collisions.

Looking to our single instance (Figure 6), we can see how much dynamic this setting is, letting no rest for the UAVs to repair the trajectories and fulfilling their missions, which last for approx. 1h. There is only a "calm" window between time $t = 291$ and $t = 744$. We can observe how the addition of incidents immediately triggers postpone action (that all agents implement to handle incidents). In particular, postpone and elevate generate numerous conflicts just after the addition of a first emergency trajectory, as shown in Figure 6a. Facing such conflicts, UAVs constantly attempt to repair the trajectories, until there is no more incidents and emergency in the airspace, as illustrated by Figure 6b, after $t = 2290$. Even when contracts are stable, UAVs may still continue violating some safety tubes, as shown in Figure 6c, since they have been computed using some time projection, that may not be very precise.

5.3 Trajectory Conservation and Delays

The second row of Figure 5 presents metrics related to the second and third optimization criteria we look at, namely trajectory conservation and delays. In Figure 5e, which reports the average percentage of conserved initial way-points in the final trajectories, skip is the less conservative behavior (up to 12.5% of way-points are skipped), since it automatically bypasses way-points when facing a conflict, which also results in lower time to terminate all the trajectories (reported in Figure 5f). Both elevate and postpone add delay for solving conflicts (either by changing altitude, thus spending time in elevation, or by postponing next way-points), as reinforced in Figures 5g and 5h showing the total accumulated delay and the average delay per UAV. Note that every behavior adds delay: this is due to the fact that UAVs handle numerous incidents (around 35 in larger instances) forcing them to stay in position for a while, thus accumulating delay. Centralized solver loses performances on larger instances since it first consider minimizing conflicts and missed way-points, and thus postpone more frequently than DCOP-based solvers. The DCOP-based approaches locally co-optimize conflicts, trajectory conservation, and delays, which makes them an efficient compromise.

Looking at our isolated instance, we can observe how skip constantly skips way-points to answer to conflicts, as shown in Figures 6d and 6e. skip deleted 3 times more points than DCOP-based behaviors. The adaptations to incidents continuously add delays to trajectories, due to the postpone action triggered in such cases.

5.4 DCOP Operation

The third row of Figure 5 reports metrics about the DCOP operations. While the generated trajectories are dense within a limited

[1]We do not include MaxSum algorithm [8], since it cannot converge in most of our instances, thus providing low quality results while requiring numerous messages.

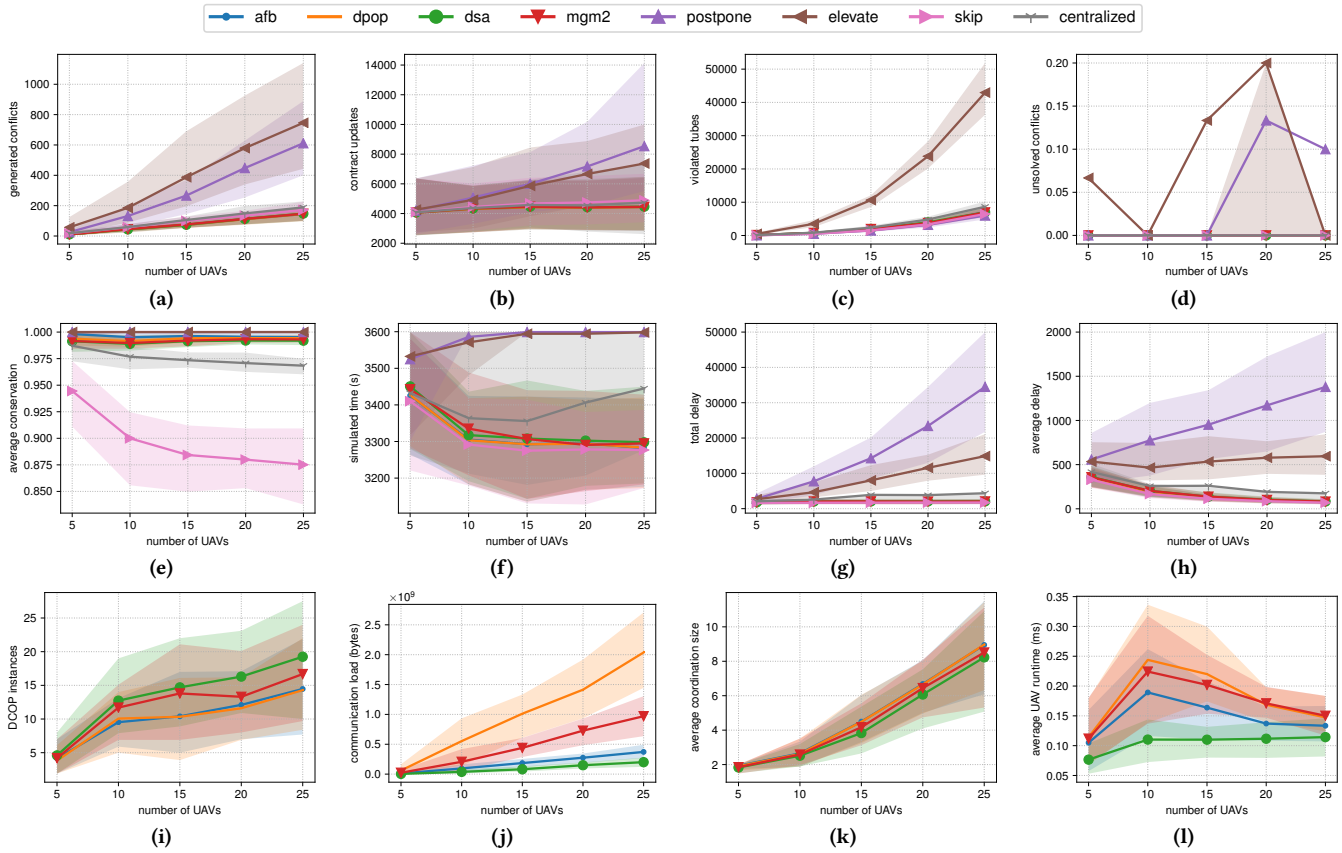

Figure 5: Average values over 30 instances for several performance metrics with increasing number of UAVs.

area, the DCOP procedures between several UAVs is constantly but not so frequently triggered, as seen in Figure 5i. This is explained by the fact that DCOP-based coordinated decisions manage to solve conflicts for the given horizon, and then do not require other coordination for some time, expect when other disturbances occur. Interestingly, incomplete solvers (dsa, mgm2) require more coordination than complete solvers (afb and dpop). In fact, since the decisions made by these incomplete solvers are non optimal, they may result on conflicting decisions requiring further deconfliction coordination. During coordination, conflicts are concerning one third of the total fleet on average, as shown in Figure 5k. Coordination can theoretically regroup many agents, but in our setting, restricted DCOP sizes allow the use of complete solution methods as afb and dpop, which does not strongly impact the individual UAV decision runtime as reported in Figure 5l. Looking at the communication load in Figure 5j, dpop requires exchanging few but large messages since the DCOPs to solve are cyclic and thus require exchanging exponential size messages. mgm2, due to its 2-coordination settings, exchanges numerous small messages.

DCOP-based behaviors are triggered when facing some of the disturbances, as shown in Figure 6f, and apply the most valued actions, namely elevate (to conserve way-points) and skip (to limit delays), as shown in Figures 6h and 6i. The postpone action is not used in such conditions, as observed in the "calm" period in Figure 6g. This explains how DCOPs achieve good results as trade-offs between contract conservation and delays. afb is the best competitor, with good deconfliction requiring low communication load. However, afb is not robust to message loss, contrary dsa. Further experiments should be done to assess the impact of communication unreliability on DCOP performances.

5.5 Summary

These results highlight how the setting is a very extreme dynamic configuration, where UAVs have to constantly update that trajectories. Clearly, the postpone behavior is bad on two dimensions: delays and conflicts (therefore safety). The elevate behavior is a good candidate for trajectory conservation, but with extra delays and numerous safety violations, while skip generates few conflicts, at the expense of some missed way-points. DCOP-based behavior is positioned as a good compromise between elevate and skip, thus meeting our objective of minimizing generated conflicts, maximizing the trajectory conservation and minimizing the delays, as good as a centralized deconfliction. In other settings, with many crossroads on structured airspace, these results may not hold. In fact, many of the performances depends on the fact that UAVs can freely change altitude or bypass some way-points. In more structured airspace, UAVs would normally change altitude in some limited

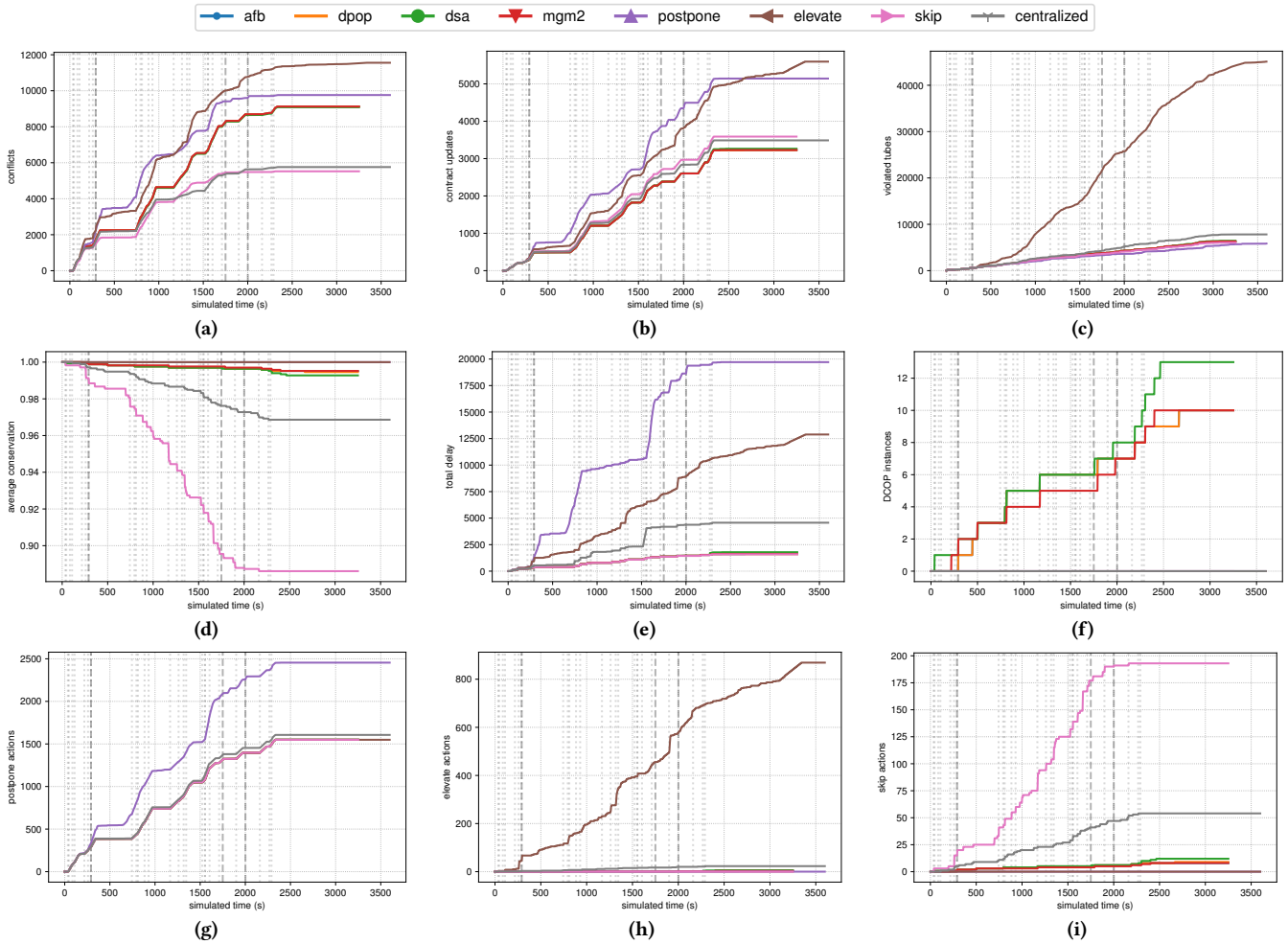

Figure 6: Evolution of performance metrics during one simulation with 25 UAVs and 3 emergency procedures (gray dashed lines) and 25 incidents (gray dotted lines).

vertical corridors, and cannot fly outside horizontal corridors, as defined in [5] for instance. In such cases, postpone would probably behaves better, while DCOPs would form larger groups of conflicting trajectories, flowing through some limited number of crossroads. We keep this investigation and analysis for future works.

6 CONCLUSION

In this paper we promoted and implemented a multi-agent coordination mechanisms to help UAVs evolving in a dynamic free route airspace to adapt their trajectories. Coordination is built upon a distributed constraint solving protocol, which aims at minimizing the future conflicts, maximizing the conservation of the initial trajectories, and reducing the overall delay implied by deconfliction actions. On the simulated scenario we developed, forming a dense airspace with random events to handle, we show that using a coordinated and adaptive deconfliction, provides solutions with reduced conflicts, missed way-points and accumulated delay, compared to uncoordinated and fixed ones. Still, this coordination requires some extra in all, the DCOP-based approach is a promising candidate for installing autonomous deconfliction in UAVs collective, and paves the way for future research.

Since UTM and U-Space visions are evolving, especially towards more structured airspace [4, 5], we envision to extend our framework and services to such settings, which promise to be harder to solve due to the lack of trajectory freedom to deconflict and the larger size of deconfliction groups. We particularly aim to adapt the DCOP solution method to use to the trajectory deconfliction settings. In fact, we used here a straightforward linear aggregation of three criteria, which still generates non-conflict-free trajectories. Multiobjective DCOPs could be a relevant approach for future developments [17]. Moreover, devising less communication intensive coordination algorithms appears to be a requirement to deploy such protocol and services on real devices and UTM. This could be achieved by either limiting the size of the deconfliction DCOPs or by relying on lightweight and robust solution methods as A-DSA, as proposed in the Internet-of-Things context [23], at the expense of completeness.

REFERENCES

[1] Airbus and Boeing. 2020. *A New Digital Era of Aviation: The Path Forward for Airspace and Traffic Management.* Technical Report. Airbus.

[2] D. Alejo, J. A. Cobano, G. Heredia, and A. Ollero. 2013. Particle Swarm Optimization for collision-free 4D trajectory planning in Unmanned Aerial Vehicles. In *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*. 298–307. https://doi.org/10.1109/ICUAS.2013.6564702

[3] Nicolas Barnier and Cyril Allignol. 2012. Trajectory deconfliction with constraint programming. *The Knowledge Engineering Review* 27, 3 (2012), 291–307. https://doi.org/10.1017/S0269888912000227

[4] Aleksandar Bauranov and Jasenka Rakas. 2021. Designing airspace for urban air mobility: A review of concepts and approaches. *Progress in Aerospace Sciences* 125 (2021), 100726. https://doi.org/10.1016/j.paerosci.2021.100726

[5] Carlos Capitán, Héctor Pérez-León, Jesús Capitán, Ángel Castaño, and Aníbal Ollero. 2021. Unmanned Aerial Traffic Management System Architecture for U-Space In-Flight Services. *Applied Sciences* 11, 9 (2021). https://doi.org/10.3390/app11093995

[6] Daniel Delahaye and Stéphane Puechmorel. 2013. *Front Matter.* John Wiley & Sons, Ltd. https://doi.org/10.1002/9781118743805

[7] D. Delahaye, S. Puechmorel, P. Tsiotras, and E. Feron. 2014. Mathematical Models for Aircraft Trajectory Design: A Survey. In *Air Traffic Management and Systems*. Springer Japan, Tokyo, 205–247.

[8] A. Farinelli, A. Rogers, A. Petcu, and N. R. Jennings. 2008. Decentralised Coordination of Low-Power Embedded Devices Using the Max-Sum Algorithm. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems*. pp. 639–646.

[9] Federal Aviation Agency. 2020. *UTM concept of operations version 2.0.* Technical Report. Federal Aviation Agency. https://www.faa.gov/uas/research_development/traffic_management/media/UTM_ConOps_v2.pdf

[10] F. Fioretto, E. Pontelli, and W. Yeoh. 2018. Distributed Constraint Optimization Problems and Applications: A Survey. *Journal of Artificial Intelligence Research* 61 (2018), 623–698.

[11] Amir Gershman, Amnon Meisels, and Roie Zivan. 2006. Asynchronous Forward-Bounding for Distributed Constraints Optimization. In *Proceedings of the 2006 Conference on ECAI 2006: 17th European Conference on Artificial Intelligence August 29 – September 1, 2006, Riva Del Garda, Italy*. IOS Press, NLD, 103–107.

[12] Youssef Hamadi. 2020. Optimization for Urban Air Mobility. In *Learning and Intelligent Optimization*, Ilias S. Kotsireas and Panos M. Pardalos (Eds.). Springer International Publishing, Cham, 1–8.

[13] Arianit Islami, Supatcha Chaimatanan, and Daniel Delahaye. 2017. *Large-Scale 4D Trajectory Planning.* Springer Japan, Tokyo, 27–47. https://doi.org/10.1007/978-4-431-56423-2_2

[14] Thomas Léauté, Brammert Ottens, and Radoslaw Szymanek. 2009. FRODO 2.0: An Open-Source Framework for Distributed Constraint Optimization. In *Proceedings of the IJCAI'09 Distributed Constraint Reasoning Workshop (DCR'09)*. Pasadena, California, USA, 160–164. https://frodo-ai.tech.

[15] J. Lieb and A. Volkert. 2020. Unmanned Aircraft Systems Traffic Management: A comparsion on the FAA UTM and the European CORUS ConOps based on U-space. In *2020 AIAA/IEEE 39th Digital Avionics Systems Conference (DASC)*. 1–6. https://doi.org/10.1109/DASC50938.2020.9256745

[16] R. T. Maheswaran, J. P. Pearce, and M. Tambe. 2004. Distributed Algorithms for DCOP: A Graphical-Game-Based Approach.. In *ISCA PDCS*. 432–439.

[17] Toshihiro Matsui, Marius Silaghi, Katsutoshi Hirayama, Makoto Yokoo, and Hiroshi Matsuo. 2012. Distributed Search Method with Bounded Cost Vectors on Multiple Objective DCOPs. In *PRIMA 2012: Principles and Practice of Multi-Agent Systems*, Iyad Rahwan, Wayne Wobcke, Sandip Sen, and Toshiharu Sugawara (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 137–152.

[18] Tim McCarthy, Lars Pforte, and Rebekah Burke. 2020. Fundamental Elements of an Urban UTM. *Aerospace* 7, 7 (2020). https://doi.org/10.3390/aerospace7070085

[19] P.J. Modi, W. Shen, M. Tambe, and M. Yokoo. 2005. ADOPT: Asynchronous distributed constraint optimization with quality guarantees. *Artificial Intelligence Journal* (2005).

[20] Cesar Nava–Gaxiola, Cristina Barrado, and Pablo Royo. 2018. Study of a Full Implementation of Free Route in the European Airspace*. In *2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*. 1–6. https://doi.org/10.1109/DASC.2018.8569543

[21] Gabriele Perozzi, Denis Efimov, Jean-Marc Biannic, and Laurent Planckaert. 2018. Trajectory tracking for a quadrotor under wind perturbations: sliding mode control with state-dependent gains. *Journal of the Franklin Institute* 355, 12 (2018), 4809–4838. https://doi.org/10.1016/j.jfranklin.2018.04.042

[22] A. Petcu and B. Faltings. 2005. A scalable method for multiagent constraint optimization. In *International Joint Conference on Artificial Intelligence (IJCAI'05)*. 266–271.

[23] Pierre Rust, Gauthier Picard, and Fano Ramparany. 2020. Resilient Distributed Constraint Optimization in Physical Multi-Agent Systems. In *European Conference on Artificial Intelligence (ECAI)*. IOS Press. http://ecai2020.eu/papers/108_paper.pdf

[24] SESAR. 2019. *U-Space: Concept of Operations.* Technical Report. SESAR Joint Undertaking.

[25] SESAR joint Undertaking. 2020. *European ATM Master Plan: Roadmap for the safe integration of drones into all classes of airspace.* Technical Report. SESAR. https://www.sesarju.eu/sites/default/files/documents/reports/MP2020FR.pdf

[26] Su YAN and Kaiquan CAI. 2017. A multi-objective multi-memetic algorithm for network-wide conflict-free 4D flight trajectories planning. *Chinese Journal of Aeronautics* 30, 3 (2017), 1161–1173. https://doi.org/10.1016/j.cja.2017.03.008

[27] Andrew D. Zeitlin. 2010. Sense amp; Avoid capability development challenges. *IEEE Aerospace and Electronic Systems Magazine* 25, 10 (2010), 27–32. https://doi.org/10.1109/MAES.2010.5631723

[28] W. Zhang, G. Wang, Z. Xing, and L. Wittenburg. 2005. Distributed Stochastic Search and Distributed Breakout: Properties, Comparison and Applications to Constraint Optimization Problems in Sensor Networks. *Artificial Intelligence* 161, 1–2 (2005), 55–87.

[29] Wenjie Zhao, Zhou Fang, and Zuqiang Yang. 2020. Four-Dimensional Trajectory Generation for UAVs Based on Multi-Agent Q Learning. *Journal of Navigation* 73, 4 (2020), 874–891. https://doi.org/10.1017/S0373463320000016