

Decentralized Multi-Agent Path Finding for UAV Traffic Management

Florence Ho¹, Rúben Geraldes², Artur Gonçalves³, Bastien Rigault, Benjamin Sportich, Daisuke Kubo, Marc Cavazza⁴, and Helmut Prendinger⁵, *Member, IEEE*

Abstract—The development of a real-world Unmanned Aircraft System (UAS) Traffic Management (UTM) system to ensure the safe integration of Unmanned Aerial Vehicles (UAVs) in low altitude airspace, has recently generated novel research challenges. A key problem is the development of Pre-Flight Conflict Detection and Resolution (CDR) methods that provide collision-free flight paths to all UAVs before their takeoff. Such problem can be represented as a Multi-Agent Path Finding (MAPF) problem. Currently, most MAPF methods assume that the UTM system is a centralized entity in charge of CDR. However, recent discussions on UTM suggest that such centralized control might not be practical or desirable. Therefore, we explore Pre-Flight CDR methods where independent UAS Service Providers (UASSPs) with their own interests, communicate with each other to resolve conflicts among their UAV operations—without centralized UTM directives. We propose a novel MAPF model that supports the decentralized resolution of conflicts, whereby different ‘agents’, here UASSPs, manage their UAV operations. We present two approaches: (1) a prioritization approach and (2) a simple yet practical pairwise negotiation approach where UASSPs agents determine an agreement to solve conflicts between their UAV operations. We evaluate the performance of our proposed approaches with simulation scenarios based on a consultancy study of predicted UAV traffic for delivery services in Sendai, Japan, 2030. We demonstrate that our negotiation approach improves the “fairness” between UASSPs, i.e. the distribution of costs between UASSPs in terms of total delays and rejected operations due to replanning is more balanced when compared to the prioritization approach.

Index Terms—Unmanned aircraft system traffic management (UTM), pre-flight conflict detection and resolution (CDR), multi-agent path finding (MAPF).

I. INTRODUCTION

WITH the growing demand in commercial services provided by Unmanned Aerial Vehicles (UAVs), the

Florence Ho, Rúben Geraldes, Artur Gonçalves, Bastien Rigault, and Helmut Prendinger are with the Digital Content and Media Sciences Research Division, National Institute of Informatics, Tokyo 101-8430, Japan (e-mail: florence@nii.ac.jp).

Benjamin Sportich is with the Department of Computer Science, Sorbonne University, 75006 Paris, France.

Daisuke Kubo is with the Unmanned Aircraft Systems Technology Team, Japan Aerospace Exploration Agency, Tokyo 305-8505, Japan.

Marc Cavazza is with the School of Computing and Mathematical Sciences, University of Greenwich, London SE10 9JR, U.K.

conception of an Unmanned Aircraft System Traffic Management (UTM) system has become necessary [17], [18], as any ‘conflict’ [19], i.e. possibility of collision between UAVs, must be avoided. Hence, the definition of UTM regulations has generated growing interest in the development of Conflict Detection and Resolution (CDR) methods to ensure separation between UAVs. Similar to Air Traffic Management (ATM), UTM will employ three redundancy layers that are applied in different phases of a UAV’s flight [15]: Pre-Flight CDR, In-Flight CDR and Collision Avoidance. This redundancy principle is based on the guidelines of the International Civil Aviation Organization (ICAO) [13].

On the one hand, the development of Collision Avoidance and In-Flight CDR methods have recently received significant attention [1], [7], [9], [14], [35], [37]. On the other hand, only few works have presented practical methods to proactively deconflict UAV traffic before UAVs take off, i.e. Pre-Flight CDR methods [10], [11].

In this article, we focus on the conception of Pre-Flight CDR methods that aim at solving conflicts between UAV operations before their takeoff. We assume four dimensional (3D plus time) trajectory (4DT) executions, whereby UAVs must follow waypoints at given timesteps.

The Pre-Flight CDR process, also called “strategic deconfliction” [28] will first provide 4DT flight plans determined before takeoff. Then, in case of disturbances, i.e. dynamic events that would disturb the plans generated by Pre-Flight CDR, In-Flight CDR will be triggered in real-time while UAVs are flying. Pre-Flight CDR is necessary to reduce the amount of conflicts that In-Flight CDR would have to process otherwise. We have previously evaluated the impact of such situation [10]. If Pre-Flight CDR was not applied, the complexity of deconfliction for In-Flight CDR would increase, and possibly jeopardize the safety objective. Thus, efficiency and safety are increased through the application of Pre-Flight CDR combined with In-Flight CDR, whereby the latter addresses disturbances.

It has been shown that the Pre-Flight CDR problem can be formulated as a Multi-Agent Path Finding (MAPF) problem [11]. However, the standard MAPF formulation does not capture real world considerations such as those of UTM. First, most existing works in MAPF assume a centralized setting, where each agent executes a unique path determined by a central computing entity. Second, they assume that a global cost is minimized over all agents, whereby all agents are cooperative such that no agent is prioritized over another, and

any agent may be instructed to take a longer path to avoid conflicts.

UTM assumes that different UAS Service Providers (UASSPs), each with their own group of UAVs, will provide services, such as de-confliction services, to UAS operators who submit UAV operation requests from customers [28]. UASSPs are independent entities with their own business objectives to maintain a certain service quality or satisfy operational constraints. For instance, if the flight path of a UAV operation significantly deviates from its nominal path to avoid conflicts, or if a UAV operation request is rejected, it could negatively affect customers’ expectations and require some compensation.

UASSPs aim to provide conflict-free paths while minimizing their own costs in terms of the amount of delays and rejections on their given UAV operations. In the UTM context, a decentralized Pre-Flight CDR process is motivated by practical requirements defined by regulating authorities such as NASA [28], whereby UASSPs communicate among themselves to resolve conflicts between their respective UAV operations. Moreover, UAV operations will have associated valuations to reflect the different levels of importance attributed by customers.

In this context, decentralized approaches for Pre-Flight CDR are necessary and beneficial because they allow UASSPs to deconflict their operations according to costs as determined by themselves. At the same time, UASSPs can protect private information that defines their views in terms of costs for replanning each operation.

In this article, we present a novel MAPF model for the Pre-Flight CDR problem with the characteristics shown in Table I. In particular, we support a notion of “fairness” that relates to the distribution of costs in terms of delays and rejected operations between UASSPs. On the other hand, the standard MAPF formulation makes no distinction between UASSPs in the resolution process. Therefore, the resolution of conflicts may result in largely different impacts on each UASSP, as there is no provision to ensure a balanced distribution of delays and rejected operations among UASSPs. By contrast, our proposed approach provides “fairness” among UASSPs, which is a core consideration in UTM.

To the best of our knowledge, we are the first to propose a decentralized MAPF process for Pre-Flight CDR, where UASSP agents with their individual costs deconflict their flight paths before takeoff. This article makes the following main contributions:

- We formulate an extended MAPF model, where UASSPs plan collision-free paths for their given UAV operations and aim to minimize their own cost objective. We define these costs based on independent practical considerations such as induced delays and rejections.
- We introduce a decentralized MAPF algorithm that supports resolution with negotiation between UASSPs representing UAV operations.

We compare our approach, which uses a bilateral negotiated resolution, to an approach using prioritization instead of negotiation.

TABLE I
MAPF CONCEPTS FOR PRE-FLIGHT CDR

UTM Standard MAPF problem concept	UTM Novel MAPF problem concept
Centralized	Decentralized
Agent = UAV (flight path)	Agent = UASSP entity managing several UAV operations (flight paths)
UAV operations are all of same importance	Each UAV operation has a valuation (low, medium, high)
Cooperative agents	Self-interested agents
Minimize “sum of costs” (durations of flight paths) over all UAV agents	Minimize individual costs (delays, etc) for each UASSP agent
→ Global optimization	→ Pareto optimization

With prioritization, UASSPs determine an order between themselves, according to which they will replan their whole set of UAV operations. Here, the lower placed UASSP will replan all its operations while considering all of the higher placed UASSP’s operations as fixed spatio-temporal obstacles. We will consider two strategies to determine the order between UASSPs: a randomized strategy and a cost-based strategy. We evaluate our approaches on scenarios based on a realistic study of predicted demand of UAV delivery operations for 2030 in Sendai, Japan.

The rest of the paper is structured as follows. Section II presents related works on MAPF and CDR approaches. Section III formalizes the Pre-Flight CDR problem where each UASSP agent with given UAV operations aims to minimize its costs in terms of delays and rejected operations. Sections IV and V describe our different proposed approaches, i.e. a prioritization process and a negotiation process. Section VI presents the performed simulations and the experimental evaluations of our approaches.

II. RELATED WORKS

A. Multi-Agent Path Finding

The Multi-Agent Path Finding (MAPF) problem has been extensively studied in many works proposing approaches to solve instances mostly for 2D benchmarks [6], [21]. In the MAPF setting, agents located in a graph must follow a path to their goal locations without colliding with each other. The existing methods generally assume a centralized setting where a single computing entity finds a solution considering all given agents, with a global objective to minimize, such as the “sum of costs” or the “makespan”. To minimize these global objectives, optimal and suboptimal approaches have been proposed [3], [29], [30], [33], [34].

Recently, new directions were established to handle real-world scenarios, notably for ground robot path finding in Amazon Robotics warehouses [21]. Several variants of the MAPF formulation have been proposed [12], [20], [22] that consider teams of agents and focus on allocating tasks to each agent while providing conflict-free paths. In [11], an extension of the MAPF problem to the Pre-Flight CDR problem in the UTM context was presented, but it used a centralized setting.

In this article, we assume that all UAV operations are submitted with already defined start and service locations. So,

in our work, we do not address the task allocation process. Instead, we consider distinct teams of UAV agents, whereby each team is represented by an UASSP agent that aims to minimize its own cost function.

Few works have considered a decentralized setting in MAPF [4], [8], [26], where agents communicate between themselves to find a conflict-free solution. However, these works consider the standard global optimization setup where agents, each with a unique path, communicate to resolve conflicts to minimize the total solution costs without expressing any self-interest.

In [2], the authors introduce a mapping to combinatorial auctions, and in [5], a taxation mechanism is proposed that incentivises self-interested agents to optimize social welfare.

In all these existing approaches, a mediating entity, such as an auctioneer, is needed to process all preferences of agents. By contrast, a decentralized process is required for the UTM architecture [28], in which all UASSP agents communicate between themselves to resolve conflicts of their operations.

B. Automated Negotiation

Several works in the Air Traffic Management (ATM) domain have proposed decentralized CDR approaches that use automated negotiation, mostly for the In-Flight CDR phase [27], [32], [36]. In [36], the authors applied a monotonic concession protocol where each aircraft can concede to the other aircraft whether it will fly a higher cost trajectory to resolve a conflict. However, these negotiation protocols assume a common cost function known by all agents or require each agent to explicitly communicate their preferences to others.

In [27], a decentralized negotiation based In-Flight CDR approach is proposed where a pair of aircrafts can negotiate their maneuvers to solve a conflict. Their short-term deconfliction approach only processes a local immediate conflict in each negotiation process, and computes simple response trajectories to the immediate conflict, thus not mitigating downstream conflicts. So the induced deviation from the initial flight path might not be minimal for a given flight path and thus it provides a sub-optimal solution in terms of total deviation.

These works on In-Flight CDR support concessions to create a balance in the costs of resolutions between aircrafts.

In contrast, we resolve conflicts in a strategic way by processing flight paths, i.e. operations, of UAVs before their actual takeoff. Thus, unlike In-Flight CDR, the Pre-Flight CDR process does not have strict real-time requirements since all flights are processed within a certain amount of time ahead of takeoff.

Other works such as [24] focus on the design of automated negotiation approaches by modeling a utility function for each user with improved preference elicitation processes. Reference [23] introduce a self-interested approach for MAPF where each self-interested agent is assumed to negotiate one individual path. However, this approach might not be scalable or practical when there is a large number of conflicts between paths, as a large amount of communications would be required to reach a conflict-free solution for all paths.

By contrast, our work aims at solving a novel MAPF problem that considers UASSP agents with given sets of flight

paths. We focus on the definition of a practical Pre-Flight CDR approach based on negotiation, rather than based on the process of users' preference elicitation.

III. PROBLEM FORMULATION

In this section, we formalize our model that extends the standard MAPF formulation with UASSP agents that are in charge of deconflicting their given set of UAV operations. Each UASSP agent aims to minimize its individual cost function, which takes into account the delays induced for each UAV operation with a given valuation, and the number of operations rejected.

A. Motivation

In the context of UTM, using a decentralized process is a practical requirement defined by regulating authorities [28].

Existing centralized methods in the standard MAPF framework consider a global objective, and thus seek globally optimal solutions without distinguishing agents' individual costs (see Table I).

By contrast, in the context of low altitude airspace traffic management, each UASSP agent's individual interest or priority is an important realistic consideration. Thus, a decentralized method allows us to address self-interested agents and maintain privacy about their preferences in terms of costs for replanning, unlike a centralized method.

Moreover, applying a decentralized resolution at the UASSP level instead of the UAV level has two main advantages:

1. It addresses practical requirements: If we resolve conflicts on the UAV level, UAVs would communicate between themselves to solve their conflicts in a decentralized way in the Pre-Flight phase. However, in the UTM system architecture defined by NASA UTM [28], it is required that only UASSP agents communicate between themselves to deconflict their operations in the Pre-Flight phase. Further, UASSPs are economic entities that have their own preferences in terms of deconfliction costs. They are responsible for conflict resolution of their operations.
2. It reduces communication costs: If UASSPs communicate to deconflict their operations, yet the processing would be for each UAV operation on a "per conflict" basis, this would require a significantly high amount of communications between UASSPs to resolve the large numbers of conflicts. Thus, in contrast, UASSP level resolution allows us to avoid this communication overhead by processing several conflicts at once.

B. Definitions

We distinguish between a *UAV agent* (a UAV operation) that performs a flight path, and a *UASSP agent* that manages a group of UAV agents (UAV operations), as shown in Table I. In this formulation of the Pre-Flight CDR problem, a UASSP agent communicates with other UASSP agents to determine the resolution for their respective group of UAV agents.

We define an agent as a UASSP that is responsible for a set of UAV agents (UAV operations). We consider K given

UASSPs agents. Each UASSP agent $(A_k)_{k \in [1;K]}$ has to resolve its own MAPF problem instance defined by a set of N_k UAV operations $\Omega_k = (a_i)_{i \in [1;N_k]}$ that are each performed by a UAV. Each operation represents a flight path with a fixed start location and a goal location. We consider an undirected graph G which is a 26-neighbor 3D cubic grid map. UAVs can ‘move’ along an edge of G or can ‘wait’ on a vertex of G .

Each operation a_i has an attributed valuation v_i that reflects its importance, which can be “low”, “medium”, or “high”, depending on customer requirement. Thus several operations may have the same valuation.

Each UAV assigned to perform an operation a_i is represented by a sphere of given radius r_i , and a center position p_i . For any pairs of UAV operations a_i and a_j , any violation of the minimum separation distance, i.e. the sum of the respective radii of the associated UAVs $r_i + r_j$, must be prevented. Hence, we define the following constraint to ensure there is no conflict at any timestep t :

$$\forall t, \text{dist}(p_i(t), p_j(t)) > r_i + r_j \quad (1)$$

In the UTM context, we also define altitude constraints for the flight paths of UAVs by reference to coordinates (x, y, z) :

$$\text{elevation}(x, y) + \text{alt}_{\min} \leq z \leq \text{elevation}(x, y) + \text{alt}_{\max} \quad (2)$$

$\text{elevation}(x, y)$ refers to the terrain elevation value of the point of coordinates (x, y) in a path, measured from mean sea level, and $\text{alt}_{\min} = 90$ m and $\text{alt}_{\max} = 150$ m are fixed altitude bounds relative to elevation. Hence, there is a 60 m altitude range for UAV operations. A solution of the given MAPF instance for A_k consists of conflict-free paths for all UAV operations in Ω_k .

C. UASSP Cost Function

Unlike the standard MAPF formulation, each UASSP agent A_k has its own independent objective which is to minimize its own cost value C_k^v . That value represents the service degradation due to conflict resolution on its given operations of same valuation, as shown in Eq. 3. An operation is rejected when no resolution is found to solve a conflict or when the generated deviation exceeds the given battery autonomy. The operation is then removed from the solution set, i.e. rejected. A rejection is considered as a more important penalty than an added delay, so the whole rejected path duration is considered as the incurred cost.

We define the cost function C_k^v for each UASSP agent A_k for all of its operations of same valuation $v \in \{\text{low}; \text{medium}; \text{high}\}$ as:

$$C_k^v(\Omega_k) = \sum_{a_i \in \Omega_k^v} c(a_i) \quad (3)$$

$$c(a_i) = \begin{cases} T'_i - T_i & \text{if path is replanned (delay)} \\ T_i & \text{if path is rejected} \\ 0 & \text{else} \end{cases} \quad (4)$$

where $\Omega_k^v \subseteq \Omega_k$ is the subset of operations of A_k with same valuation v , $c(a_i)$ represents the individual cost for each operation a_i with initial flight duration T_i and flight duration obtained after replanning T'_i .

D. Fairness

Fairness is a core consideration in the UTM context, as several economic and competitive entities are involved to provide services to customers in the shared low altitude airspace. We distinguish two notions of “fairness” according to the type of agent, for a *UAV agent* and for a *UASSP agent*, and we address both notions:

- *UAV agent (operation) level*: the order of valuations must be respected among all operations. Here, we address this issue by prioritizing higher valued UAV operations when in conflict, i.e., they become spatio-temporal obstacles for the lower valued operations which in turn must be replanned.
- *UASSP agent level*: we conceive fairness as the more balanced distribution of costs, such as total delays and rejected operations, among all UASSP agents for all operations of same valuation.

Fairness at the UAV operation level simply prioritizes according to the given valuations of the operations. On the other hand, fairness at the UASSP agent level requires the conception of a CDR approach that allows UASSP agents to determine a mutual solution that they would consider as “fair”.

E. Optimality

We distinguish two notions of optimality whereby each reflects a different objective:

1. *Optimality in terms of total deviation caused by resolving conflicts for all given UAV agents of an UASSP agent*: This relates to the difference in the total durations between the conflict-free flights and the initial flights of the given UAV agents (delays). The optimality hereby refers to the solution that minimizes the deviation caused for all UAV agents’ flight paths, i.e., the solution that minimizes the “sum of costs” objective, as in standard MAPF.
2. *Optimality in terms of fairness for all UASSP agents*: This concerns the *distribution* of costs related to deviations among different UASSP agents. The optimality hereby refers to the solution that maximizes fairness, i.e., the distribution of deviations costs among UASSP agents.

Here, we focus on fairness among different UASSP agents, which is a novel objective for optimality.

Note that in the UTM context, it is important to make a trade-off between theoretical optimality of a solution and practical constraints, such as scalability.

IV. PREPROCESSING PHASE

This section describes the initial steps that each UASSP agent performs independently before processing Pre-Flight CDR with other UASSPs. For this purpose, we address specific properties of the configuration of UAV delivery service scenarios. Each UASSP agent is responsible for a given set of hubs. Each UAV assigned to perform an operation departs from a given hub and returns to the same hub once its mission is completed. We assume that each hub can service a

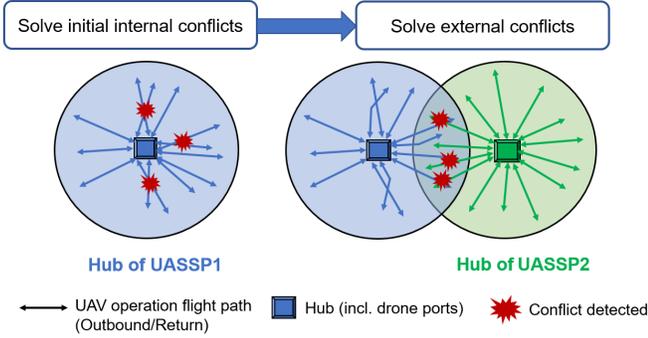


Fig. 1. Flowchart of the Pre-Flight CDR process for a UASSP agent: (1) Each UASSP agent first solves all existing conflicts among its own set of operations, using a standard MAPF solver such as ECBS [3]; (2) Each UASSP agent communicates with other UASSPs to deconflict operations between them.

predetermined limited surrounding area (‘service area’) with a fixed radius as represented in Fig. 1, whereby specific service locations are assigned to each UAV. Therefore, the hubs of a UASSP agent may have overlapping service areas with other hubs pertaining to other UASSPs, where conflicts can occur between UAV operations of different UASSPs.

First, each UASSP processes its own operations so that its own paths are initially conflict-free, as shown in Fig. 1. Then, each UASSP agent A_k only needs to communicate the subset of operations $\Omega_k^{shared} \subseteq \Omega_k$ that may be in potential conflicts with the other UASSPs based on their overlapping service areas as depicted in Fig. 1. For this purpose, each UASSP performs an initial step where operations that are crossing an overlapping area are filtered with a simple geometrical verification of each flight path segments, based on the radius of each service area.

In other words, we draw a distinction between conflicting paths belonging to a same UASSP (‘internal conflicts’) and those between different UASSPs (‘external conflicts’). Without this step, new ‘internal conflicts’ might be detected and created after solving ‘external conflicts’, which might be impractical, as this would increase the amount of communications. Hence, we always start with conflict-free paths from each UASSP side.

V. DECENTRALIZED APPROACHES

We assume that each UASSP agent uses a given MAPF solver to solve their ‘internal conflicts’, such as the bounded suboptimal ECBS algorithm [3] (here, we refer to the optimality in terms of total deviation as mentioned in SECT. III-E). So, each UASSP agent processes its own MAPF instance in a bounded suboptimal way with the use of the ECBS algorithm. This bounded suboptimality guarantee is relative to the consideration of other UASSPs agents’ operations as spatio-temporal obstacles. Then, each UASSP agent starts the de-confliction with other UASSP agents, based on its conflict-free set of operations.

In this section, we introduce two distinct decentralized algorithms to solve the Pre-Flight CDR problem between UASSPs agents.

- A *prioritization* approach where UASSPs determine an order of replanning between themselves. Here each UASSP will replan their respective operations considering the trajectories of operations of higher placed UASSPs as fixed spatio-temporal obstacles.
- A *pairwise negotiation* approach where UASSPs make concessions on the operations that they both will replan. This allows UASSPs to share the costs between them by determining which operations should be replanned and which not.

Let us first consider the prioritization approach as a baseline approach. Then we will describe the pairwise negotiation resolution approach, and then generalize to the resolution with more than two agents.

A. Prioritization Approach

Similar to the Cooperative A* approach [31] used to solve MAPF instances for a centralized setting, we hereby propose a decentralized resolution with prioritization between UASSPs. In Cooperative A*, each agent is initially associated to a unique and distinct priority, and when two agents are in conflict, the agent with lower priority must replan its path considering the agent with higher priority as obstacle.

Differently, since we consider UASSP agents with given UAV operations, the determined priority order for each UASSP agent applies to all of its operations, thus several paths at once. However, randomly attributing priorities to UASSP agents might be unfair, as it would distribute costs among UASSP agents without a rationale, and thus may lead to arbitrarily high individual costs for UASSP agents.

Therefore, we propose an informed ‘‘cost-based’’ strategy that relies on the computed costs of each UASSP agent, so as to determine the priority ordering among UASSP agents (see Algorithm 1). Here, UASSP agents will mutually determine their pairwise order by communicating their incurred costs with Eq. 3.

Each UASSP agent computes its incurred cost as if it was to resolve all conflicts alone, i.e. by determining conflict-free paths while considering all the paths of the other given UASSP as spatio-temporal obstacles.

Then, the UASSP which would have incurred a higher cost is placed higher, as ‘‘winner’’, and does not have to replan its paths, while the UASSP with lower cost, as ‘‘loser’’, has to replan all its paths in conflict with those of the ‘‘winner’’. The rationale here is that the ‘‘loser’’ has to re-plan because it results in less total costs.

Note that each UASSP agent carries out the same method of conflict detection computation to determine which conflicts need to be solved with the other UASSP agent. This ensures that if a conflict is detected by one UASSP agent, it will also be detected by the other UASSP agent.

Regarding the computation of conflict resolution, one UASSP agent is not repeating the computation that has been performed by the other one, in the sense that each UASSP agent will compute and propose a different conflict-free solution to the other UASSP agent. Such creation of a conflict-free solution is done by the UASSP agent’s re-planning of its own

Algorithm 1 Pairwise Prioritization

Data: $(A_i; A_j)$ two UASSP agents with respective associated set of operations Ω_i^{shared} and Ω_j^{shared}
Result: Conflict-free paths for all UASSPs
/* Pseudocode for agent A_i */
/* Cost evaluation for each agent operations */
 $\Omega_i^{replanned} \leftarrow \text{SolveMAPF}(\Omega_i^{shared}, \Omega_j^{shared});$
Send $C_i^v(\Omega_i^{replanned})$ to A_j ;
Receive $C_j^v(\Omega_j^{replanned})$ from A_j ;
/* Determine if "winner" or "loser" */
if $C_i^v(\Omega_i^{replanned}) \leq C_j^v(\Omega_j^{replanned})$ **then**
 /* A_i is the "loser" and A_j 's paths become obstacles for A_i */
 $\Omega_i^{shared} \leftarrow \Omega_i^{replanned};$
/* Else A_i is the "winner" and does not replan its paths */

UAV operations, while considering the other UASSP agent's UAV operations as spatio-temporal obstacles.

B. Negotiation Approach

The prioritization approach presents a simple decentralized MAPF solver that incorporates individual costs of each UASSP agent and provides a resolution in a systematic manner. However, this method might be considered "unfair", as some UASSP agents may have to sustain a large amount of costs by replanning all their operations, only because they are ranked lower than other UASSP agents.

Therefore, we propose a simple yet practical resolution method based on negotiation by incremental concession making. This allows UASSPs to bargain and effectively share their replanning costs by mutually determining which of their operations to replan or not to replan. Thus, there is no "loser" who has to bear all costs or "winner" who does not replan at all.

Operators submit operations some time ahead of their desired start time, e.g., at least 30 minutes before their UAV desired takeoff time. Typically, the chosen time window is large enough to handle a significant amount of operations from different UASSPs. Note that the actual negotiation time is a few minutes.

1) *Bargaining Problem:* This approach can be assimilated as solving a bargaining problem, which is a game theory concept that aims to find an equilibrium when conflicts of interest arise between players with separate and conflicting objectives [25].

We propose a mapping of the Pre-Flight CDR problem to the bargaining problem as presented in Table II. Here, conflict resolution is framed as a two-player game where each UASSP agent would prefer the other agent to bear the costs, i.e. resolve the given conflicts by replanning their own operations.

2) *Theoretical Properties:* Solving the bargaining problem means finding an agreement $s \in F$ for both UASSP agents

TABLE II
REDUCTION OF THE PRE-FLIGHT CDR PROBLEM TO THE BARGAINING PROBLEM

Bargaining	Pre-Flight CDR
Feasibility set $F \subset \mathbb{R}^2$	All attainable costs for each UASSP
Disagreement point $d \in F$	Cost for each UASSP resolving all conflicts alone
Agreement point $s \in F$	Conflict-free solution with better costs than d for each UASSP

A_i and A_j according to their respective cost function C_i^v and C_j^v . The agreement point s hereby represents the costs computed by each UASSP agent for a conflict-free solution where certain operations are replanned between both UASSP agents, and s is on the Pareto frontier [25]. In particular, if the Pareto frontier of the costs to all agents is known, then the agreement is the Kalai-Smorodinsky solution [16]. The properties of this solution are based on the axioms of game theory: Pareto optimality, symmetry, invariance and monotonicity [16]. However, here the Pareto frontier is not known, as the cost of each proposal (conflict-free solution) can only be valued by the UASSP agent that will replan its operations. Thus, the process converges to a solution that is as close as possible to the optimal "fair" solution with its costs on the Pareto frontier.

3) *Our Approach:* A pair of agents communicate with each other to negotiate which paths to replan within a fixed maximum number of rounds N_{Max} (see Algorithm 2).

For this purpose, at each round, they follow three main steps as follows:

- 1) Each UASSP agent A_i makes a *proposal* $\Omega_i^{proposal}$ to the other agent. A *proposal* contains all operations of the agent whereby the agent proposes to solve certain conflicts by replanning certain of its operations, while assuming the other agent will solve the remaining conflicts.
- 2) Then, each UASSP agent A_i evaluates the proposal received from the other agent by computing a *response* $\Omega_i^{response}$. A *response* is a conflict-free solution computed by the agent which replans its operations considering the paths proposed by the other agent as spatio-temporal obstacles.
- 3) Each agent determines the costs from their proposal made $C_i^v(\Omega_i^{proposal})$ and their response to the proposal of the other agent $C_i^v(\Omega_i^{response})$. An *agreement* is reached if the cost of the response is lower than the cost of the proposal: $C_i^v(\Omega_i^{response}) \leq C_i^v(\Omega_i^{proposal})$.

Otherwise, the bargaining process iterates again for another round through these steps while incrementing the costs for the respective new proposals until an agreement is reached. An agreement represents conflict-free solutions that effectively distribute costs between the two UASSPs.

The key step in this approach is the determination of a proposal by each agent in each round. A proposal is a concession on which operations to replan whose associated cost increases at each round to ensure convergence to an agreement.

Algorithm 2 Pairwise Negotiation

Data: $(A_i; A_j)$ two UASSP agents with respective associated set of operations Ω_i^{shared} and Ω_j^{shared} ; N_{Max}

Result: Conflict-free paths for all UASSPs

```
/* Pseudocode for agent  $A_i$  */
/* Initial proposal and response */
 $\Omega_i^{proposal} \leftarrow \Omega_i^{shared}$ ;
Send  $\Omega_i^{proposal}$  to  $A_j$ ;
Receive  $\Omega_j^{proposal}$  from  $A_j$ ;
 $\Omega_i^{response} \leftarrow SolveMAPF(\Omega_i^{shared}, \Omega_j^{proposal})$ ;
/* Determine the groups of operations
 $g \in G_i$  involved in external
conflicts, and sort by increasing
value of individual costs  $C_i^v(g)$  */
 $G_i \leftarrow GenerateGroupsOperations(\Omega_i^{response})$ ;
 $\delta_i \leftarrow \lceil |G_i|/N_{Max} \rceil$ ;
while true do
  /* Generate new updated proposal with
  incremented cost */
   $g_{\delta_i} \leftarrow getFirstGroups(G_i, \delta_i)$ ;  $\triangleright$  Select  $\delta$  groups
  with lowest costs to re-plan
   $\Omega_i^{proposal} \leftarrow Update(g_{\delta_i}, \Omega_i^{proposal})$ ;
  Send  $\Omega_i^{proposal}$  to  $A_j$ ;
  Receive  $\Omega_j^{proposal}$  from  $A_j$ ;
  /* Compute response to proposal of
  other agent */
   $\Omega_i^{response} \leftarrow SolveMAPF(\Omega_i^{shared}, \Omega_j^{proposal})$ ;
  /* Evaluate response and compare to
  proposal */
  if  $C_i^v(\Omega_i^{response}) \leq C_i^v(\Omega_i^{proposal})$  then
     $\perp$  break;  $\triangleright$  Agreement reached
  else
     $\perp$   $\delta_i \leftarrow \delta_i + \lceil |G_i|/N_{Max} \rceil$ ;  $\triangleright$  Update  $\delta_i$ 
```

To generate a proposal, an agent must identify which of its operations are in conflict with the other agent and determine which operations it will replan to resolve some of the identified conflicts. The proposal is updated at each round by additionally replanning some operations according to the criterion of their incurred costs.

For this purpose, both agents must communicate an initial proposal which is a “zero-cost” solution where they do not replan any of their paths, i.e., the replanning cost is zero, and assume that the other agent will resolve all of the existing conflicts. Here, for each agent the computed response is the same computed solution as in Section V-A, where each agent computes the conflict-free paths by considering all the paths of the other agent as obstacles.

This first step allows both agents to identify the existing conflicts between them, so that each agent A_i can determine the set G_i of groups of operations involved in a conflict. A group of operations $g \in G_i$ is a subset of operations of the same agent which are impacted by the same conflict with an operation of the other agent.

There may be several distinct conflicts between agents, thus several groups of operations. Then, a set contains the groups of operations to be proposed for replanning at each round of negotiation. Each group $g \in G_i$ is associated to a cost value $C_i(g)$ computed with Eq. 3, which represents the individual incurred cost to replan the operations of the group. G_i is sorted by ascending order of costs for each group so that the groups with lower costs are selected first.

The agents then determine the value δ_i , which is the number of groups of operations from the sorted set G_i that they will need to re-plan to update their current proposal at each round, so that the negotiation process converges in less than N_{Max} rounds. In the case that both agents agree on their respective proposals, the proposal with the lowest variance in costs between the two agents is then chosen.

4) *Convergence Analysis:* This approach will always terminate in less than N_{Max} rounds since each agent will progressively have to deviate their own proposal closer to the “zero-cost” solution of the opposite UASSP agent.

The N_{Max} value is the maximum number of rounds to drive the costs of the proposed solutions for the given pair of UASSP agents from their “zero-cost” proposal to their “maximum cost” proposal. The “maximum cost” proposal represents the response solution to the “zero-cost” proposal of the opposite UASSP agent, whereby the UASSP agent will solve all conflicts by itself.

In our negotiation process, at each round, the number of conflicts to be solved by each UASSP agent to generate its new proposal is determined by adding a constant increment to δ_i , which is determined as a function of the maximum number of conflicts and N_{Max} (see Algorithm 2).

So, if the process reaches the N_{Max}^{th} round, each UASSP will propose to solve all existing conflicts by itself, which is the same solution as proposed in the prioritization approach. Hence, the negotiation process will terminate in at most N_{Max} rounds.

In practice, an agreement between UASSPs can be found well before reaching this last N_{Max}^{th} round, and thus the actual number of rounds is much lower than N_{Max} . The value of N_{Max} is determined empirically based on the average number of conflicts between UASSPs given their number of operations and the practical computational costs.

The larger the value of N_{Max} is, the more precise the increment on the proposals at each round will be. Thus, the final solution will be closer to the optimal Pareto solution. However, if this value is too large, then this would require a heavier load of communications between UASSPs before reaching an agreement.

C. Generalization to Multilateral Case

We have presented two different decentralized approaches to resolve conflicts between pairs of UASSP agents. For more than two UASSP agents, each agent would need to communicate with all of the other agents and wait to collect all of their responses and vice versa for each involved agent, until an agreement is reached that satisfies all agents simultaneously.

However, such process would be complex and impractical to deploy since the amount of communications required to

Algorithm 3 General approach with $K \geq 2$ agents

Data: $(A_k)_{k \in [1;K]}$ UASSP agents with associated set of operations Ω_k^{shared}

Result: Conflict-free paths for all UASSPs

```
/* Pseudo-code for agent  $A_i$  */
OtherBlockedPaths =  $\emptyset$  ▷ List of paths from other agents considered as obstacles
Is_Processed  $\leftarrow$  false;
while true do
  /* Determine maximum number of conflicts with other agents */
   $(A_j; N_{ij}^{Conflicts}) =$ 
  MaxConflictsPair( $(A_k)_{k \in [1;K]; k \neq i}$ ) ▷ Send/Receive update message to/from other UASSPs
  /* If  $A_i$  has the highest number of conflicts with agent  $A_j$  */
  if  $(A_i, A_j) == \arg \max_{(A_k, A_l)_{k, l \in [1;K]; k \neq l}} N_{kl}^{Conflicts}$  then
    /* Apply resolution process with  $A_j$  */
     $\Omega_i^{shared} \leftarrow$  Pairwise_Resolution( $A_i, A_j$ );
    Is_Processed  $\leftarrow$  true; ▷ Send update message to other UASSPs
  else if Is_Processed then
    /*  $A_i$ 's paths become obstacles for the other agents if not re-selected next */
    OtherBlockedPaths  $\leftarrow$ 
    OtherBlockedPaths  $\cup \Omega_i^{shared}$ ;
    Send(OtherBlockedPaths); ▷ Send update message to other UASSPs
  break;
```

converge to an agreement and synchronization issues might be large, especially when many agents are simultaneously involved in conflicts between their operations.

Therefore, we propose to address the $K \geq 2$ UASSP agents case with a *sequential bilateral* approach as shown in Algorithm 3.

In this approach, agents sequentially deconflict between each other in pairs using the previously presented approaches. The sequence of pairwise processing is determined with a predefined criterion that all agents first communicate to determine which pair should first deconflict between each other. We hereby use the number of conflicts detected between agents as a criterion to determine the sequence of processing for all agents.

Our approach is a more practical resolution method for several UASSP agents as the used criterion allows us to reduce the amount of communications needed to converge to the final conflict-free solution for all UASSP agents. This might lead to a lower overall solution quality because of the local, rather than global, aspect of this approach.

Note that in the prioritization approach, the paths of the “winner” of pairwise resolution are declared as obstacles for the other agents, which have not yet been deconflicted between each other. In the negotiation approach, the paths of the

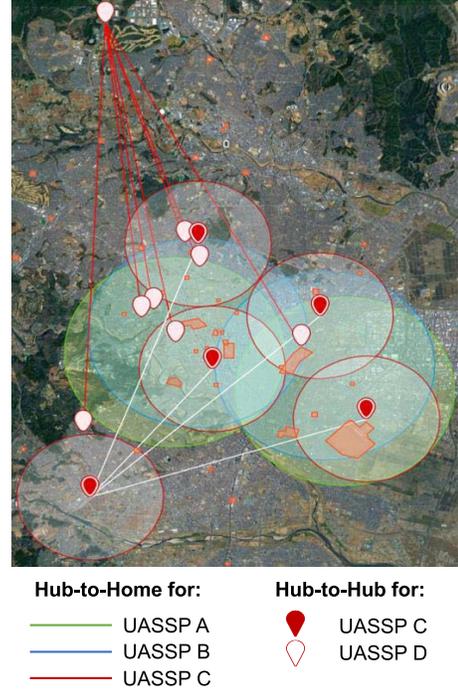


Fig. 2. Map of Sendai 2030 scenario area, including hub locations for all UASSPs and service areas (circles).

UASSP agent with the lowest remaining number of conflicts are declared as obstacles. Those are the ones that are less likely to have an impact on the other agents’ paths.

Moreover, our proposed pairwise negotiation approach aims at maximizing the fairness objective as defined in Section III-E. Thus, we aim at obtaining a solution that is as close as possible to the optimal solution in terms of fairness.

Note that the termination guarantee mentioned in Sect. V-B.4 still holds here, since the multilateral case is a sequence of bilateral processes.

VI. EXPERIMENTS

In this section, we first describe the properties of the model case used for our simulations. Then, we present and analyze the results obtained in our experiments.

A. Model Case Scenario

This section describes the model case used in our experiments. The model case is based on a study that aims to project UAV service demand in the Sendai region in Japan in 2030. The study was conducted by a consulting company, as part of a large-scale governmental project on designing, simulating and specifying the UTM system.

The dimensions of the considered area in the given region are 14.35 km \times 17.10 km. The representation of our space is a 3D grid map composed of cells with edges of 30 meters long. Thus, according to the dimensions of the model case, we consider a 3D grid map of dimensions 478 \times 570 cells, which is delimited in altitude by a two cells range (60 m range) relative to the elevation. The positions of static obstacles, i.e., blocked cells, is fixed according to the given elevation map of

the region. Further, there are 41 distinct no-fly zones identified by the study.

The study considers three major logistics companies (hereby anonymized as A, B and C) that provide deliveries of goods such as mail and package delivery, as well as a Red Cross blood center anonymized as D that collects blood samples between medical centers. The different hub locations are set by the study considering the expected demand in the area in 2030. The assignment of the service locations is assumed to be done independently by UAS operators within a given service radius around a hub. Figure 2 depicts positions of hubs and their associated service area with a different color for each company. Inside these areas, we fix the minimum flight path length to 300 m.

We distinguish two types of deliveries: *Hub-to-Home*, that are deliveries performed from hubs to service locations (homes) located within a given service radius, and *Hub-to-Hub*, that are deliveries performed between given pairs of hubs only. The study indicates a given amount of *Hub-to-Home* and *Hub-to-Hub* deliveries for each of the companies.

In the study, one day of delivery service represents 13 hours, from 8 am to 9 pm. According to customers' demand in the region, expected weight of deliveries and expected capabilities of UAVs, the study estimates that there is a total demand of up to 13,910 operations per day in normal season and up to 21,235 operations per day in busy season, considering the given companies altogether. We consider quadcopter UAVs that have holonomic motion, and can move in any direction at any time, or hover. Each UAV has an attributed speed between 15m/s and 18m/s and a given radius r between 15m and 30m, as defined in Section III.

In this article, we assume that each of the given companies uses its own UASSP, thus four UASSPs are considered in our experiments, and we consider fixed ratios of submitted operations for each UASSP according to the study.

B. Experimental Results

All approaches are implemented in Java within the JADE framework to simulate communications between UASSP agents, and are run on a 3.2GHz Intel Core i7-8700 desktop with 16 GB RAM. The starting times of operations are uniformly distributed within a 1-hour time window. In all experiments, the total number of submitted operations varies from 200 to 1000 operations in total, with N_{Max} set up to 30, and the operations are distributed among the given UASSPs with fixed proportions according to the study.

We evaluate the performances of our approaches with two main metrics:

- Total accumulated delay, i.e. the sum of all delays of all operations of an UASSP agent;
- Amount of operations rejected of an UASSP agent

We also compute the standard deviation in total delays to quantify the difference in the distribution of delays among the UASSPs. 30 instances were generated for each experiment.

1) *Comparison of Prioritization Approach With Randomized Ordering and Cost-Based Ordering*: First, we compare the difference in terms of total accumulated delays between

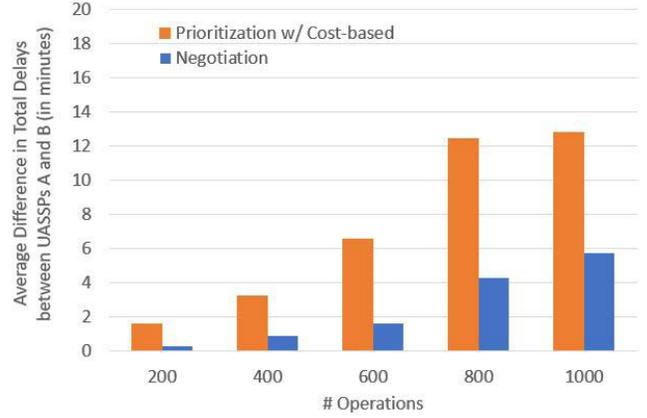


Fig. 3. Comparison of the average difference in total delays among UASSPs A and B between the negotiation approach and the prioritization approach with cost-based strategy.

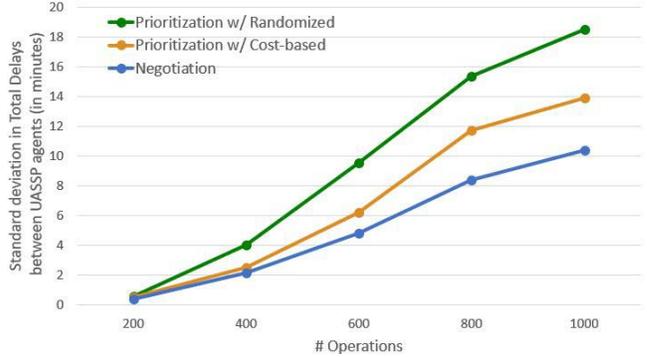


Fig. 4. Comparison of the average standard deviation in total delays among all UASSPs between the negotiation approach and the prioritization approach with (1) randomized and with (2) cost-based strategy.

cost-based ordering as presented in Section V-A and a randomized ordering approach, where agents determine their priorities randomly and replan accordingly. For the randomized strategy, in our experiments, we generated all possible combinations of orders among all four UASSPs. Table III reports the average delays obtained for each UASSP over all the combinations. As expected, the randomized strategy leads to larger delays in average for each agent than the cost-based strategy. Fig. 4 shows a higher standard deviation in total delays between the UASSPs.

2) *Comparison of Prioritization Approach and Negotiation Approach*: We compare the performance of the prioritization with cost-based ordering approach to the negotiation approach. Overall, the impacts of the methods can be observed mainly on UASSPs A and B, which have the highest number of operations and larger overlapping service areas, so most conflicts happen between them. The costs for UASSPs C and D do not significantly change due to their hubs configurations as shown in Fig. 2. UASSP C has less conflicts with the other UASSPs due to the smaller service areas of its hubs, and UASSP D has fewer operations to process.

We observe that the total delays are more uniformly distributed between UASSPs A and B with the negotiation approach than with the prioritization approach as shown in Fig. 3. More precisely, with the use of resolution with negotiation,

TABLE III

COMPARISON OF THE TOTAL DELAYS (IN MINUTES) AND % OF OPERATIONS REJECTED AMONG UASSPS BETWEEN THE NEGOTIATION APPROACH AND THE PRIORITIZATION APPROACH WITH RANDOMIZED STRATEGY AND WITH COST-BASED STRATEGY

#Ops	UASSP	Randomized		Cost-based		Negotiation	
		delay (in min)	% ops rejected	delay (in min)	% ops rejected	delay (in min)	% ops rejected
200	A	1.4	0.3	2.15	0.25	1.25	0
	B	2.4	0.2	0.56	0	1.54	0
	C	0.95	0.3	0.44	0.3	0.58	0.2
	D	0.3	0	0.21	0	0.21	0
400	A	9.3	1.5	7.39	0.33	5.47	0.25
	B	5.7	0.7	4.14	0.3	4.5	0.3
	C	4.55	0.8	4.27	0.42	4.45	0.4
	D	0.47	2.1	0.44	1.9	0.51	0
600	A	22.8	1.4	16.12	0.52	13.1	0.4
	B	14.1	2.7	9.56	2	11.47	1.4
	C	10.2	1.3	8.3	1.1	9.52	0.83
	D	1.8	3.5	1.15	2.8	1.3	2.6
800	A	32.3	0.9	27.4	0.6	22.27	0.5
	B	23.8	2.5	15	1.7	17.56	1.56
	C	15.6	1.7	14.17	1.28	13.7	1.28
	D	0.6	3.8	0.5	3.3	0.5	2.5
1000	A	39.2	1.8	34.3	0.97	29.1	0.22
	B	26.5	4.3	21.51	3.5	23.4	1.92
	C	16.7	2.6	14	2.2	15	1.9
	D	1.5	6.8	1	6.25	1	1.7

the standard deviation in total delays becomes lower than with prioritization with cost-based ordering, in particular from 600 operations (see Fig. 4). This indicates a reduction in the difference in costs between UASSP agents when solving conflicts between their operations, as the bargaining process allows one to share the replanning costs among UASSP agents.

The solution obtained with the negotiation approach aims at minimizing the gaps in total delays between pairs of UASSP agents. That is, the negotiation approach tries to maximize “fairness” and determine a solution close to the Pareto optimal solution, whereby the gap in costs (e.g. in terms of total delays) between all agents is minimal. As observed in our experimental simulations, the average difference in *total* delays between pairs of UASSP agents is around 6 minutes for UASSPs A and B that have the most conflicts on average between their operations (see Fig. 3).

The amount of rejected operations is also slightly reduced with the negotiation approach, shown in Table III, as rejecting an operation has more impact than delaying one operation. Yet, since our considered space is not cluttered but rather an open space, this number remains low. We observe that operations were mostly rejected for the following reasons: either (1) a conflict occurs near the start location of an operation, where the other operation is considered as an obstacle and thus cannot be replanned, or (2) a conflict occurs near a start location because of added delay on some operation.

The trade-off between these two approaches is in terms of processing time and solution quality. While resolution with prioritization generates higher degradation in costs than the resolution with negotiation, its processing is faster than the negotiation approach.

The processing time includes the communication time and the computation time. The communication time is negligible; it is less than a few seconds on average. On the other hand, the computation time represents the major part of the processing time, as it involves the generation of a response to a proposal for each UASSP agent at each negotiation round. This requires replanning paths with the given MAPF solver, and this time

increases with the number of operations as more conflicts occur. However, with a scalable MAPF solver like ECBS, computation time remains low even for a thousand operations.

Both approaches are able to find conflict-free solutions for all UASSP agents in less than 5 minutes on average, which suggests good scalability. Note that Pre-Flight CDR does not have strict deadlines as opposed to In-Flight CDR or Collision Avoidance, so communication and processing are not time critical.

We followed the Sendai 2030 model case, because it is a realistic and representative study of predicted demand of UAV delivery operations provided by a major consulting company. We expect that our methods and results are applicable in similar settings, e.g., a small number of UASSPs, with a large number of operations.

Also, we demonstrated that the negotiation approach always generates higher “fairness” than the prioritization approach, because it allows a more balanced distribution of total delay costs between UASSP agents.

VII. CONCLUSION

The development of a UTM (Unmanned Aircraft System Traffic Management) system is required to safely integrate UAVs in low altitude airspace. UAS operators will task UAVs to perform flight operations for different applications, such as surveillance, delivery, and so on. Thereby, they will rely on UAS Service Providers (UASSPs) to provide conflict-free paths for their UAVs.

Since several independent UASSPs share the same airspace, and conflicts between UAV operations belonging to different UASSPs need to be resolved, Pre-Flight CDR (Conflict Detection and Resolution) methods are necessary to strategically provide 4DT (3D space+time) conflict-free flight paths before UAVs take off to perform their operations.

In this article, we extend the MAPF (Multi-Agent Path Finding) framework to model the Pre-Flight CDR problem. Since the standard, *centralized* MAPF formulation does not make any distinction between UASSP agents, we extend the MAPF formulation by supporting an individual cost function for each UASSP agent. The objective for each UASSP agent is thus to determine a conflict-free solution for their operations while minimizing their individual cost. These cost functions aggregate delays generated for each operation in conflict and the amount of operations rejected.

In response to the requirements of the UTM community [28], we present a novel *decentralized* resolution method for MAPF instances between UASSP agents. This method which aims at ensuring “fairness” in the distribution of costs, mainly in terms of total delays among all UASSP agents. Specifically, we introduce two efficient decentralized algorithms to address this problem: (1) resolution with prioritization as a baseline method and (2) resolution with negotiation as a practical method.

We experimentally compare the proposed approaches in a realistic model case based on a real world study of UAV delivery service in the Sendai region in Japan projected for 2030. Resolution with prioritization with a cost-based strategy

for ordering among UASSPs allows one to generate solutions with on average lower costs for all UASSPs, as compared to a randomized ordering. Then, our results suggest that resolution with negotiation generates a better distribution in costs among all UASSP agents, in particular in total delays, than resolution based on prioritization with a cost-based strategy.

Thus, unlike the standard MAPF objective, our negotiation approach allows each UASSP agent to reduce its individual costs, and also improves “fairness” among UASSP agents.

Since Pre-Flight CDR methods are not time critical, the negotiation process can start ahead as early as possible, as long as the UASSP agents are able to converge to a solution before the actual start of their operations. In particular, in our case study in Sendai, we assumed that there are up to 1600 operations in total per hour, and in our experiments we also considered a 1-hour time window for the scheduled start times of the submitted operations. Assuming that operations are uniformly submitted in time, we thus consider that an average of 800 operations are submitted in 30 minutes. Therefore, a sufficient and convenient time interval to start the negotiation process between UASSP agents could happen every 30 minutes, so that a significant number of submitted operations can be processed.

Finally, we hope that our work on Pre-Flight Conflict Detection and Resolution can contribute to a candidate solution for UTM technology, with the inclusion of individual cost functions to reflect the economic aspects of each UAS Service Provider.

REFERENCES

- [1] J. Alonso-Mora, T. Naegeli, R. Siegwart, and P. Beardsley, “Collision avoidance for aerial vehicles in multi-agent scenarios,” in *Autonomous Robots*. New York, NY, USA: Springer, 2015, pp. 101–121.
- [2] O. Amir, G. Sharon, and R. Stern, “Multi-agent pathfinding as a combinatorial auction,” in *Proc. 29th AAAI Conf. Artif. Intell. (AAAI)*, 2015, pp. 2003–2009.
- [3] M. Barer, G. Sharon, R. Stern, and A. Felner, “Suboptimal variants of the conflict-based search algorithm for the multi-agent pathfinding problem,” in *Proc. 7th Annu. Symp. Combinat. Search (SOCS)*, 2014, pp. 19–27.
- [4] S. Bhattacharya, “Distributed optimization with pairwise constraints and its application to multi-robot path planning,” in *Robotics, Science and Systems VI*, vol. 177. Cambridge, MA, USA: MIT Press, 2011.
- [5] Z. Bnaya, R. Stern, A. Felner, R. Zivan, and S. Okamoto, “Multi-agent path finding for self interested agents,” in *Proc. Symp. Combinat. Search (SOCS)*, 2013, pp. 1–9.
- [6] A. Felner *et al.*, “Search-based optimal solvers for the multi-agent pathfinding problem: Summary and challenges,” in *Proc. 10th Int. Symp. Combinat. Search (SOCS)*, Jun. 2017, pp. 29–37.
- [7] A. G. Foina, C. Krainer, and R. Sengupta, “An unmanned aerial traffic management solution for cities using an air parcel model,” in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Jun. 2015, pp. 1295–1300.
- [8] A. Gilboa, A. Meisels, and A. Felner, “Distributed navigation in an unknown physical environment,” in *Proc. 5th Int. Joint Conf. Auto. Agents Multiagent Syst. AAMAS*, 2006, pp. 553–560.
- [9] F. Ho, R. Galdes, A. Goncalves, M. Cavazza, and H. Prendinger, “Improved conflict detection and resolution for service UAVs in shared airspace,” *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1231–1242, Feb. 2019.
- [10] F. Ho *et al.*, “Pre-flight conflict detection and resolution for UAV integration in shared airspace: Sendai 2030 model case,” *IEEE Access*, vol. 7, pp. 170226–170237, 2019.
- [11] F. Ho, A. Salta, R. Galdes, A. Goncalves, M. Cavazza, and H. Prendinger, “Multi-agent path finding for UAV traffic management,” in *Proc. 18th Int. Conf. Auto. Agents Multi-Agent Syst. (AAMAS)*, 2019, pp. 131–139.
- [12] W. Hoenig, S. Kiesel, A. Tinka, J. W. Durham, and N. Ayanian, “Conflict-based search with optimal task assignment,” in *Proc. 17th Int. Conf. Auto. Agents Multi-Agent Syst. (AAMAS)*, 2018, pp. 757–765.
- [13] *Global Air Traffic Management Operational Concept*, 1st ed, document 9854, ICAO, Montreal, QC, Canada, 2005.
- [14] Y. I. Jenie, E.-J.-V. Kampen, C. C. de Visser, J. Ellerbroek, and J. M. Hoekstra, “Selective velocity obstacle method for deconflicting maneuvers applied to unmanned aerial vehicles,” *J. Guid., Control, Dyn.*, vol. 38, no. 6, pp. 1140–1146, Jun. 2015.
- [15] Y. I. Jenie, E.-J. van Kampen, J. Ellerbroek, and J. M. Hoekstra, “Taxonomy of conflict detection and resolution approaches for unmanned aerial vehicle in an integrated airspace,” *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 3, pp. 558–567, Mar. 2017.
- [16] E. Kalai and M. Smorodinsky, “Other solutions to Nash’s bargaining problem,” *Econometrica*, vol. 43, no. 3, pp. 513–518, May 1975.
- [17] P. Kopardekar, K. Bilimoria, and B. Sridhar, “Initial concepts for dynamic airspace configuration,” in *Proc. 7th AIAA ATIO Conf, 2nd CEIAT Int. Conf Innov Integr Aero Sci., 17th LTA Syst. Tech Conf., Followed 2nd TEOS Forum*, Sep. 2007, p. 7763.
- [18] P. Kopardekar, J. Rios, T. Prevot, M. Johnson, J. Jung, and J. E. Robinson, “Unmanned aircraft system traffic management (UTM) concept of operations,” in *Proc. AIAA Aviation Technol., Integr., Oper. Conf.*, 2016, pp. 1–16.
- [19] J. K. Kuchar and L. C. Yang, “A review of conflict detection and resolution modeling methods,” *IEEE Trans. Intell. Transp. Syst.*, vol. 1, no. 4, pp. 179–189, Dec. 2000.
- [20] H. Ma and S. Koenig, “Optimal target assignment and path finding for teams of agents,” in *Proc. Int. Joint Conf. Auto. Agents Multi-Agent Syst. (AAMAS)*, 2016, pp. 1144–1152.
- [21] H. Ma *et al.*, “Overview: Generalizations of multi-agent path finding to real-world scenarios,” 2017, *arXiv:1702.05515*. [Online]. Available: <https://arxiv.org/abs/1702.05515>
- [22] H. Ma, J. Li, T. K. S. Kumar, and S. Koenig, “Lifelong multi-agent path finding for online pickup and delivery tasks,” in *Int. Joint Conf. Auto. Agents Multi-Agent Syst. (AAMAS)*, pp. 837–845, 2017.
- [23] M. Machida, “Polynomial-time multi-agent pathfinding with heterogeneous and self-interested agents,” in *Proc. 18th Int. Conf. Auto. Agents Multi-Agent Syst. (AAMAS)*, pp. 2105–2107, 2019.
- [24] Y. Mohammad and S. Nakadai, “Optimal value of information based elicitation during negotiation,” in *Proc. 18th Int. Conf. Auto. Agents Multi-Agent Syst. (AAMAS)*, pp. 242–250, 2019.
- [25] J. F. Nash, “The bargaining problem,” *Econometrica*, *J. Econ. Soc.*, vol. 18, no. 2, pp. 155–162, Apr. 1950.
- [26] P. Pianpak, T. C. Son, Z. O. Toups, and W. Yeoh, “A distributed solver for multi-agent path finding problems,” in *Proc. 1st Int. Conf. Distrib. Artif. Intell. DAI*, 2019, pp. 2:1–2:7.
- [27] A. R. Pritchett and A. Genton, “Negotiated decentralized aircraft conflict resolution,” *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 1, pp. 81–91, Jan. 2018.
- [28] J. Rios, “NASA UTM strategic deconfliction final report,” NASA, Ames Research Center, Washington, DC, USA, Tech. Rep., 2018.
- [29] G. Sharon, R. Stern, A. Felner, and N. Sturtevant, “Conflict-based search for optimal multi-agent path finding,” in *Proc. 26th Conf. Artif. Intell. (AAAI)*, 2012, pp. 563–569.
- [30] G. Sharon, R. Stern, M. Goldenberg, and A. Felner, “The increasing cost tree search for optimal multi-agent pathfinding,” in *Proc. 22nd Int. Joint Conf. Artif. Intell. (IJCAI)*, 2011, pp. 662–667.
- [31] D. Silver, “Cooperative pathfinding,” in *Proc. AAAI Conf. Artif. Intell. Interact. Digit. Entertainment*, 2005, pp. 117–122.
- [32] D. Sislak, P. Volf, and M. Pechoucek, “Agent-based cooperative decentralized airplane-collision avoidance,” *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 1, pp. 36–46, Mar. 2011.
- [33] T. S. Standley, “Finding optimal solutions to cooperative pathfinding problems,” in *Proc. 24th Conf. Artif. Intell. (AAAI)*, 2010, pp. 28–29.
- [34] P. Surynek, A. Felner, R. Stern, and E. Boyarski, “Efficient SAT approach to multi-agent path finding under the sum of costs objective,” in *Proc. ECAI 22nd Eur. Conf. Artif. Intell.*, 2016, pp. 810–818.
- [35] S. Vera, J. A. Cobano, D. Alejo, G. Heredia, and A. Ollero, “Optimal conflict resolution for multiple UAVs using pseudospectral collocation,” in *Proc. 23rd Medit. Conf. Control Autom. (MED)*, Jun. 2015, pp. 28–35.
- [36] S. Wollkind, J. Valasek, and T. Ioerger, “Automated conflict resolution for air traffic management using cooperative multiagent negotiation,” in *Proc. AIAA Guid., Navigat., Control Conf. Exhib.*, Aug. 2004, p. 4992.
- [37] J. Yang, D. Yin, Y. Niu, and L. Zhu, “Cooperative conflict detection and resolution of civil unmanned aerial vehicles in metropolis,” in *Advances in Mechanical Engineering*. London, U.K.: SAGE, 2016.



Florence Ho received the master's degrees in operations research and finance from the University of Paris 1 Pantheon-Sorbonne and in applied mathematics and computer science from the Institut National Polytechnique de Toulouse, France, in 2016. She is currently pursuing the Ph.D. degree with the National Institute of Informatics, Tokyo, Japan. Her current research interests include the conception and the development of in-flight and pre-flight conflict detection and resolution algorithms for UAV operations in the context of unmanned aircraft systems traffic management (UTM).



Benjamin Sportich received the B.S. degree in computer science and mathematics and the master's degree in artificial intelligence, decision theory and multiagents systems from Sorbonne University, France. His current research interests include multiagents and multirobots systems to genetic algorithms and reinforcement learning.



Rúben Geraldes received the B.Sc. degree in information system and computer engineering from the Instituto Superior Técnico, Universidade de Lisboa, Lisbon, Portugal. He was a JST Project Researcher on a big data project with the National Institute of Informatics, Tokyo, Japan. He is currently a Specialist Researcher with the National Institute of Informatics, working on the research and development of a UTM system for the low-altitude airspace in Japan. His main research interests include computer graphics and multimedia systems, distributed systems and IT systems, and human-agent interactions.



Daisuke Kubo received the Ph.D. degree in aeronautical engineering from The University of Tokyo in 2008. He is currently a Senior Researcher working at the Japan Aerospace Exploration Agency (JAXA) in the Unmanned Aircraft Systems Technology Team of Aeronautical Technology Directorate. He has more than 15 years of experience in unmanned aircraft systems (UAS), including a variety of technology fields flight dynamics, flight control, airframe design, and operations technologies, and a wide range of platform size from an indoor micro aerial vehicle to a large high-altitude long endurance UAS conceptual design. He has organized the research community on UAS operational technology in Japan beyond the boundary of organizations and held workshops and published survey articles.



Artur Gonçalves received the master's degree in electrical and computer engineering from the Instituto Superior Técnico, University of Lisbon, Portugal, in 2015. Since 2016, he has been with the National Institute of Informatics, Tokyo, Japan, where he integrates a team working on unmanned aircraft systems traffic management (UTM) research and system development. He is focused on developing system prototypes and visualization tools. His research interests include computational geometry and cyber-physical systems.



Marc Cavazza received the Ph.D. and M.D. degrees from Paris Diderot University. He is currently a Professor and the Head of the School of Computing and Mathematical Sciences, University of Greenwich. One of his main research interests is the use of artificial intelligence techniques for virtual agents, a topic on which he has published extensively at relevant conferences, such as AAMAS. He has recently been involved in work applying non-admissible heuristic search that has appeared at IJCAI and in *Frontiers in Neuroinformatics*. He has coauthored over 300 publications and has regularly served on the program committees of AAMAS, ACM Multimedia, and ACM Intelligent User Interfaces.



Bastien Rigault received the master's degree in distributed and embedded systems from Pierre and Marie Curie University, Paris, France. He is currently working as an Assistant Researcher with the National Institute of Informatics, Tokyo, Japan. He participates in the research and development of a UTM system targeting low-altitude airspace in Japan, studying pre-flight phase and in-flight phase automated solution and airspace management concepts. His research interests include distributed application and algorithm, UNIX operating system development, and machine learning.



Helmut Prendinger (Member, IEEE) received the master's and Ph.D. degrees in logic and artificial intelligence from the University of Salzburg, Austria. He held positions as a research associate and a JSPS post-doctoral fellow with The University of Tokyo. In 1996, he was a Junior Specialist with the University of California at Irvine, Irvine. He is currently a Full Professor with the National Institute of Informatics, Tokyo. His current research interests include unmanned aircraft systems traffic management (UTM) and machine learning (ML), especially deep learning for drone use cases. His team contributes to developing the entire UTM system as part of the large-scale Japanese government project. He has published more than 230 refereed papers in international journals and conferences. His H-index is 38.