Improved Conflict Detection and Resolution for Service UAVs in Shared Airspace

Florence Ho, Ruben Geraldes, Artur Gonçalves, Marc Cavazza, and Helmut Prendinger

***Abstract*—In future UAV-based services, UAV (Unmanned Aerial Vehicle) fleets will be managed by several independent flight operation service providers in shared low-altitude airspace. Therefore, Conflict Detection and Resolution (CDR) methods that can solve conflicts—possible collisions between UAVs of different service providers—are a key element of the Unmanned Aircraft System Traffic Management (UTM) system. As our CDR method, we introduce an adaptation of ORCA, which is a state-of-the-art collision avoidance algorithm hitherto mainly used in a limited theoretical scope, to realistic UAV operations. Our approach, called Adapted ORCA, addresses practical considerations that are inherent to the deployment of UAVs in shared airspace, such as navigation inaccuracies, communication overhead, and flight phases. We validate our approach through simulations. First, by empirically tuning the ORCA parameters look-ahead time window and deconfliction distance, we are able to minimize the ORCA generated deviations from the nominal flight path. Second, by simulating realistic UAV traffic for delivery, we can determine a value for separation distance between UAVs that uses airspace efficiently.**

***Index Terms*—UAV Coordination, UTM, Conflict Detection and Resolution, Task Allocation, Path Planning**

I. Introduction

ONE of the preconditions for the successful real-world deployment of Unmanned Aerial Vehicle (UAV) fleets is the development of a safe and efficient Unmanned Aircraft System (UAS) Traffic Management (UTM) system [1]. In the near future, several independent UAS Service Providers (UASSPs) will task multiple UAVs with limited capacities to visit specific locations, and operate in low-altitude and possibly high-density airspace that is shared among the UASSPs.

For safety, the path of each UAV must avoid *static* obstacles, such as terrain elevation and no-fly zones, and *dynamic* obstacles, such as other UAVs controlled by other service providers. With dynamic obstacles, Conflict Detection and Resolution (CDR) methods become indispensable for safe air traffic. In UTM, and similar in ATM (Air Traffic Management), CDR methods refer to different levels of "redundancy" [1], [2], [3].

- Pre-Flight CDR: here, conflicts are detected and resolved based on flight plans submitted to the UTM, before the actual flights. In ATM, such method is called Strategic Conflict Management.
- In-Flight CDR: because of changing weather conditions, or some emergency, sone conflict-free paths created pre-flight CDR might not be safe anymore. Therefore, in-flight CDR methods are required that adapt UAVs movement in real-time, during the flight. In ATM, such techniques are called Separation Provision.
- See and Avoid (SAA): those methods provide a final failsafe and might be based on video processing or other sensing technology

In this paper, we will focus on In-Flight CDR methods. Please note that those methods might be centralized (as in our paper), or decentralized, e.g., running onboard the vehicle.

Additionally, the design and specification of a complete UTM system will require many considerations [2], [1], some of which are not yet determined, such as requirements for communication, application of dynamic airspace configuration, and so on.

Our implementation of the UTM system simulator comprises the following main components of a real UTM system.[1]

- UAS Service Provider (UASSP): all UAS Operators use a UASSP for task assignment (incl. path planning) of UAVs given service requirements;
- Core UTM: all UASSPs connect to a central Core UTM that hosts the CDR (Conflict Detection and Resolution) service for in-flight CDR.

In this paper, we focus on airspace shared among UAVs from different, independent UAS service providers, whereby all UAVs are assumed to be connected to the UTM via UASSPs. Moreover, in the low-altitude shared airspace, interactions with manned aircraft are limited, and airports are considered as no-fly zones for UAVs, so we do not consider manned aircraft in this setting. We adopt a partly simplified setting with respect to UAV types, and consider only one model of quadcopter for which we take realistic flight parameters into account.

The operational scenarios considered in this paper are mainly the delivery of goods by UAVs from different UASSPs, including commercial and public services. For that purpose, we present a two-stage approach to UAVs delivery operations in shared airspace [4]. First, in the *preparatory phase*, for all UAVs connected to an UASSP, paths are planned that avoid collisions with static obstacles, such as terrain elevation and no-fly zones. Note that in this step, we do not consider the possibility of conflicts, even within the fleet of UAVs controlled by one UASSP. Second, in the *in-flight phase*, as a standard approach in UTM [2], the CDR method modifies

F. Ho, R. Geraldes, A. Gonçalves and H. Prendinger are with National Institute of Informatics, Tokyo, Japan (e-mail: florence@nii.ac.jp, rubengeraldes@gmail.com, artur.alves.goncalves@gmail.com, helmut@nii.ac.jp).

M. Cavazza is with Greenwich University, London, UK (e-mail: m.cavazza@greenwich.ac.uk).

[1] This setup is also realistic in terms of current regulatory environment.

trajectories of those UAVs, within or across fleets, that are at risk of collision.

While most of the recent works focus on adapting ORCA's computations to obtain collision-free maneuvers for UAVs, the novelty of our approach is in proposing a practical integration of ORCA as an in-flight CDR mechanism to the UTM system. The Standard ORCA algorithm is usually applied in theoretical simulations with simplifying assumptions that do not hold in practice, such as instant velocity changes or perfect communication. So, the main challenge of our work is to effectively adapt Standard ORCA to address real-world constraints and thus make ORCA applicable to the UTM context.

We also conduct simulations of a real area in Japan with realistic settings regarding the demand for UAVs and delivery tasks.

Specifically, this paper makes three main contributions:

1) We are the first to adapt Standard ORCA to the practical UTM context, where quadcopter UAVs operate in shared low-altitude airspace. With our new version of ORCA, called Adapted ORCA, we address the communication overheads induced by Standard ORCA, the UAV navigation inaccuracies, and the distinct UAV flight phases, such as take-off and landing.
2) For super-conflict scenarios, we propose an empirical tuning of ORCA parameters, namely the look-ahead time window τ, and the deconfliction distance dec_dist, to reduce the path deviation generated by ORCA.
3) We also evaluate the performance of our approach in a series of realistic simulations based on real-world delivery scenarios. With the collected simulation data, we show that our approach ensures zero percent physical collisions.

The rest of the paper is structured as follows. Section II presents related works and provides some background on the techniques used in this paper. Section III introduces our problem. Sections IV and V describe our two-phase process that comprises a preparatory phase and an in-flight phase. Further we also introduce our adaptation of ORCA in the presented setting. Section VI presents our simulations scenarios, and Section VII shows the overall results and analysis. Section VIII concludes our paper.

II. Related Works

A. Task Allocation and Path Planning

The occupancy of airspace by UAVs is determined by their flight paths, which are established through a two-step process: first, task allocation decides on the sequence of destinations for the UAV (and associated actions) and second, path planning designs an optimal trajectory under terrain and airspace constraints.

The task allocation problem is a widely studied combinatorial optimization problem, which has generated many approaches and variants that have mostly been studied in isolation, independently of the constraints of real-world deployment, such as route planning for UAVs.

In our context, for each UASSP, the objective is to task UAVs to go to several locations, which is a Multi-Depot Vehicle Routing Problem (MDVRP) [5]. It is a generalization of the Traveling Salesman Problem (TSP) for multiple vehicles. In most of the formulations, the costs of going from one location to another are known beforehand [5], [6]. In our case, a single UAV has to avoid static obstacles on its path, such as no-fly zones, and it is necessary to take into account terrain elevation, which is relevant considering the altitude at which UAVs operate. Thus, a path planning process that computes collision-free paths representing the real costs in-between locations is necessary [7], [8].

We propose to incorporate this information through an elevation map, thereby implementing 2.5D path planning on top of a cell-based discretization for optimal coverage [9]. Preparatory (static) path planning can be solved traditionally with a heuristic search algorithm using distance-based heuristics such as A*. Our discretization method produces paths that may not constitute optimal trajectories from a kinematic perspective as often reported in robotics [10]. In the specific case of UAVs, despite their additional movement abilities, an optimized trajectory is likely to be more energy-efficient by limiting hovering and turning. We have thus resorted to the Theta* [11] variant of A*, as this any-angle approach is more suitable for quadcopters.

B. Conflict Detection and Resolution (CDR)

Kuchar and Yang [12] provide an extensive survey on in-flight CDR methods for aircrafts and defined the notion of conflict as "an event in the future in which two or more aircrafts will experience a loss of minimum separation between each other" ([12], p. 4). Airspace segregation and corridor concepts are already applied for aircrafts [3] and can also reduce the number of conflicts. However, the possibility of conflicts is still non-negligible in the presence of dynamic events such as wind, tracking errors, or flight delays. Several works centered on UAV deployment have recently emerged, such as the ORCHID project [13], [14], which focuses on optimally assigning tasks to UAVs in a disaster response context. However, their work targets only one single UAS Operator, who is in control of the entire UAV fleet, whereas we focus on an airspace shared among multiple *independent* service providers, so that conflicts cannot be ruled out beforehand. [15], [16] and [17] proposed a coupling of task allocation and path planning for UAVs. Yet, they did not address collision avoidance between moving UAVs. Moreover, they only consider a 2D context, thus assuming UAVs keep a constant altitude. By contrast, we take into consideration 3D elements, based on a realistic elevation map.

Regarding UAV collision avoidance methods, several approaches have been proposed. [18] apply a speed change method to solve conflicts but are limited by frontal conflicts and uncertainties. In a similar way, [19], [20] use speed and heading changes restricted to 2D; hence they do not consider any altitude change. In the AgentFly project, [21], [22] propose decentralized algorithms for collision avoidance based on game theory, but they do not consider a realistic UTM setting with task allocation or static obstacles. With priority-based techniques [23], [2], UAVs are ordered into a sequence and planning is done one-by-one, such that each UAV avoids collisions with the higher-priority UAVs. This greedy approach tends to perform well in uncluttered environments, but it is in general incomplete and often fails in dynamic environments. With knowledge-based methods [24], a lookup table of conflict resolution is generated offline with the use of a Markov decision process [24] and then applied online. The generation usually requires large computation time.

Recently, the computation of collision-avoiding velocities with the use of velocity obstacles shows widespread adoption [25], [26], [27], [2], [28], [29]. Among them, Optimal Reciprocal Collision Avoidance (ORCA) [30] is the prominent approach.

In our UTM context, we focus on developing in-flight CDR methods for UAVs. ORCA is a method originally applied in simulations of collision avoidance for holonomic agents, such as pedestrians or ground robots. In this paper, we propose an adaptation of ORCA, so it can be incorporated into a multi-layer CDR approach.

Several other works have recently also applied ORCA to the case of UAVs by adapting its computations. For instance, Alonso-Mora et al. [31] have incorporated motion equations for quadcopters into ORCA's computations to obtain more precise collision-free maneuvers for UAVs. Yet, all these works focus on designing a local collision avoidance method without considering aspects of the UTM context, such as automated UAVs that carry out missions, position uncertainty, or safe separation between UAVs. By contrast, we propose a novel version of ORCA that supports CDR in a realistic UTM context.

III. Problem Formulation

In this section, we define the overall problem. Our main objective is to ensure safe UAV operations in shared airspace, i.e., no collisions between any UAVs or with static obstacles, and compliance with UTM requirements while providing minimum evasive behavior in case of conflict.

The environment is assumed to be known and the state space is in three dimensions. Hence a location will have three coordinates: latitude, longitude and altitude relative to the elevation from mean sea level. UAVs are allowed to fly between legal bounds that are hereby fixed to $min_{alt} = 30$ meters (m) for the minimum altitude, and $max_{alt} = 150$m for the maximum altitude relative to the elevation of a point of given latitude and longitude coordinates.

Our problem is directly motivated by several practical applications like delivery or surveillance tasks. We assume the dynamic situation where an operator in charge of a given fleet of UAVs continuously receives tasks requests at regular time intervals.

A. Task Allocation

We consider the sub-problem of task allocation for each UAV fleet. In this sub-problem, the objective is to minimize the total sum of traveled distances for all considered UAVs of the fleet.

For each UAV fleet, we define:

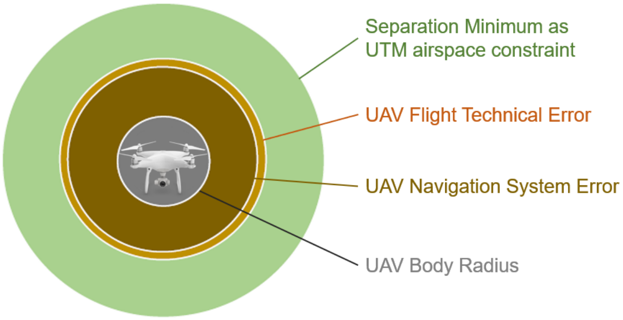

Separation Minimum as UTM airspace constraint
UAV Flight Technical Error
UAV Navigation System Error
UAV Body Radius

Fig. 1: Safety layers considered for the determination of the total separation distance

- A set of m UAVs $U = \{1, \dots, m\}$: each UAV $i \in U$ has an initial location l_i considered as the first take-off point and also the final location, since typically each UAV has to return to its hub once its mission is completed.
- The battery capacity of each UAV $D_i > 0$ which allows the UAV to travel a maximum distance.
- The payload capacity of each UAV $P_i > 0$ for carrying goods.
- A set of n tasks $T = \{T_1, \dots, T_n\}$: each task $T_j \in T$ has a required payload $p_j \geq 0$ and a given location.
- The cost function we use in the following is defined as the distance to travel from one given task location T_i to the location of another task T_j: $d(T_i, T_j)$.

We denote the path of a UAV i as being the set of p ordered waypoints: $\pi_i = \{(x_1, y_1, z_1), \dots, (x_p, y_p, z_p)\}$. The altitude constraints are defined as:

$$\forall (x_k, y_k, z_k) \in \pi_i, \quad \begin{aligned} elevation(x_k, y_k) + min_{alt} \leq z_k \\ z_k \leq elevation(x_k, y_k) + max_{alt} \end{aligned} \tag{1}$$

The following assumptions apply to the sub-problem:

- Each UAV returns to the same initial location of departure.
- Each task is considered as a destination location and is reached exactly once by a UAV.
- The total distance traveled does not exceed the UAV's battery capacity.
- The payload demand does not exceed a UAV payload capacity.

These are the assumptions used in our task allocation algorithm.

B. Conflict Detection and Resolution

As previously mentioned, a *conflict* is the predicted loss of minimum separation between two or more aerial vehicles [12].

In the UTM context, similar to the ATM context, no physical collisions are ever acceptable. For that purpose, the actual body radius of each UAV is enlarged by adding extra layers that represent (i) different uncertainty margins, such as navigation system error and flight technical error, and (ii) the minimum separation distance determined as as a UTM airspace constraint (see Fig. 1).

To accurately determine the individual radii, we introduce an expression that incorporates a first layer that we call the $Constructor$ layer, whose value can be retrieved from UAV manufacturers data (see Eq. 2). It includes the actual size of a UAV, $BodyRadius$, and several navigation error factors relating to operational risk as listed in [32], such as the Navigation System Error NSE, which is the difference between the true position of a UAV and its displayed position, the Flight Technical Error FTE, which is the difference between the required flight path and the displayed position of a UAV, and the $SafetyFactor$, an arbitrary safety margin coefficient.

$$Constructor = (BodyRadius + NSE + FTE) \cdot SafetyFactor \tag{2}$$

Then, a second safety layer called $SeparationMinimum$ is added, which is an arbitrary distance whose value is evaluated in this paper. Such layer is also adopted by the UTM regulation [1], [33]. In the case of quadcopters, which can be considered as holonomic agents, the impact of using a specific PID controller and an adaptive fuzzy method would be minimal and have little influence on our CDR approach.

Finally, the *UAV-specific radius* r is shown in Eq. 3.

$$r = Constructor + 0.5 \cdot SeparationMinimum \tag{3}$$

Now, a *near-collision* happens when the spheres representing the separation minima of all involved UAVs actually intersect.

Conflict resolution is triggered when a conflict is predicted and has to ensure that there is no loss of minimum separation, i.e., the separation minima of UAVs never intersect.

C. Approach

While recent works using UAVs focus on the specific computations of collision-free maneuvers, the novelty of our work is the integration of ORCA as a CDR mechanism in the UTM context, which features special requests that need to be addressed.

We first present our two-stage approach used for UAV operations in shared airspace. An example of the overall pipeline is shown in Fig. 2. In the *preparatory phase*, for each fleet of UAVs connected to an UASSP, paths are generated that avoid collisions with static obstacles, such as terrain elevation or no-fly zones. In this step, we do not consider the possibility of conflicts, even within each fleet of UAVs controlled by an UASSP. Then, in the *in-flight phase*, as a standard approach in UTM [2], [33], our CDR method modifies the nominal trajectories for UAVs of all fleets, to avoid loss of minimum separation.

IV. Preparatory Phase: Task Allocation and Flight Path Generation

We present our *Preparatory* procedure (see Fig. 2), which aims at generating a feasible sequence of tasks for each UAV according to a defined cost function.

The simultaneous computation of task allocation and path planning can quickly become intractable even for relatively small problem instances [17]. However, due to scalability considerations, an optimal planner is not necessary. To improve the efficiency of the computations, we first prune unlikely allocations with a soft clustering approach [34]. Such soft clustering is advantageous, as it encodes uncertainties on data, namely the unknown real distance to be computed by path planning, and allows tasks to belong to more than one cluster.

As mentioned in Section II, we use Theta* [11] for any-angle path planning to match with quadcopters motion. For each pair of locations of the same cluster, we compute all paths with Theta* and their associated costs. We use a generated elevation map, thus the search space is comparable to a 2.5D representation and each node in the grid contains its latitude, longitude, and elevation. Also, no-fly zones can be incorporated by putting an infinite cost in the evaluation function. The corresponding waypoints and costs for the paths between every pair of task locations are stored in a lookup table for each UAV.

Finally, we apply Tabu Search (TS), a metaheuristic that efficiently solves large sized optimization problems [35]. TS is an improved version of the Local Search which incorporates a memory of a fixed number of recent moves used, called “tabu list”. The size of the tabu list is dynamically adjusted in function of the improvement of the current solution [35]. With the real path costs previously computed by Theta*, TS determines the sequences of tasks to be allocated to each UAV with respect to the constraints presented in Section III and detailed in Algorithm 1.

One important step in the TS metaheuristic is the neighbourhood selection process. This step determines the strategy to guide successively the search process towards an optimum. In our case, a move means assigning a task from one UAV to another, or changing the order of assignment between two tasks assigned to the same UAV.

This layered approach between task allocation and path planning prevents from a combinatorial growth in the number of possible permutations with the number of UAVs and tasks. The result is a set of task allocations and individual flight paths for each UAV, which is compatible with their real-world speed, payload and battery life.

V. In-Flight Phase: Conflict Detection and Resolution (CDR)

The flight paths associated to task allocations are calculated without any consideration of potential airspace conflicts between UAVs. Hence there is a need to implement suitable in-flight CDR mechanisms.

In this section, we introduce our CDR process called *Adapted ORCA*, which improves over the existing ORCA algorithm [30] hereby referred to as *Standard ORCA*. Importantly, Adapted ORCA is compatible to real-world UAV deployment. The proposed centralized method can be complemented by decentralized on-board CDR methods as a redundancy mechanism.

We first introduce the ORCA parameter notation in Table 1. For a detailed description of the ORCA velocity computations, we refer to [30] .

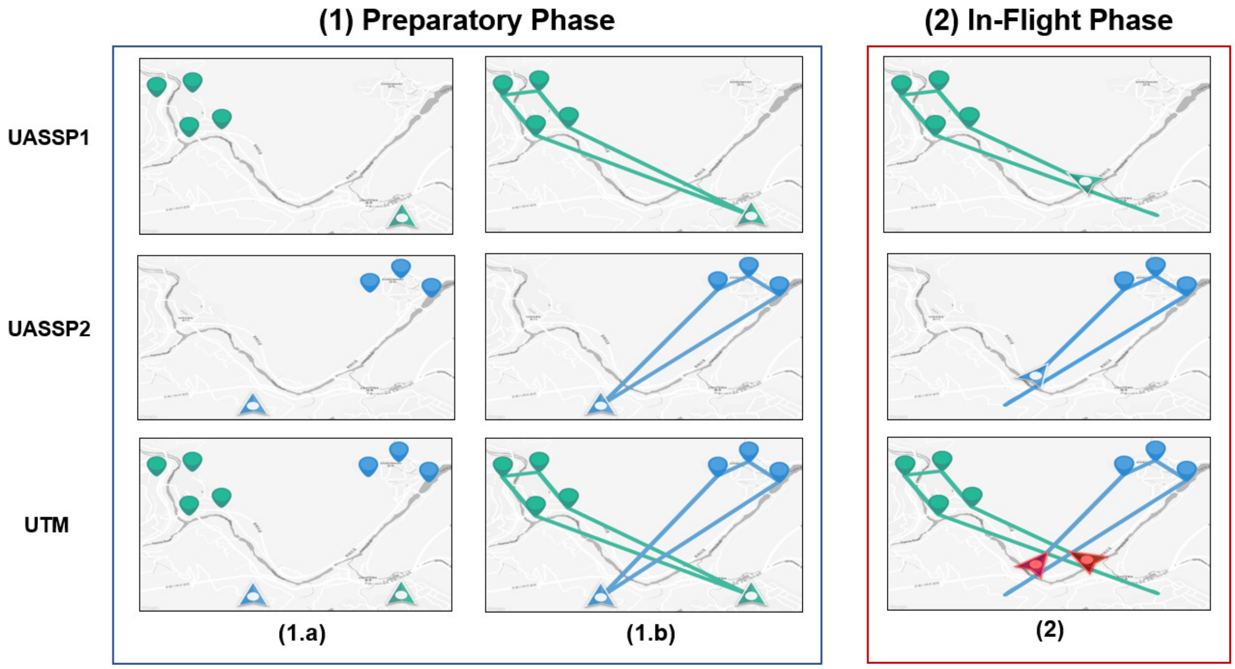


Fig. 2: Example application of our two-stage approach with 2 UASSPs. *Preparatory Phase*: (1.a) Tasks requests are received by each UASSP at regular time intervals and path costs are precomputed with 2.5D path planning; (1.b) Tasks are allocated and paths are generated for each UAV; *In-flight Phase*: (2) UAVs start flying in shared airspace and all UAVs transmit their telemetry to the Core UTM system that hosts the in-flight CDR algorithm that resolves conflicts for all UAVs in the given area.

Standard ORCA has been proven (i) to guarantee collision-free motion for the given time window τ and (ii) to generate a local minimum deviation path [30]. Our proposed CDR method, Adapted ORCA, conserves the theoretical properties of Standard ORCA, while supporting its application to real-world UAV deployment.

In the following, we describe in detail the features of our Adapted ORCA as a CDR approach and discuss the differences with Standard ORCA.

A. *Conflict Detection in Adapted ORCA*

In this section, we describe our conflict detection process that improves upon Standard ORCA. As previously mentioned, a conflict is defined as a predicted loss of minimum separation.

In Standard ORCA, each agent applies the velocity computed by the algorithm at each time step. Velocity obstacles are computed at each time step to detect possible conflicts as shown in Fig. 3, and a new velocity is computed with respect to these constraints. However, this approach results in a possible overhead of computations and communication. If there is no conflict, the same velocity as the current velocity agent is computed.

Thus, we propose to avoid this overhead by introducing a conflict detection mechanism that is composed of two steps: a shallow step and a detailed step.The mechanism is also described in Algorithm 2.

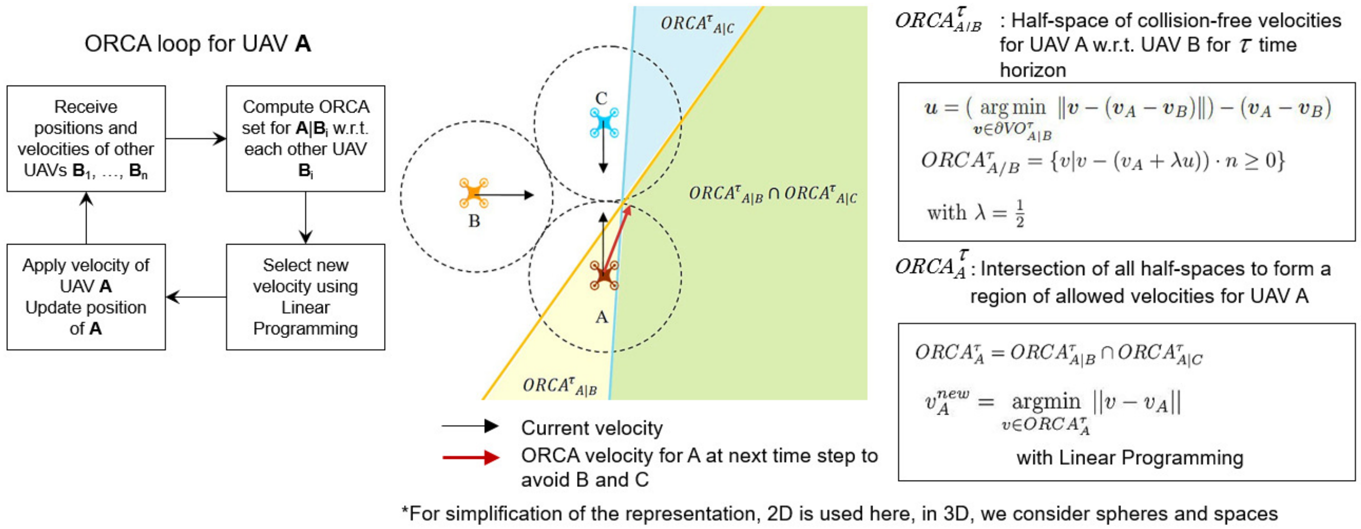
1) Shallow Step in Conflict Detection: The shallow step identifies potential conflicts by the use of a fixed distance called *deconfliction distance* dec_dist, which is a center to center point distance between two UAVs. It is a basic threshold distance, also originally used in Standard ORCA to filter agents that are too far apart, such that at time t, if the positions of two UAVs A and B are such as: $dist(p_A(t), p_B(t)) > dec\_dist$, no computations are made and thus any potential conflict is ignored. Based on the physical interpretation of dec_dist, we can define a lower bound on its value as follows:

$$dec\_dist \geq sep\_dist + \tau \cdot (2 \cdot MaxSpeed) \tag{4}$$

with $MaxSpeed$ the given maximum speed of each UAV.

However, the use of the deconfliction distance dec_dist alone cannot identify false-positives, i.e., a potential conflict was declared, but the UAVs would never collide, e.g., because they fly in parallel. We refine the conflict detection phase to address this issue.

2) Detailed Step in Conflict Detection: At time t, if the shallow step identified a potential conflict, we perform a more precise conflict detection computation over the τ time window by comparing the difference in magnitude between the velocity v^{ORCA} and the preferred velocity of the UAV v^{pref}, which is also the current velocity, as in Eq. 5. This assumes that a UAV will keep its current velocity during the time window τ, which is generally the case as delivery UAVs tend to travel at constant speeds for most of their journey and external effects such as wind operate on a larger timescale than our sampling rate.


ORCA loop for UAV **A**
Receive positions and velocities of other UAVs **B**$_1$, …, **B**$_n$
Compute ORCA set for **A**|**B**$_i$ w.r.t. each other UAV **B**$_i$
Select new velocity using Linear Programming
Apply velocity of UAV **A** Update position of **A**
$ORCA^\tau_{A|C}$
C
B
A
$ORCA^\tau_{A|B} \cap ORCA^\tau_{A|C}$
$ORCA^\tau_{A|B}$
Current velocity
ORCA velocity for A at next time step to avoid B and C
$ORCA^\tau_{A|B}$: Half-space of collision-free velocities for UAV A w.r.t. UAV B for τ time horizon
$u = (\underset{v \in \partial VO^\tau_{A|B}}{\arg\min} \|v - (v_A - v_B)\|) - (v_A - v_B)$
$ORCA^\tau_{A/B} = \{v | v - (v_A + \lambda u)) \cdot n \geq 0\}$
with $\lambda = \frac{1}{2}$
$ORCA^\tau_A$: Intersection of all half-spaces to form a region of allowed velocities for UAV A
$ORCA^\tau_A = ORCA^\tau_{A|B} \cap ORCA^\tau_{A|C}$
$v^{new}_A = \underset{v \in ORCA^\tau_A}{\text{argmin}} ||v - v_A||$
with Linear Programming

*For simplification of the representation, 2D is used here, in 3D, we consider spheres and spaces

Fig. 3: Standard ORCA: ORCA loop at each iteration and an example of the ORCA computation of the collision-free half-spaces and the obtained collision-free velocities for three UAVs in conflict.

Algorithm 1: Task allocation process with Tabu Search

```
Data: $T$: tasks ; $U$: UAVs
Result: $z_{best} \leftarrow (\mathscr{T}_i, \Pi_i)_{\forall i \in U}$: solution as a sequence of
        tasks and associated paths for each UAV i
$(C_i)_{\forall i \in U} \leftarrow FuzzyClustering(U, T, \gamma)$; // Cluster
 of tasks locations for UAV i
$M \leftarrow PathPlanning((C_i)_{\forall i \in U})$; // Lookup table
 containing all costs and paths
 computed by Theta* between pairs of
 tasks locations for UAV i
Generate an initial feasible solution $z_0$ of cost($z_0$) with a
 greedy search;
$z_{best} \leftarrow z_0$;
$z \leftarrow z_0$;
TabuList $\leftarrow \emptyset$;
iter $\leftarrow 0$;
while (iter $< Max_{iter}$) do
    Find the best solution $z'$ of the current neighbourhood
    with move $m' \notin TabuList$ and $cost(z')$ in $M$;
    TabuList.push($m'$);
    if $cost(z') \leq cost(z_{best})$ then
        $z_{best} \leftarrow z'$ ;
    if $TabuList.size > MaxTabuSize$ then
        TabuList.removeFirst();
    $z \leftarrow z'$;
    iter++;
```

$$\text{if } (||v^{ORCA} - v^{pref}|| > \epsilon) \text{ then } inConflict = true \qquad (5)$$

In case of conflict, the velocity computed by ORCA is transmitted to the given UAVs to avoid a loss of minimum separation in τ seconds, else nothing is transmitted and the UAV keeps its current velocity.

Parameter	Description
Preferred velocity v^{pref}	Velocity of a UAV directed towards its goal if there was no conflict
Time step Δt	Time step of each ORCA iteration which is set to the same value as the UAV telemetry update rate (5Hz)
Time horizon τ	Arbitrarily fixed time window in which the velocity computed by ORCA is guaranteed to be collision-free.
Radius r	Radius of the sphere surrounding a UAV
Separation distance $sep\_dist = 2r$	Center to center point distance between 2 UAVs which is used to declare a *near* collision in case of violation
Deconfliction distance dec_dist	Center to center point distance between 2 UAVs which is used to filter unlikely conflicts.
Reciprocity coefficient λ	Percentage of the repartition in deviation between 2 UAVs in conflict: $\lambda = 0.5$ for each UAV in case of equal reciprocity
ORCA velocity v^{ORCA}	Velocity computed by ORCA algorithm at each time step, which is guaranteed to be collision-free for a fixed time horizon τ

TABLE 1: ORCA parameters

B. *Conflict Resolution in Adapted ORCA*

Following the conflict detection phase, the conflict resolution phase is triggered. ORCA computes collision-free velocities iteratively and transmits them to the given UAVs at a fixed rate (see Fig. 3). Note that ORCA avoids any knock-on or "domino" effect since it takes into account all the UAVs that are in conflict with each other during the time window τ in the area considered. A new collision-free velocity is computed with the intersection of the collision-free spaces for each UAV involved in a conflict at each iteration of Adapted ORCA.

Next, we turn to explaining concepts that have been newly integrated to Adapted ORCA, and were not present in Standard ORCA.

1) Start and End of Conflict Resolution: While conflict detection runs permanently, conflict resolution is a conditional event triggered by conflict detection. This creates an alternative behavior between a non-conflict state where a UAV follows its initially planned trajectory, and a conflict state, where a UAV

Algorithm 2: Adapted ORCA

```
Data: $U$: UAVs
for $u_i \in U$ do
    while $p_i != Goal_i$ do
        Update $p_i$;
        for $u_j (j \neq i) \in U$ do
            // Conflict Detection
            if $dist(p_i, p_j) \leq dec\_dist_i$ then
                if $||v_i^{ORCA} - v_i^{pref}|| > \epsilon$ then
                    // Conflict Resolution
                    if $u_i.State == u_j.State == In\text{-}flight$ then
                        $\lambda_{i/j} = 0.5$;
                    else if $u_i.State == In\text{-}flight$ && $u_j.State != In\text{-}flight$ then
                        $\lambda_{i/j} = 1$;
                    else
                        $\lambda_{i/j} = 0$;
                    $iter = 0$
                    while $iter \leq \Delta T$ do
                        $v_i \leftarrow v_i^{ORCA}$;   // Apply corresponding ORCA computations to UAV i
                        $iter++$;
```

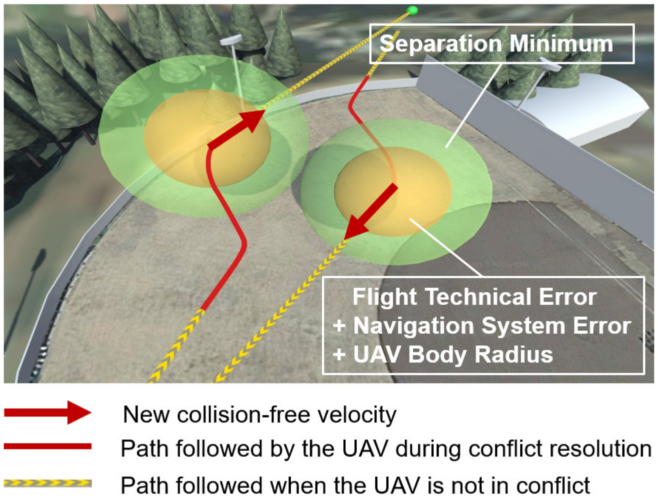

Separation Minimum
Flight Technical Error
+ Navigation System Error
+ UAV Body Radius
New collision-free velocity
Path followed by the UAV during conflict resolution
Path followed when the UAV is not in conflict

Fig. 4: Example of Adapted ORCA solving a conflict between 2 UAVs.

follows newly generated instructions to avoid potential loss of minimum separation.

So, we formalize the notion of “start and end of conflict resolution”, i.e., when and how a path should be modified by ORCA. The start of the CDR step is dependent on the fixed *dec_dist* parameter that defines the distance from where a UAV will receive ORCA velocities to change its original trajectory in case of conflict, as shown in Fig. 4.

When the condition in Eq. 5 turns false, i.e., the velocity computed by ORCA v^{ORCA} for a UAV in conflict becomes sufficiently close to its preferred velocity v^{pref}, the conflict is declared as solved and the UAV resumes its initial path towards its next waypoint.

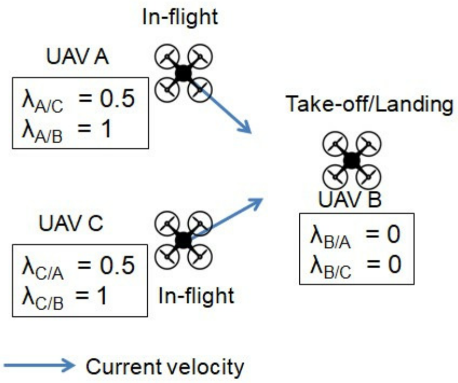


Fig. 5: Example of three UAVs in conflict in a top down view. The state (in-flight, take-off or landing) of each UAV is indicated together with the corresponding values of λ. The velocity of UAV B is not shown as it has vertical direction.

In order to not create an oscillating behavior between non-conflict and in conflict states, this check is done after an arbitrarily fixed minimum conflict duration ΔT, i.e., number of ORCA iterations.

2) UAV Flight States: UAVs can have three flight states. (1) *Take-off*, (2) *In-flight*, and (3) *Landing*. Because Standard ORCA was initially applied to ground robots, the take-off and landing phases were not considered. There may be conflicts occurring between UAVs that are in-flight and UAVs that are taking-off or landing.

In the UTM context, those specific phases of a UAV flight plan cannot be interrupted. Adapted ORCA addresses this situation by dynamically adapting the reciprocity coefficient λ parameter to each type of conflict situation considering the status (in-flight versus take-off or landing) of each UAV as also described in Algorithm 2. When a UAV is in take-off or landing state, it should not be deviated as a rule, since these phases can require specific and more constrained maneuvers. So the UAV in in-flight state takes full responsibility to avoid lack of minimum separation.

For this purpose, the value of the λ reciprocity coefficient of ORCA has to be dynamically changed depending on the conflict situation, since it constrains the set of permitted velocities when computing the solution velocity v^{ORCA} for a UAV (Fig. 5). Currently, we do not consider scheduling to avoid such conflicts.

3) UAV Acceleration Constraints: UAVs have physically imposed limits both on their speed and on their acceleration. Hence, we add a constraint taking into account the maximum acceleration $MaxAcc$, so that the new velocity v generated by ORCA is reachable:

$$||v^{ORCA} - v(t)|| \leq MaxAcc \cdot \Delta t \qquad (6)$$

C. Empirical Tuning of Parameters in Adapted ORCA

Standard ORCA has several parameters that can affect its performance. Specifically, the look-ahead time window τ

associated with the deconfliction distance dec_dist can affect the performance of an ORCA based CDR method because it influences the velocity computations, and thus the deviation trajectory.

Thus, unlike Standard ORCA, which arbitrarily fixes the values of these parameters, Adapted ORCA can be empirically tuned to perform optimally in certain scenarios. Section VII will introduce a super-conflict scenario and demonstrate such empirical fine-tuning.

VI. Simulation Scenarios

To determine the potential of a real world deployment of our CDR approach, we developed a simulation platform that simulates realistic service UAV operations. We use Monte Carlo simulations to provide a statistical evaluation of our method. Those simulations allow to assess the extent of the losses of minimum separation by varying the values of the minimum separation. Since real-world flight experiments are not practical at this time, Monte Carlo simulations are the most viable method to validate our CDR approach.

The following control parameters were used both in the super-conflict scenario and the real-world scenario. First, the total separation distance $sep\_dist = 2r$ (assuming same-type vehicle) with possible values of UAV-specific radius r as presented in Eq. 3 are determined with:

- $BodyRadius = 0.3$m (size of a DJI Phantom 4 UAV)
- $NSE = 2.0$m (GPS standard accuracy)
- $FTE = 0.5$m (empirically measured for DJI Phantom 4)
- $SafetyFactor = 1.1$ (arbitrarily fixed)

Hence, the resulting $Constructor$ layer has 3m radius.

Then, the $SeparationMinimum$ values hereby tested range from 4m to 24m. Thus, applying Eq. 3, we consider a range of values for r from 5m to 15m in the following simulations.

Second, for the time window τ, the optimal value to trigger our CDR method is empirically determined in our simulations. We have been experimenting with values between 4s to 10s.

Third, for the deconfliction distance dec_dist, the value of must satisfy the lower bound in Eq. 4, and thus we consider a range of values from 30m to 45m in our experiments.

A. Super-Conflict Scenario

We created extreme conflict scenarios in which up to 10 UAVs placed in antipodal positions and located in a small area of 200m × 200m are heading to a single collision point. We aim to derive performance parameters for ORCA, whereby the values of dec_dist and τ efficiently trigger conflict resolution and show that deviation from the initial trajectory can even be reduced.

In this scenario, the ϵ parameter used for conflict detection in Eq. 5 is fixed to 0.05, and the minimum conflict duration ΔT is fixed to 10 iterations of ORCA algorithm.

B. Real-world delivery Scenario

We propose experimental scenarios based on a real world setup of the specific region of Okutama, a rural area in Japan as shown in Fig. 6.

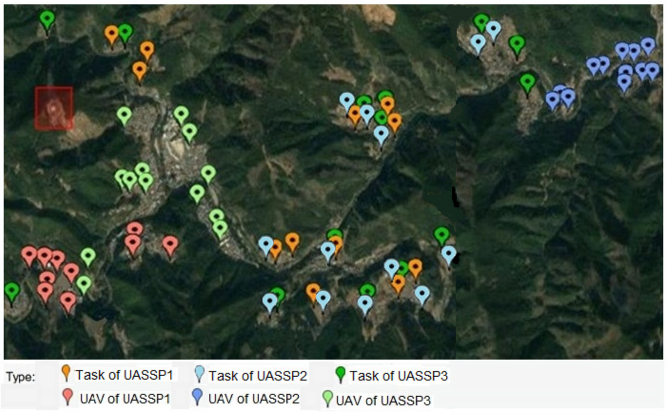


Fig. 6: Map visualization of the Okutama area showing the UAVs initial take-off locations and task locations used in our simulations. There are fleets of three UAS Service Providers (indicated by red, blue, green) with 35 UAVs in total. UAVs are dispatched to three distinct zones that have commercial and/or health facilities. A total of 39 task locations are positioned in isolated areas that are within the range of the UAVs of each fleet. There is also one no-fly zone (red square to the left).

We performed our simulations in an area of about $8km^2$. The rural area of Okutama can be a first potential candidate for real-world deployment since airspace in urban areas is not available under current legislative requirements.

Our scenarios were conceived based on a real-world preliminary study conducted on Okutama. We collected statistics on various aspects of the region such as age range, population density and repartition. With these figures and through various interviews with local authorities, also considering the topography of the area, we have designed our delivery scenarios as shown in Fig.6.

Consequently, our study reflects realistic expectations for UAV service in a given airspace. For instance, the number of simultaneous UAVs in the airspace was derived from the expected number of deliveries, average flying time and hours of operation. The considered initial take-off locations for all UAVs are derived from the locations of shops, hospitals and other service facilities in the area.

In those scenarios, several UASSPs are in the same area, each in charge of their own fleet of UAVs. We fix each UAV capacity (26 min of maximum flight duration as a lower bound with a maximum speed of 5m/s, and a maximum acceleration of 3m/s^2), and maximum payload (1kg), as defined by current quadcopters manufacturers specifications.

The UAVs used in our simulations are DJI Phantom 4 copters have an average battery capacity of 26 min. We put a margin of 10 % on the actual battery consumption limit in the generated flight paths of each quadcopter to ensure that there is sufficient battery in case of deviation during the flight. The reliability of these parameters have been tested in real world experiments with experimental flight tests.

The telemetry update time step Δt is fixed at 0.2s. We located each task so that at least one UAV of the corresponding fleet can reach it as shown in Fig. 6. We also added the constraint that two UAVs' initial locations and tasks locations cannot be closer to each other less than the defined $s\ ep\_dist$.

During the simulations, batches of tasks among the possible task locations of Fig. 6 are randomly generated for each fleet at fixed time intervals. We ran 100 simulation samples for each experiment.

Each simulation represents a 4-hour service scenario, and we assume a fixed small amount of time for battery recharging every time the UAV returns to its initial location. Simulations were ran with an accelerated timeframe so as to reduce simulation time.

In order to avoid a 'consumption 'of UAVs by collisions that would hinder the production of simulation results, we determine that whenever a near-collision takes place (where no CDR is active), those UAVs disappear, and "spare" UAVs are automatically created at the same initial take-off locations of the involved UAVs, but at a different time for each, to not potentially recreate the same collision.

In the scenarios, UAVs are tasked mostly to transport small, light and valuable items, such as medicine, equipment and specific food items. There are three UAS Service Providers (see Fig. 6):

- UASSP1 (UAVs positions in red): Health related delivery and transport (medicines, biological samples) (10 UAVs; 12 tasks locations)
- UASSP2 (UAVs positions in blue): Daily items and food (12 UAVs; 13 tasks locations)
- UASSP3 (UAVs positions in green): Daily items and food (13 UAVs; 14 tasks locations)

All terrain information is provided via the elevation map generated by the existing Mapzen Terrain Tile Service.

The quadcopters flight motions are simulated by our own in-house simulator which uses data acquired from physical trials. At the moment, available UAV simulators are single-computer stand-alone systems that do not scale well and are thus impractical for our broader purpose of simulating large-scale UTM scenarios. Hence, we built our own distributed networked simulator that scales simulations up to thousands of concurrent UAVs.

All algorithms previously described are implemented in Java and we ran our experiments on a 2.9 GHz Intel Core i5-4210 CPU with 16 GB RAM. In our setting, Adapted ORCA exhibits fast runtimes (100ms in average), on the same order of magnitude as Standard ORCA, which has already shown high scalability with applications involving thousands of agents.

VII. Experimental Results

A. Super-Conflict Scenario

The two metrics used in the high density experiment are:

- Average time optimality per UAV. This study aims to evaluate the optimality, or degradation of optimality, in UAVs' evasive behavior.
- Deviation in distance. This study investigates the amount of detour required to avoid loss of minimum separation.

Average time optimality per UAV is formalized as follows. For a UAV j, time optimality degradation is defined as the ratio of the time it would have taken for j to reach its goal location if there was no conflict t^{ideal} to the actual total time taken t^{actual}. Then, the average time optimality per UAV with N referring to the total number of UAVs considered is defined as:

$$Opt_j = \frac{t^{ideal}}{t^{actual}}\ ;\ AverageTimeOptimality = \frac{\sum_{j=1}^{N} Opt_j}{N}$$

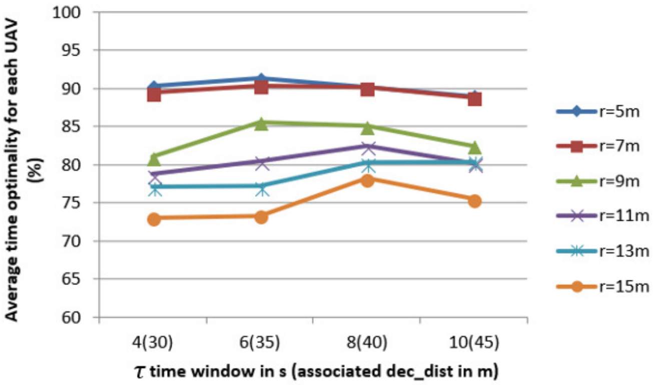


Fig. 7: Evolution of average time optimality for Adapted ORCA in extreme conflict scenarios with MaxSpeed = 5m/s and different values of r relative to τ and associated dec_dist values.

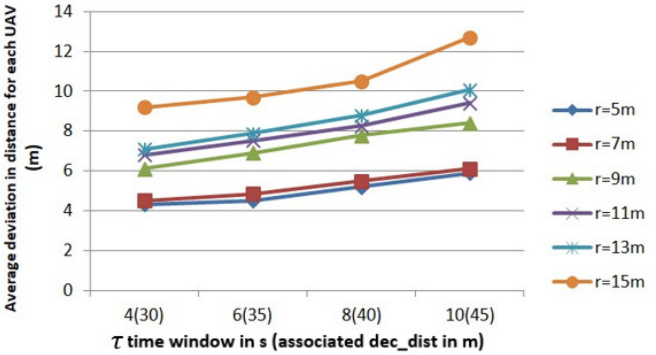


Fig. 8: Evolution of average deviation in distance for Adapted ORCA in extreme conflict scenarios with MaxSpeed = 5m/s and different values of r relative to τ and associated dec_dist values.

Deviation in distance, similar to time optimality, measures the difference between the actual distance traveled and the ideal distance a UAV would have traveled based on their initial (optimal) paths.

In our first study, we will investigate the influence of τ and dec_dist parameters on ORCA resolution. The value of the look-ahead time horizon τ, and by extension dec_dist, influence the quality of the resolution.

Since those parameters are used in the computation of the new velocity by ORCA (in case of conflict), our hypothesis is that there is a value or interval of values of τ and hence dec_dist for which the deviation from an initial trajectory is minimal while ensuring no loss of minimal separation.

We propose to determine those values empirically through our simulated experiments. We fix MaxSpeed to 5m/s and vary

τ with dec_dist and the separation radius r values. In Fig. 7 and 8, we show that choosing the appropriate values of τ and dec_dist can minimize the deviation with the considered UAVs.

In the case of super-conflict s cenarios, t hese v alues allow us to initiate Adapted ORCA velocity computations with more informed values than choosing an arbitrary value.

With a larger radius r, the UAVs are then considered to be larger moving agents, so the deviation in trajectory is larger, as shown in Fig. 8. Further, the average time optimality in Fig. 7 is inferior for the values of r ranging from 9m to 15m compared to the 5m and 7m values.

We also observe that a larger r value requires a larger τ with the associated dec_dist value to minimize the deviations from the initial trajectories. For instance, for r = 9m, the highest time optimality is reached for a value of τ = 6s (with dec_dist = 35m), whereas for r = 11m, it is reached for τ =8s (with dec_dist = 40m).

Otherwise, for smaller values of τ, we observe a larger degradation in time optimality, since UAVs in conflict do not have sufficient time to react and tend to get slower. For higher values of τ, the UAVs tend to react too early and hence take unnecessary detour.

B. Real-world Delivery Scenario

In the delivery scenario, we study three types of setups:

- Frequency of loss of minimum separation with and without CDR. This study provides an empirical estimation of the likelihood of safety issues in shared airspace.
- Influence of the separation radius on occurrences of loss of minimum separation with and without CDR. Note that only the value for separation minimum is varied, while we take the entire separation radius to allow center point to center point distance measurement.
- Influence of the distribution of UASSPs' fleets on loss of minimum separation with and without CDR.

First, we investigate the frequency of loss of minimum separation with and without CDR. We fix the number of fleets (UASSPs) to 3, the total number of possible task locations to 39 and vary the total number of UAVs from 15 to 35 among the possible locations shown in Fig. 6. The maximum number of UAVs is derived from the preliminary UTM studies in [2] given the rural area and its population density.

We first simulate scenarios without the use of our CDR method and report the near-collisions (violations of minimum separation distance sep_dist) that occurred.

In Fig. 9, considering a separation radius r of 15m, as expected, the total number of near-collisions observed in all the simulations increases with the number of UAVs in the same area, hence with a higher density. More precisely, the number of near-collisions observed increases in a quadratic manner.

As shown in Fig. 9, these realistic simulations also assessed the safety provided by our CDR approach, Adapted ORCA, in particular with the separation distance to ensure no loss of minimum seperation (no near-collisions) and hence 0% of physical collisions.

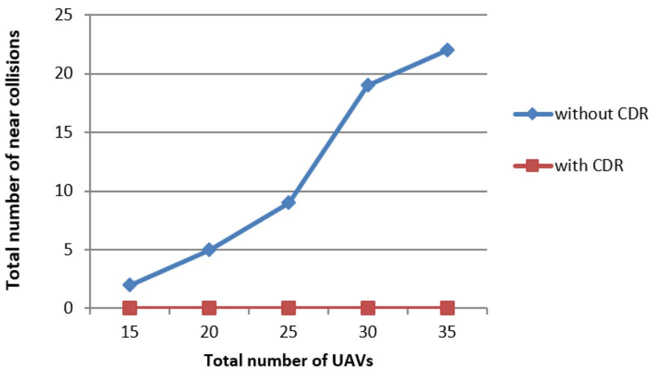


Fig. 9: Impact of CDR on the number of total near-collisions (loss of minimum separation) considering r = 15m.

Let's look at the situation where no CDR method is in place. Most of the near-collisions that happened were in-flight, but near-collisions while one UAV is taking-off or landing also happened to a fewer extent (Table 2) All cases of loss of minimum separation was observed in those scenarios were between 2 UAVs only.

Between UAVs in in-flight state	94 %
Between UAVs in in-flight and take-off/landing state	6 %

TABLE 2: Average % of near-collisions in UAVs' states (with no CDR method implemented).

Moreover, Table 3 shows that most near-collisions happened between UAVs of fleets from different UASSPs. Based on the minimum cost objective of the task allocation, it is unlikely that a conflict occurs between UAVs of the same UASSP, unless there is a large number of tasks concentrated on a particular area, and visiting them all would exceed the capacity of one UAV.

Between UAVs of the same fleet	11.1 %
Between UAVs of different fleets	88.9 %

TABLE 3: Average % of near-collisions within same fleet and between different fleets (with no CDR method implemented).

Second, we study the influence of the separation radius r on the number of near collisions, i.e., violations of minimum separation. We fix the number of fleets (UASSPs) to 3, the total number of UAVs to 35 and the total number of possible task locations to 39, as shown in Fig. 6. Then, we analyze the number of near-collisions for the different values of r = 5,..., 15m, as presented in Fig. 10. The total number of near-collisions increases with the considered radius r.

Given the rationale adopted by UTM regulations, even for the minimum value of r = 5m, near-collisions happen if no CDR method is in place. The use of a CDR method, such as Adapted ORCA, prevents such violations.

Finally, we investigate the influence of the distribution of fleets (UASSPs) on loss of minimum separation. We keep the same scenario with 35 UAVs in total, with r = 15m, and the 39 existing tasks locations in total. Here, we vary the number of UASSPs from three UASSPs with 12 or 13 UAVs each,

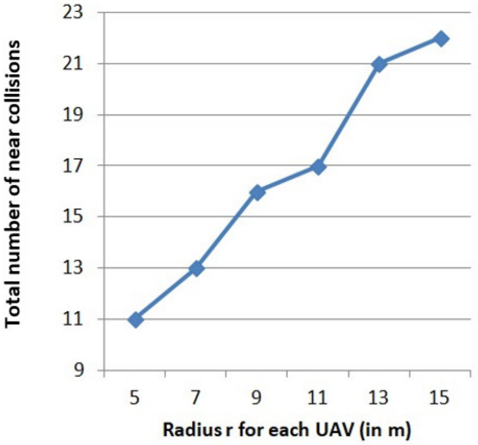


Fig. 10: Total number of near-collisions relative to the radius r

to six UASSPs with 5 or 6 UAVs each, so that there are 35 UAVs in total. We observe in Table 4 that the number of near-collisions increases with the number of UASSPs in the same area. Hence, losses of minimum separation would happen with a larger number of UASSPs rather than having more UAVs in each UASSPs.

Number of fleets	3	4	5	6
Number of collisions	22	22	23	25

TABLE 4: Total number of near-collisions with different number of fleets (with no CDR method implemented).

VIII. Conclusions

Future UAV-based services will require a fully implemented UTM system to ensure safe and efficient operations in low-altitude airspace. UAS Operators will choose from several UASSPs to use flight operation services such as automated task allocation and generation of collision-free flight paths. Yet, the Conflict Detection and Resolution (CDR) method hosted by the Core UTM is needed to solve in-flight conflicts among UAVs of different, independent service providers.

The contribution of this paper is a new CDR method based on ORCA, called Adapted ORCA, which is a practical and effective CDR mechanism for a realistic UTM operational context.

To validate our CDR method, we designed and implemented a simulation platform that runs realistic service UAVs operations. First, we looked at extreme-conflict simulations, and conducted an analysis of ORCA parameters that can affect its solution quality. The simulations yielded optimized values for the key parameters look-ahead time window and deconfliction distance. Those values can be reused in similar situations, such as disaster situation, to optimize the performance of our CDR method based on Adapted ORCA.

Second, we performed extensive simulations on realistic scenarios based on a real world study. With the sample collected, we obtained safety parameters, i.e., the frequency of loss of minimum separation, if no CDR method is implemented. With those simulated scenarios, we were able to study and evaluate the impact of the minimum separation distance value in a realistic set of operations in a rural environment, with the purpose of ensuring physical safety between all UAVs.

Future work will address some simplifying assumptions in this paper. *First*, we currently assume homogeneous UAV fleets consisting entirely of same-type quadcopters. Here, we plan to work on an heterogeneous airspace with UAVs having different dynamics and capabilities, including plane-type UAVs. This will require the study of a generalized CDR system. *Second*, we currently assume reliable communication. So we also want to address situations with no or delayed communication, where vehicle-to-vehicle (V2V) communication with decentralized CDR methods is required. *Third*, we plan to address situations where automated UAVs may encounter human-controlled aircraft, such as hobby drones, or helicopters.

Finally, we hope that our work can provide a first assessment into the research challenges of future UTM-based services and their management via a UTM system.

Acknowledgment

This paper is based on results that were obtained as part from the research conducted by the National Institute of Informatics (NII) in a project recommissioned by the Japan Aerospace Exploration Agency (JAXA), which is commissioned by the New Energy and Industrial Technology Development Organization (NEDO).

References

[1] P. Kopardekar, J. Rios, T. Prevot, M. Johnson, J. Jung, and J. E. Robinson, "Unmanned Aircraft System Traffic Management (UTM) Concept of operations," in *Proceedings of AIAA Aviation Technology, Integration, and Operations Conference*, 2016.

[2] 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 Transactions on Intelligent Transportation Systems*, vol. 18, no. 3, pp. 558–567, 2016.

[3] P. Kopardekar, K. Bilimoria, and B. Sridhar, "Initial concepts for dynamic airspace configuration," in *Proceedings of AIAA Aviation Technology, Integration, and Operations Conference*, 2007.

[4] F. Ho, R. Geraldes, A. Goncalves, M. Cavazza, and H. Prendinger, "Simulating shared airspace for service uavs with conflict resolution," in *International Conference on Autonomous Agents and Multiagent Systems*, 2018, pp. 2192–2194.

[5] J. Montoya-Torres, J. Franco, S. Isaza, H. Felizzola, and N. Herazo-Padilla, "A literature review on the vehicle routing problem with multiple depots," *Computers & Industrial Engineering*, vol. 79, 2014.

[6] A. Farinelli, A. Rogers, and N. R. Jennings, "Agent-based decentralised coordination for sensor networks using the max-sum algorithm," *Autonomous Agents and Multiagent Systems*, vol. 28, no. 3, pp. 337–380, 2014.

[7] S. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.

[8] S. Karaman, M. R. Walter, A. Perez, E. Frazzoli, and S. Teller, "Anytime motion planning using the RRT*," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2011, pp. 1478–1483.

[9] L. D. Filippis, G.Guglieri, and F.Quagliotti, "Path planning strategies for UAVS in 3D environments," *Journal Of Intelligent and Robotic Systems*, vol. 65, pp. 247–264, 2012.

[10] B. Araki, J. Strang, S. Pohorecky, C. Qiu, T. Naegeli, and D. Rus, "Multi-robot path planning for a swarm of robots that can both fly and drive," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2017, pp. 5575–5582.

[11] A. Nash, K. Daniel, S. Koenig, and A. Felner, "Theta*: Any-angle path planning on grids," in *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*, 2007, pp. 1177–1183.

[12] J. K. Kuchar and L. C. Yang, “A review of conflict detection and resolution modeling methods,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 4, pp. 179–189, 2000.

[13] S. D. Ramchurn, T. D. Huynh, Y. Ikuno, J. Flann, F. Wu, L. Moreau, N. R. Jennings, J. E. Fischer, W. Jiang, T. Rodden, E. Simpson, S. Reece, and S. J. Roberts, “HAC-ER: A disaster response system based on human-agent collectives,” in *Proceedings of the 2015 International Conference on Autonomous Agens and Multiagent Systems*, 2015, pp. 533–541.

[14] Z. Beck, L. Teacy, A. Rogers, and N. Jennings, “Online planning for collaborative search and rescue by heterogeneous robot teams,” in *Proceedings of the 2016 International Conference on Autonomous Agents and Multiagent Systems*, 2016, pp. 1024–1032.

[15] S. Kiesel, E. Burns, C. Wilt, and W. Ruml, “Integrating vehicle routing and motion planning,” in *International Conference on Automated Planning and Scheduling*, 2012, pp. 137–145.

[16] L. F. Bertuccelli, H.-L. Choi, P. Cho, and J. How, “Real-time multi-UAV task assignment in dynamic and uncertain environments,” in *Proceedings of American Institute of Aeronautics and Astronautics Guidance, Navigation, and Control Conference*, 2009, pp. 1–16.

[17] A. Richards, J. Bellingham, M. Tillerson, and J. How, “Coordination and control of multiple UAVs,” in *Proceedings of AIAA Guidance, Navigation, and Control Conference*, 2002.

[18] S. Vera, J. A. Cobano, D. Alejo, G. Heredia, and A. Ollero, “Optimal conflict resolution for multiple UAVs using pseudospectral collocation,” in *23rd Mediterranean Conference on Control and Automation*, 2015, pp. 28–35.

[19] J. Yang, D. Yin, and L. Shen, “Reciprocal geometric conflict resolution on unmanned aerial vehicles by heading control,” in *Journal of Guidance, Control, and Dynamics*, 2017, pp. 2511–2523.

[20] J. Yang, D. Yin, Q. Cheng, and L. Shen, “Two-layered mechanism of online unmanned aerial vehicles conflict detection and resolution,” in *IEEE Transactions on Intelligent Transportation Systems*, 2017, pp. 1–15.

[21] D. Sislak, P.Volf, S. Kopriva, and M. Pechoucek, “Agentfly: Scalable, high-fidelity framework for simulation, planning and collision avoidance of multiple UAVs,” *Sense and Avoid in UAS: Research and Applications*, pp. 235–264, 2012.

[22] D. Sislak, J. Samek, and M. Pechoucek, “Decentralized algorithms for collision avoidance in airspace,” in *Proceedings of the 2008 International Conference on Autonomous Agents and Multiagent Systems*, 2008, pp. 543–550.

[23] M. Cap, P. Novak, A. Kleiner, and M. Selecky, “Prioritized planning algorithms for trajectory coordination of multiple mobile robots,” *IEEE Trans. Automation Science and Engineering*, vol. 12, no. 3, pp. 835–849, 2015.

[24] H. Y. Ong and M. J. Kochenderfer, “Short-term conflict resolution for unmanned aircraft traffic management,” in *Proceedings of the 34th IEEE/AIAA Digital Avionics Systems Conference*, 2015.

[25] S. Roelofsen, A. Martinoli, and D. Gillet, “Distributed deconfliction algorithm for unmanned aerial vehicles with limited range and field of view sensors,” in *Proceedings of the American Control Conference*, 2015, pp. 4356–4361.

[26] P. Janovsky, M. Cap, and J. Vokrinek, “Finding coordinated paths for multiple holonomic agents in 2D polygonal environment,” in *Proceedings of the 2014 International Conference on Autonomous Agents and Multiagent Systems*, 2014, pp. 1117–1124.

[27] S. J. Guy, J. Chhugani, C. Kim, N. Satish, M. Lin, D. Manocha, and P. Dubey, “Clearpath: Highly parallel collision avoidance for multi-agent simulation,” in *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2009, pp. 177–187.

[28] P. O. Alexander Alexopoulos, Amr Kandil and E. Badreddin, “A comparative study of collision avoidance techniques for unmanned aerial vehicles,” in *Proceedings of the 2013 IEEE International Conference on Systems, Man, and Cybernetics*, 2013, pp. 1969–1974.

[29] M. Choi, A. Rubenecia, T. Shon, and H. H. Choi, “Velocity obstacle based 3D collision avoidance scheme for low-cost micro UAVs,” *Sustainability*, vol. 9, no. 7, 2017.

[30] J. van den Berg, S. J. Guy, M. Lin, D. Manocha, C. Pradalier, R. Siegwart, and G. Hirzinger, “Reciprocal n-body collision avoidance,” *Robotics Research: The 14th International Symposium ISRR*, vol. 70(1), pp. 3–19, 2011.

[31] J. Alonso-Mora, T. Naegeli, R. Siegwart, and P. Beardsley, “Collision avoidance for aerial vehicles in multi-agent scenarios,” in *Autonomous Robots*, 2015, pp. 101–121.

[32] J. Jung, S. N. Souza, M. A. Johnson, A. K. Ishihara, H. C. Modi, B. Nikaido, and H. Hassee, “Applying required navigation performance concept for traffic management of small unmanned aircraft systems,” in *Proceedings of the International Council of the Aeronautical Sciences*, 2016.

[33] K. Dalamagkidis, K. P. Valavanis, and L. A. Piegl, “On integrating unmanned aircraft systems into the national airspace system,” *International Series on Intelligent Systems, Control, and Automation: Science and Engineering*, vol. 36, 2009.

[34] J. Bezdek, R. Ehrlich, and W. Full, “FCM: The fuzzy C-means clustering algorithm,” *Computers & Geosciences*, vol. 10, pp. 191–203, 1984.

[35] F. Glover and M. W. Laguna, *Tabu Search*. Springer, 1997.

Florence Ho received the Master’s degrees in Operations Research and Finance from University Paris Pantheon Sorbonne, and in Applied Mathematics and Computer Science from Institut National Polytechnique de Toulouse, France in 2016. Since 2017, she is pursuing a PhD degree at National Institute of Informatics, Tokyo, Japan. Her current research focuses on the development of conflict detection and resolution algorithms for UAV operations in the context of Unmanned Aircraft Systems Traffic Management (UTM).

Ruben Geraldes is currently a Specialist Researcher at National Institute of Informatics, Japan, working on the Research and Development of a UTM system for the low-altitude airspace in Japan. Previously, he worked as a JST Project Researcher on a big data project at National Institute of Informatics, Japan. He holds a BSc in Information System and Computer Engineering from Instituto Superior Tecnico, Universidade Lisboa, Portugal. His main interests are Computer Graphics and Multimedia Systems, Distributed Systems and IT Systems, Human-Agent Interactions.

Artur Gonçalves received the Master’s degree in Electrical and Computer Engineering from Instituto Superior Tecnico, Universidade de Lisboa, Portugal, in 2015. Since 2016, he has been at the National Institute of Informatics in Tokyo, Japan, where he integrates a team working on 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 is Professor and Head of the School of Computing and Mathematical Sciences at the 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 co-authored over 300 publications and has regularly served on the program committees of AAMAS, ACM Multimedia and ACM Intelligent User Interfaces. He holds a PhD and an MD from University Paris Diderot.

Helmut Prendinger received the Master's and Doctoral degrees in logic and artificial intelligence from the University of Salzburg, Salzburg, Austria. He is a Full Professor at the National Institute of Informatics, Tokyo. Previously, he held positions as a Research Associate and as a JSPS Postdoctoral Fellow at the University of Tokyo. In 1996, he was a Junior Specialist at the University of California, Irvine. The current focus of his work is Unmanned Aircraft Systems Traffic Management (UTM) and Machine Learning (ML), especially Deep Learning. His team contributes to developing the entire UTM system, including conflict detection and resolution methods, under governmental funding by NEDO. Helmut Prendinger has published extensively; more than 230 refereed papers in international journals and conferences. His h-index is 38.