ENCOUNTER AWARE FLIGHT PLANNING IN THE UNMANNED AIRSPACE

Maxim Egorov, Vanessa Kuroda, Peter Sachs, Airbus UTM, San Francisco, CA

Abstract

Ensuring safe and efficient operations for unmanned aircraft is vital to their integration into the civil airspace. This paper proposes a set of encounter-based metrics that can be used to estimate the complexity and thus the capacity of an unmanned airspace. We show that these metrics are able to provide a more detailed view of safety failure characteristics in the airspace than traditional metrics. The encounter-based metrics are used to formulate a novel path planning algorithm that combines 4D trajectory optimization with a flight scheduling heuristic in order to reduce the number of difficult to solve encounters in the airspace. The proposed algorithm is distributed in nature and can be applied to an airspace managed by multiple service suppliers that are capable of generating flight plans. Through simulation, we show that this approach is able to outperform other distributed flight planning methods and can perform nearly as well as a centralized approach that jointly optimizes flight plans simultaneously. We also demonstrate that this approach can scale to high operational volumes and is able to efficiently handle spikes in airspace usage.

Introduction

From delivery drones to autonomous vertical take-off and landing (VTOL) passenger aircraft, applications of unmanned aircraft systems (UASs) have become a key part of the discussion about the future of our skies [1]. It is estimated that the number of UAS will dramatically increase within the next 20 years [2]–[4], and new approaches will be needed to manage this influx of aircraft outside of traditional air traffic management (ATM) systems. A number of proposals exist for modernizing the traffic management process for UAS under a single framework [5]–[7]. NASA refers to this framework as UAS Traffic Management (UTM) [5], and we adopt their terminology in this paper for consistency. While UTM is a complex system of systems and has many technological challenges that need to be overcome, safe and efficient path planning for flight trajectories is one of the most critical.

Path planning has been studied extensively in a number of applications from robotics [8] and autonomous cars [9] to drones [10] and commercial aircraft [11]. However, the path-planning work done in the context of UTM has been limited to low vehicle encounter densities [12]–[14] or standalone algorithms [15]. In both cases, it is not clear how well the approaches will scale to an airspace with high density UAS operations in a real world setting. In addition to difficult scalability requirements, path planning in the context of UTM may need to be performed in a distributed setting. In the NASA UTM architecture, unmanned flight operations may be managed by different UAS Service Suppliers (USSs) [5]. This would require a distributed approach to flight planning, and would make it more difficult to achieve a globally optimal airspace utilization. In that architecture, strategic deconfliction is defined by the requirement that a UTM operation should be free of 4-D intersection with any other UTM operation prior to departure operation [16], and in this work we aim to address exactly this problem.

We focus on a class of motion-planning algorithms known as trajectory optimization [17]. At a high level, trajectory optimization methods attempt to design a trajectory by minimizing a generalized cost function and meeting a set of constraints. A number of trajectory optimization methods have been developed in recent years that are able to compute collision-free trajectories from a naive straight line initialization [18]–[20]. These methods translate well to path planning in UTM, because prior to optimization a flight-request for a UAS can be modeled as a point to point 3D trajectory that may not be collision-free.

We approach the flight planning problem by first identifying the limitations of existing methods in airspace with large numbers of operations. We focus on unstructured airspace with free-flight mechanics where the only means of traffic management is through flight planning. By first focusing

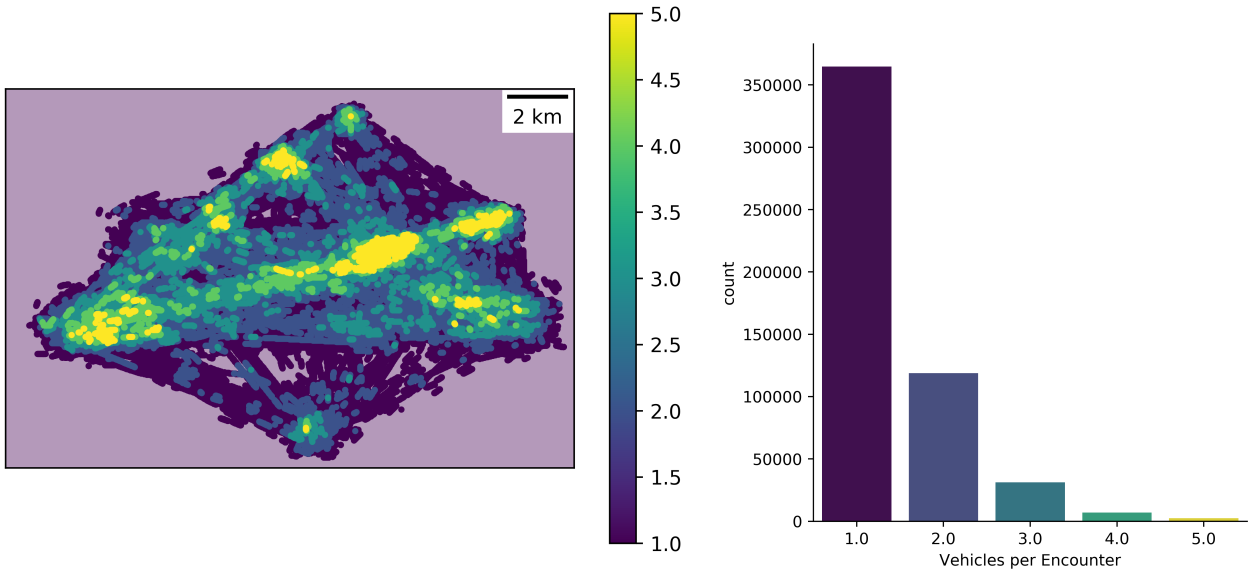

Figure 1. Encounter density map (left) and the corresponding encounter distribution (right) for the multi-modal traffic region.

on individual UAS encounters, we are able to construct a metric that captures macroscopic complexity properties of the whole airspace. The formalism presented here identifies precisely how the high density airspace leads to degrading safety even when state-of-the-art path planning algorithms are in use. While conceptually straightforward, we show that the path-planning problem becomes more difficult to solve as the number of vehicles in individual encounters increases throughout the airspace. By generalizing this notion into a single metric we are able to incorporate it into different planning algorithms and into UTM architectures both with centralized and distributed planning.

The goal of this paper is develop a robust and efficient flight planning approach that can operate in an airspace with high operational volumes while adhering to the constraints of a UTM. The contributions of this work are as follows:

1) To our knowledge, the first quantitative comparison of planning approaches in high density unmanned airspace for distributed and centralized UTM architectures.
2) A set of encounter-based metrics that provide an aggregate estimate of complexity in the unmanned airspace, and an estimated impact of a single flight on airspace complexity.
3) A novel approach to flight planning that incorporates these metrics and is able to significantly outperform the benchmark methods.

The remainder of the paper is organized as follows. We first introduce airspace encounters metrics on which the methodology proposed in this paper is based on, and propose a notion of airspace capacity for unmanned and unstructured flight as well as provide a definition for a dense airspace in the context of UTM. Next, we describe the path-planning problem for UAS, and outline how a distributed or a centralized path planning architecture could work in UTM. We then describe our algorithmic contribution as a trajectory optimization approach combined with an encounter-aware scheduling heuristic. We provide simulation results that compare our algorithmic approach to existing methods, and provide some avenues for future work in the conclusion.

Encounter Distributions and Airspace Impact Metrics

This section defines two metrics for characterizing an airspace, the encounter distribution and the encounter expectation. The encounter distribution captures macroscopic encounter properties of an airspace, while the encounter expectation quantifies the expected encounters for a single flight trajectory.

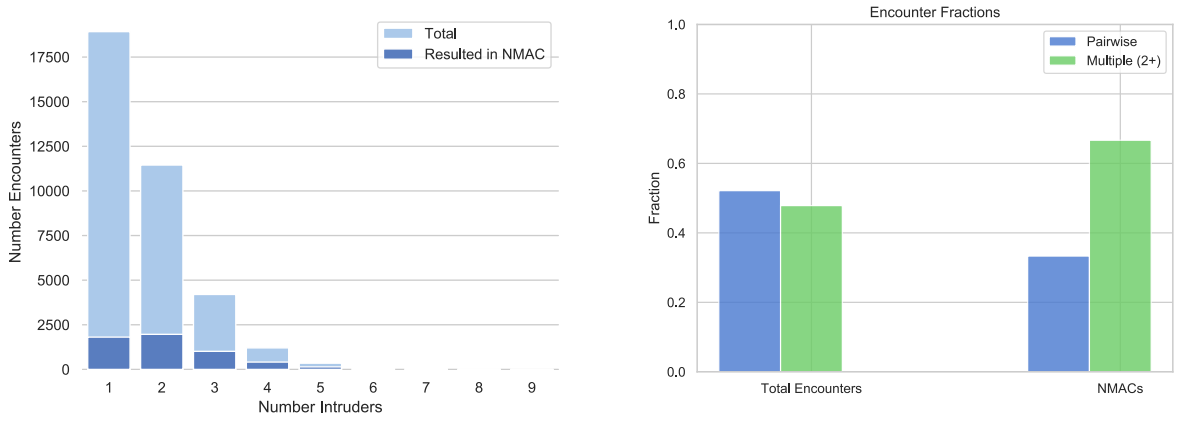

Figure 2. The encounter distribution and the corresponding fraction of loss of separation events (left), and the fractional rate of pairwise and multi-threat encounter (right).

A. Encounter Distributions and Airspace Capacity

Before defining the encounter distribution, we first formalize the notion of an encounter. The most general description is defined by a Boolean function that takes the states of two vehicles in a shared airspace as input, and outputs whether the vehicles are in an encounter or not. Typically, the states include position and velocity of the vehicles, and could also include more complex information such as their expected trajectories as well. In this work, we use a simplified definition where two vehicles within some constant distance of each other are considered to be in an encounter. However, the more general definition is not limited to a constant spherical geometry and can use more complex geometries that could be dependent on time, as well as vehicle position and velocity.

The encounter distribution specifies the probabilities over the types of encounters that exist in an airspace. Specifically, given an encounter in an airspace, it depicts the probabilistic values for the number of vehicles in that encounter. An example of an encounter distribution and a corresponding geographic encounter density map is shown in Figure 1. The map and distribution were generated using results from a four-hour simulation over a multi-modal traffic region at 1000 takeoffs per hour. The multi-modal region is intended to model an urban area with distributed demand clusters for UAV operations and is described in more detail in the Results Section. Note that the encounter distribution can be dynamic and vary with time.

One way to quantify airspace capacity is to estimate the decision making limitations of the system responsible for managing that airspace. A number of metrics have been identified and analyzed to quantify dynamic density in the context of dynamic airspace configuration for the Next Generation Air Transportation System (NextGen) [21], [22]. These metrics attempt to capture the limitation of the cognitive load on air traffic controllers managing an airspace. It's not clear how useful these metrics are in the context of UTM, and we leave that evaluation for future work. Instead, we make a simple conjecture that for a given encounter in the airspace, the path planning problem becomes more difficult as the number of vehicles in that encounter increases. In short, an encounter with more vehicles in it is a complex multi-agent problem and is less likely to be resolved by a planning algorithm. These problems are known to be difficult from a combinatorial standpoint [23], and it is usually impossible to provide tractability or completeness guarantees unless a restrictive set of constraints is met [24], [25]. These conditions are observed in our simulations as well. In simulation, we observe that encounters with more than two vehicles in them contribute disproportionately more loss of separation events than their pairwise counterparts (see Figure 2).

We use the notion of encounter distributions to define an operationally dense airspace by observing that an airspace dominated by encounters with more than two vehicles will be difficult to manage from

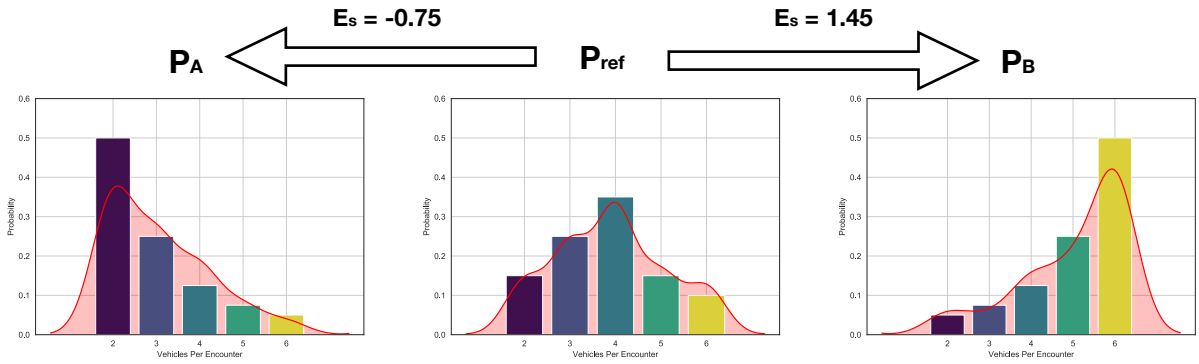

Figure 3. A visualization of the encounter shift from a reference distribution P_{Ref} (middle) to P_A (left) and P_B (right). Minimizing the encounter shift causes the encounter distribution to move towards encounters with a lower number of vehicles.

a planning perspective. Formally, we can define a dense airspace by determining if the expectation for an encounter with more than two UAS is above some threshold value λ, $\mathbb{E}[N_{UAS} > 2] \geq \lambda$, where N_{UAS} is the number of UAS in an encounter. A simple choice is $\lambda = 0.5$, where over half of the encounters have more than two UAS in them. We adopt this value for λ in the remainder of this work. We thus consider an airspace to be operating past its capacity when $\mathbb{E}[N_{UAS} > 2] \geq 0.5$.

The advantage of using encounter distributions to describe the complexity of an airspace lies in their ability to track how different configurations of the airspace impact the encounter distribution and thus the complexity of the airspace. Here, we define a single-valued metric based on total variational divergence that we call the encounter shift. This metric can be though of a signed distance between encounter distributions, and can be used to evaluate the impact that different planning approaches have on the complexity of the airspace. Given two encounter distributions P_A and P_B, the encounter shift is given by

$$E_S(P_A \parallel P_B) = \frac{1}{2} \sum_{x \in \Omega} w_x (P_A(x) - P_B(x)), \qquad (1)$$

where Ω is the support of the encounter distribution and w_x are importance weights associated with the number of vehicles in an encounter. In Equation 1 P_A is the reference encounter distribution, and P_B is the distribution of which we want to measure the encounter shift. This expression differs from total variational divergence in that it is both a signed and a weighted sum. The fact that the expression is signed allows us to track the direction of the shift, and the weights allow us to place more importance on the encounters with greater number of vehicles in them. Figure 3 shows how the encounter shift is affected by the shape of the encounter distribution. When the encounter distribution moves towards encounters with lower numbers of vehicles in them, the shift is negative. A move the other way creates a positive shift. In practice, minimizing the encounter shift also minimizes the complexity of the encounters in the airspace, making it simpler for path-planning algorithms to manage.

B. Encounter Driven Airspace Impact Prediction

We now consider how the encounter shift can be used during path planning to decrease the overall complexity of the airspace. Consider the contribution of total encounters from a single flight plan to the airspace as a whole. This contribution can be divided into two parts:

1) **Immediate Impact**: the encounters a particular flight creates with existing traffic
2) **Future Impact**: the encounters a flight will create with traffic that will be flying in the future

Given a flight plan that leads to N_I^n immediate encounters with n vehicles, it is possible to write the

total expected encounters with n vehicles from the flight as

$$N_n = N_n^I + \mathbb{E}[N_{UAS} = n] \tag{2}$$

Where $\mathbb{E}[N_{UAS} = n]$ is the expected number of future encounters with n vehicles from the flight in question. To compute $\mathbb{E}[N_{UAS} = n]$, we use historical traffic data for the airspace of interest.

The process of computing the expected number of encounters with n vehicles, $\mathbb{E}[N_{UAS} = n]$, can performed by discretizing historical encounter information and performing statistical inference on it. This process allows us to predict the number of encounters that are not immediately accounted for a given flight, i.e. the encounters that may occur from other vehicles that will be flying in the future. Formally, our goal is to compute the full joint probability of all encounters over a given trajectory

$$\begin{aligned} P(e_1^1, e_1^2, \ldots, e_5^5 \mid \text{trajectory}) = \\ P(e_1^1 \mid \text{trajectory}) \times \\ P(e_1^2 \mid \text{trajectory}, e_1^1) \ldots P(e_5^5 \mid \text{trajectory}, e_1^1, \ldots e_5^4) \end{aligned} \tag{3}$$

where e_n^i denotes the i^{th} n vehicle encounter on the trajectory. In the expression we chose to truncate the permutations for i and n at five, although neither i or n need to be bound.

There are a number of methods to compute the joint probability, such as Navie Bayes [26] by making the assumption that all encounters are independent. However, because our primary objective is to estimate the encounter distribution for a single flight, we can instead determine the probability of encounters with n vehicles over a trajectory for n independent values of interest. We can then use those probability values to construct the encounter distribution. If we continue to make the assumption that all encounters are independent events, we can write the probability of a encounters with n vehicles over the whole trajectory as a modified geometric distribution that has the form

$$P(N_{UAS} = n) = p_n^T \prod_{t=1}^{T-1} (1 - p_n^t) \tag{4}$$

where T denotes the total number of timesteps in the trajectory of the vehicle, and p_n^t is the probability of an encounter with n vehicles at time-step t. We can obtain p_n^t using a normalized historical encounter data from the airspace discretized into finite sized 3D bins or voxels (see Figure 4. In this work, we use timesteps that are one second in length. A temporal component can be added to the encounter data if the historical data is expected to vary with time. While such data may not be readily available, we assume it exists in this work, and we obtain it through simulation.

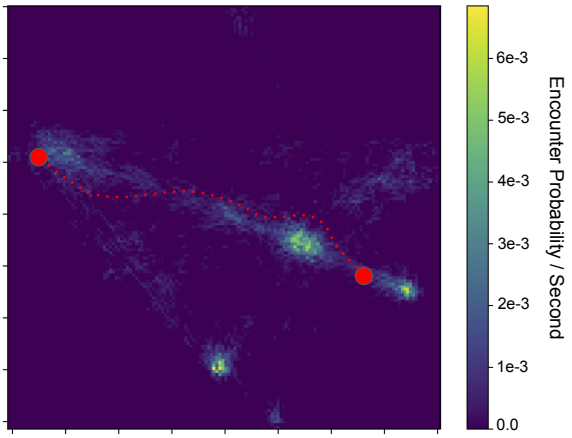

Figure 4. An example trajectory over a geographic heat-map of encounter distribution density

By combining the expression for total expected encounters in Equation 2 with the encounter shift, we can compute the encounter shift between the underlying encounter distribution of the airspace and the one generated for a single flight plan. This metric provides a prediction of the impact a single flight has on the airspace as a whole. While statistically, this approach is prone to effects of variance, we observe that for an airspace simulation with a non-trivial number of flights, it is possible to use this metric to drive the airspace either towards or away from complex operations.

Problem Formulation

This section outlines the flight-planning problem in UTM. We define the general requirements for a flight plan, and describe the mathematical formalism that goes along side of it. We pose the problem in the context of two UTM architectures: one where traffic management capabilities are completely distributed between multiple USSs, and one where centralized flight planning is possible.

C. Collision-Free Flight Plans

In this work, a flight-plan is defined to be a set of 4D waypoints parameterized by (x, y, z, t) that meets the time constraint $t_{i+1} > t_i$ between two waypoints p_i and p_{i+1}. A flight plan is generated during the pre-flight planning phase of an operation, and we leave the problem of in-flight re-routing and contingency planning for future work. Computing a conflict-free flight plan before the flight requires accounting for any uncertainties associated with the trajectories of other UAS in the airspace to ensure no safety violations occur after takeoff. There are a number of other ways to represent a flight plan. A common approach is to use a set of time-bounded 3D polygons around the flight volume, also known as intent [27]. However, this representation is more restrictive and constrains the approaches that can be used for flight planning.

Flight planning can be formulated as an optimization problem, where we attempt to minimize an objective function subject to inequality and equality constraints:

$$\begin{aligned} \text{min} \quad & f(\mathbf{x}) && \text{(5a)} \\ \text{subject to} \quad & g_i(\mathbf{x}) \leq 0, \;\; i = 1, 2, \ldots n_{ineq} && \text{(5b)} \\ & h_i(\mathbf{x}) = 0, \;\; i = 1, 2, \ldots n_{eq} && \text{(5c)} \end{aligned}$$

where $\mathbf{x}$ is a set of waypoints representing the flight plan, and f, g_i, h_i are scalar functions. The objective may consider the risk associated with a trajectory [28], and numerical terms that encourage a minimal length path. Typically, the optimization is performed over a $T \times K$ dimensional vector with T being the number of time-steps and K the number of degrees of freedom. In this work, we treat time as a degree of freedom, and instead have T flight plan waypoints we optimize over.

D. Centralized and Distributed Planning

The choice between centralized and distributed planning typically depends on the effectiveness of communication networks, the available resources, and the constraints imposed by the system architecture. In this work, we make the following distinction between centralized and distributed planning:

1) **Centralized**: a planning approach that can jointly optimize plans simultaneously. This approach requires a single central entity to be responsible for flight planning.
2) **Distributed**: a planning approach that is distributed between multiple entities. Existing plan information can be freely exchanged between entities to ensure deconfliction, but it is not possible to jointly optimize plans.

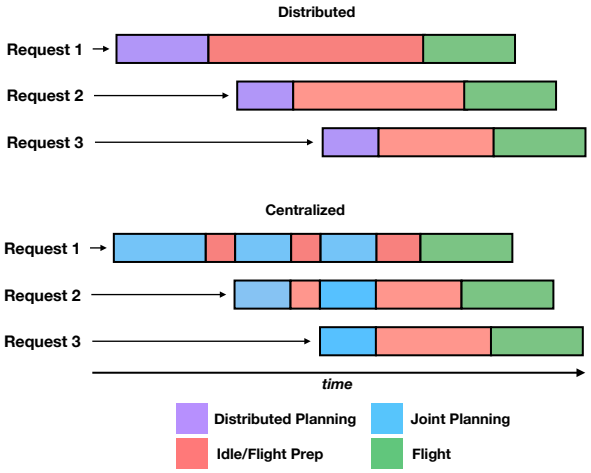

Figure 5. Distributed (top) and centralized (bottom) planning architectures. A flight plan is optimized jointly in the centralized architecture, and individually in the distriubted architecture.

Note that the primary distinction between centralized and distributed planning is the ability of the centralized planner to optimize plans simultaneously. In the context of trajectory optimization, this implies an optimization over a shared objective. While a lot of work has been done on solving optimization problems with a shared objective in a distributed setting [29], [30], the assumptions made in these settings, such as a requirement that the cost function be convex, are usually too restrictive for path-planning algorithms like trajectory optimization methods. Recent work has also shown that coordination between multiple asynchronous planners is feasible [31], but it is not clear how well the approach can scale to relatively short, high-frequency operations like the ones envisioned in UTM. We assume that such methods are too restrictive at this time, and that joint optimization of a single planning objective with a shared set of constraints between distributed planners is not feasible. We thus leave any distributed architectures with capabilities for joint multi-plan optimization as future work.

For a UTM, the ability to perform centralized planning depends on the architecture of the system.

For example, centralized planning is not possible in the NASA UTM architecture unless a single USS is designated as the sole planner for the airspace in a given region. Folding the planning function into a single Air Navigation Service Provider (ANSP) would also enable centralized planning. The intent of the NASA UTM architecture is to be distributed in nature, and that leads to a number of trade-offs from a planning perspective. At a high level, the advantage of a distributed architecture is that it enables multiple entities to independently manage their UAS operations, while adhering to coordination when needed to ensure safety. However, this leads to sub-optimal airspace utilization in regions where multiple USSs are operating simultaneously, and can lead to cascading effects that cause severe airspace congestion in highly used regions [32]. This problem is particularly evident in free-flight operations where multiple USSs may need to plan around each other without any structural constraints in the airspace to organize operations. While negotiation schemes have been discussed in the context of the distributed UTM architecture [16], the implementation and the impact of the negotiation on sub-optimal use of airspace is not clear.

In the centralized architecture, all flight plans are generated by a single planner which allows them to be globally optimal. However, the disadvantage of the centralized approach is that it may not be scalable to high operational frequencies and would require operators to share their mission data with a central entity. It may also be less adaptable to emerging use cases and technologies, making it less desirable from an operators' stand point. The planning cycles for each architecture are shown in Figure 5. In the distributed architecture, plans are computed one at a time if they are owned by different USSs. While completed plans can be shared between USSs, there is no information sharing during the planning process itself. In the centralized architecture, all plans are jointly optimized, because a single entity is responsible for collecting flight requests and computing associated plans. The advantage of the centralized architecture is that it can jointly optimize flight plans as requests come in, for all flights that have not yet started, and it does not require additional negotiation schemes or protocols.

Flight Planning

In this section, we first outline the flight planning algorithm used in this work. We then describe the methodology for incorporating encounter distributions into the flight-planning process. We conclude the section with an overview of how this algorithm fits into the overall UTM architecture, and explain how the distinction between centralized and distributed UTM architectures impacts flight-planning.

E. 4D Trajectory Optimization for Highly Utilized Airspace

We use gradient-based trajectory optimization to compute flight trajectories in this work. One of the biggest challenges in creating collision-free flight plans is that the environment in consideration has both static and dynamic obstacles in it. The static obstacles are used to model constructs like temporary flight restrictions (TFRs) that remain unchanged for the duration of a single flight. Dynamic obstacles include other UAS, and large airspace constructs that are in motion such as weather events, and are more difficult to incorporate into the planning process. While the problem of motion planning with dynamic obstacles has been studied extensively, most approaches focus on online re-planning in static environments [33], velocity obstacles [34], or discrete grid-based search [35]. While these approaches can be applied to flight planning, they are either difficult to scale to continuous state spaces or are most suitable for problems where online re-planning is appropriate.

To naturally handle dynamic obstacles this work treats time as an optimization variable during planning. However, we introduce a heuristic that is able to efficiently approximate the safe time intervals between each waypoint in dynamic environments during optimization. In this way, our approach is similar to safe interval path planning [36], but is capable of scaling to continuous state spaces while remaining computationally efficient. The heuristic uses a distance measure to add or remove 4D waypoints from the flight trajectory as the plan is being generated. The bounds of the measure are based on geometric distance and are set prior to the optimization process. If a segment between two waypoints is not collision free and falls outside of the bounds, the segment is either expanded or contracted. A segment that is geometrically longer than the upper bound, is split

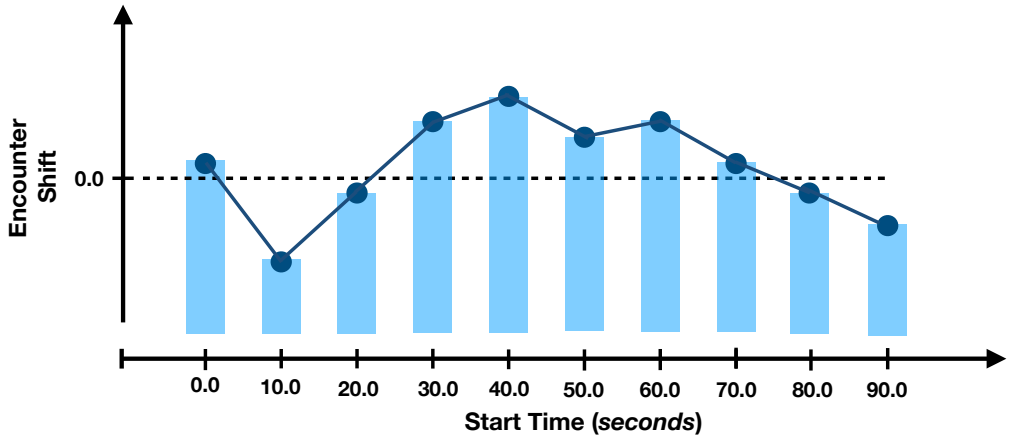

Figure 6. An example of encounter aware scheduling for a single flight-plan. The horizontal axis represents the potential start-time for a flight, while the vertical axis represents the encounter shift from that flight at the given start time

evenly in half by introducing a new waypoint, while a segment that is shorter is merged with neighboring segments by removing a waypoint. The heuristic is important for two primary reasons:

1) It enforces a bound on the number of waypoints in the trajectory
2) By expanding and contracting the segments between waypoints, they can converge to the length of the smallest collision free interval

Because this heuristic is used at each iteration of the optimization process, the optimization is not guaranteed to converge to a local minima. The heuristic modifies the parameter space of the objective if it either adds or removes a waypoint. To handle this problem, we split the optimization into two parts, with the first allowing the addition and removal of waypoints, and the second freezing the number of waypoints in the trajectory in place and proceeding with standard gradient optimization. We found this approach to work particularly well, because the first half of the optimization process allows the trajectory to converge to a reasonable number of waypoints given the obstacles in the configuration, while the second fine-tunes that representation of the trajectory with respect to the constraints and the objective. We leave convergence proofs of this approach for future work.

In our approach, we apply the penalty method as a way to enforce constraints [37]. The penalty method transforms a constrained optimization problem into an unconstrained problem by including the constraints in the objective. Using the l_1 exact penalty method the transformed objective takes on the following form:

$$\tilde{f}(\mathbf{x}) = f(\mathbf{x}) + \mu \sum_{i=1}^{n_{ineq}} |g_i(\mathbf{x})|^+ + \mu \sum_{i=1}^{n_{eq}} |h_i(\mathbf{x})| \quad (6)$$

where $|g_i(\mathbf{x})|^+ = \max(0, g_i(\mathbf{x}))$ is the penalty satisfying the inequality constraint, and $|h_i(\mathbf{x})|$ is the penalty satisfying the equality constraint. By formulating the constraints in this way, and picking a sufficiently large penalty coefficient μ, the optimization eventually drives all of the constraint violations to zero. The iterative approach is outlined in Algorithm 1.

Input:
T_{term}: terminating conditions
i_{waypt}: iteration to freeze number waypoints
α: initial step-size
$\tilde{f}$: optimization objective
$\mathbf{x}_0$: initial trajectory from a flight request
while not T_{term} **do**
  if $i < i_{waypt}$ **then**
    if not *collisionFree*($\mathbf{x}_i$) **then**
      $\mathbf{x}_i$ = expandWaypoints($\mathbf{x}_i$)
      $\mathbf{x}_i$ = contractWaypoints($\mathbf{x}_i$)
    end
  end
  $\mathbf{x}_{i+1} \leftarrow \mathbf{x}_i - \alpha_i \nabla \tilde{f}(\mathbf{x}_i)$
end

Algorithm 1: Gradient based trajectory optimization with waypoint expansion

We pick i_{waypt} such that it is equal to roughly half of the iterations that will be performed during the optimization.

As stated previously, by formulating the objective in this way and allowing waypoint expansion, we are able to optimize the trajectory directly in the 4D space while considering dynamic obstacles such as UAS and weather events. We are also able to compute the gradients directly by keeping both the constraints and the original objective as smoothly differentiable functions. This allows our planning algorithm to scale well to complex environments with many other co-existing UAS. In this work, we use a real-time constraint and an iteration constraint $i = 500$ as the termination conditions, with $i_{waypt} = 250$. The step size α is incrementally reduced at each iteration after i_{waypt} has been exceeded.

F. Encounter-Aware Flight Planning

In the previous section, we introduced a trajectory optimization approach that can efficiently perform deconfliction and scale to airspace with a large number of UAS. However, the algorithm only works to deconflict a trajectory against the existing trajectories and obstacles in the airspace, without considering capacity constraints and the overall impact on the airspace as a whole. To address this problem, we introduce a decomposition in our planning approach between the lower-level trajectory optimization and the high-level traffic management decision making such as schedule allocation, altitude assignment, and corridor control. We adopt an approach similar to the NASA SAFE50 architecture [38], and allow the scheduling of the flight to be independent of the trajectory generation. While scheduling and trajectory generation could be coupled we introduce this decomposition to allow our approaches to scale and to more easily integrate the encounter metrics introduced earlier into the flight-planning process. We defer the coupling of the trajectory-optimization and the higher-level traffic management for future work.

To introduce encounter metrics into the planning process, we focus on a single aspect of high-level traffic management - flight-scheduling. We consider a simple approach to scheduling, where we attempt to optimize the future take-off time of the UAS with regards to a quantity of interest. For example, if a flight request come in at time t_0, it may take-off anytime between t_0 and t_{max}, where t_{max} is the maximum delay that flight can incur. If the take-off time is optimized in regard to the total number of encounters with already planned flights, we can compute the total number of those encounters for hypothetical flights that would start in the time range $[t_0, t_{max}]$ and determine the optimal take-off time. We call this approach *First Come Scheduling*. The name was chosen, because the first vehicles to request plans in an airspace will not incur any delays, while plans at later times may be asked to delay in order to minimize the number of encounters with existing vehicles. For simplicity, we discretize the time range at fixed intervals Δt, and evaluate the immediate encounters for a flight request starting at each of the discrete start times.

We introduce the encounter shift metric into this scheduling approach by considering how the encounter shift changes as we change the take-off time of the flight request. We evaluate the encounter shift of a flight-request between the encounter distribution of the airspace at each point in the interval $[t_0, t_{max}]$, and pick a take-off time with lowest encounter shift value. The evaluation process is outlined in Figure 6 for a request on the time interval $[0, 90]$. In the example, the encounter shift is minimized at $t = 10\,\text{s}$, and we call this scheduling process the *Encounter Schedule*. In the remainder of this work we use $t = 10\,\text{s}$ and $t_{max} = 90\,\text{s}$ The trajectory optimization approach outlined earlier can be used following either of the scheduling approaches. Because the scheduling and the trajectory optimization are decoupled it is also possible to evaluate the effectiveness of the approaches individually.

While it is possible to minimize the encounter shift by introducing structure into the airspace, we leave that investigation for future work. Instead, we focus on using take-off scheduling that is encounter aware to minimize the encounter shift for each flight. In particular, one important aspect of minimizing the encounter shift during scheduling is that it helps address the problem of planning in many-vehicle encounters. By optimizing the take-off time for each flight with regards to the encounter shift, we naturally drive the airspace to have fewer many vehicle encounters. We demonstrate this property in detail in the Results Section.

G. Jointly Optimized Flight Planning

Jointly optimizing flight plans can be done with a centralized planner, that can process all flight requests within an airspace and perform multi-trajectory optimization. During this optimization process, the planner computes multiple optimized trajectories that are collision free simultaneously. Note that this does not require that the take-off times for the trajectories being optimized are the same only that the planning process is mutually shared for all of the flights. The formulation of the problem is nearly identical to the single trajectory optimization in Equation 5 except the optimization vector $\mathbf{x}$ now includes multiple trajectories. The complexity of optimizing multiple trajectories simultaneously grows quadratically with the number of encounters despite the optimization vector growing linearly, so the problem becomes more computationally expensive to solve.

It is important that the planner work in a realistic airspace simulation where flight requests can come in well before the flight or just a few moments prior to take-off. However, jointly optimizing all known flight requests is not feasible, particularly if the number of operations is high. In this work, we enforce a limit on the number of trajectories the planner optimizes simultaneously by defining a time horizon at which the joint optimization ends t_{opt}. The approach is as follows:

Input: t_{opt}
while $t < t_{opt}$ **do**
  if *new requests* **then**
    $\mathscr{F} \leftarrow$ flight requests in range $[t, t_{opt}]$
    $\mathscr{T} \leftarrow JointOpt(\mathscr{F})$
  end
  take off for all in $\mathscr{T}$ starting at t
  $t \leftarrow t + \Delta t$
end

Algorithm 2: Rolling horizon joint trajectory optimization

After the planner has gone through a single optimization cycle, the process can start with a new t_{opt}. In this work, we use $\Delta t = 1\,\mathrm{s}$. While it is possible to efficiently perform coordinated planning even in the presence of asynchronous schedules [31], we do not explore this route in this work. We assume a centralized planner that can process flight requests simultaneously for when we perform joint optimization of multiple trajectories.

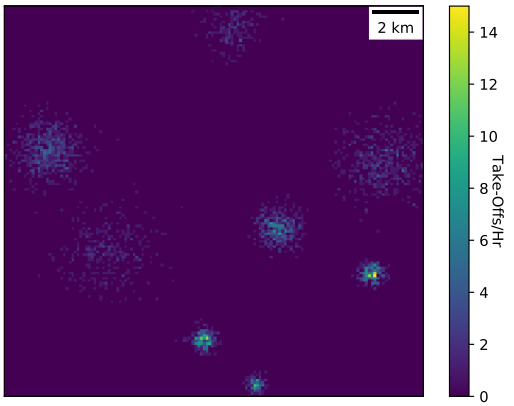

Figure 7. Take-off heat-map for the multi-modal region used to evaluate the approaches in this work.

Results

The scalability and effectiveness of the encounter aware planning approach is demonstrated through hundreds of hours of airspace simulation. We compare the approach to stand alone trajectory optimization, two schedule based approaches for deconfliction, and a joint optimization planning approach. We examine the performance of each in a high-fidelity airspace simulation.

H. Simulation Model

We use a high-fidelity airspace simulator to evaluate our approach. In the simulation, flight-requests are generated using a stochastic process and are based on a demand model. The demand model used in this work is multi-modal from a geographic perspective. For a map of take-off heat-map of a 5 hour simulation run, see Figure 7. The region is $18\,\mathrm{km} \times 18\,\mathrm{km}$ in size. The requests are passed onto a planner which can be configured to follow any of the management approaches and their combination discussed in the previous section. Planning is performed in a distributed way except for joint optimization.

To model the vehicles in our simulation, we use a simple point-particle dynamic model with a hybrid PID and logic control for guidance. The prescribed cruise speed and cruise altitudes of the vehicles are $40\,\mathrm{m\,s^{-1}}$ and $120\,\mathrm{m}$ respectively. While the simulated vehicles have on board sensing and conflict resolution

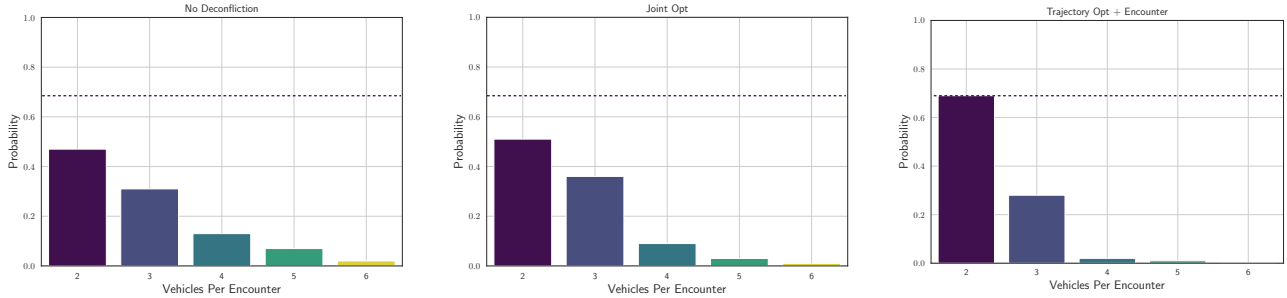

Figure 8. The encounter distributions with 1000 take-offs per hour for no deconfliction (left), joint optimization (center), and encounter aware trajectory optimization (right). The horizontal line represents the fraction of pairwise encounters in the the encounter aware trajectory optimization simulations.

capabilities we do not consider them in this work, and instead focus on the evaluation of the planning approaches proposed.

I. Airspace Simulations

The goal of the airspace simulations is to evaluate the viability of the approaches presented earlier in the context of strategic pre-flight deconfliction. We consider point to point operations with take-off rates ranging from 100 to 1000 per hour. The following approaches are evaluated in this section:

- **Trajectory Optimization**: standalone trajectory optimization approach. Referred to as Traj Opt in the evaluations.
- **First Come Scheduler**: a scheduler that attempts to minimize the number of encounters for a flight request.
- **Encounter Scheduler**: a scheduler that attempts to minimize the encounter shift for a flight request.
- **Encounter Aware Trajectory Optimization**: combined encounter scheduler and trajectory optimization. Referred to as Traj Opt + Encounter.
- **Joint Optimization**: jointly optimized planning approach using a centralized planner. Referred to as Joint Opt.

For each approach, we collected approximately 28 hours of simulated operational time, resulting in 95% confidence intervals for the results at the lowest take-off rate of 100 per hour, and 99% confidence interval at 1000 take-offs per hour.

Qualitatively, the differences in encounter distributions between encounter aware flight planning and the joint optimization approaches are shown in Figure 8 for 1000 take-off/hr simulations. The encounter distribution for an airspace managed by the encounter aware planner has a significantly larger fraction of pairwise encounters compared to encounter with more than two vehicles. This is in contrast to the encounter distribution from simulations with no deconfliction and the encounter distribution with joint optimization, where the encounters are more evenly distributed. A key property of the encounter aware planner is that it drives the airspace into a less complex regime by shifting the encounter distribution to simpler encounter sets. This has a significant impact on the overall safety of the airspace (see Figure 9). We note that the encounter aware trajectory optimization planner outperforms all the planning approaches including the one using joint optimization at 1000 take-offs/hr. This is a particularly surprising result, as we expect joint optimization to be the superior approach because it is centralized. However, we suspect that at the higher take-off/hr rates, we are reaching the limit of how well a centralized method can perform and thus see a worsening performance compared to the encounter aware method. Route efficiencies are shown in Figure 10 as ratios of actual over nominal route lengths. Deconfliction approaches that use only scheduling have an efficiency of 1.0, and are not shown. Overall, we see that joint optimization is the most efficient approach, with the encouter aware planning performing nearly as well.

We also examine the resulting encounter characteristics for each method of the airspace in question in Table I. The table shows the encounter shift for each method at different take-off/hr rate as well as the percentage of the encounter which have more than two vehicles in them. Both of these metrics are

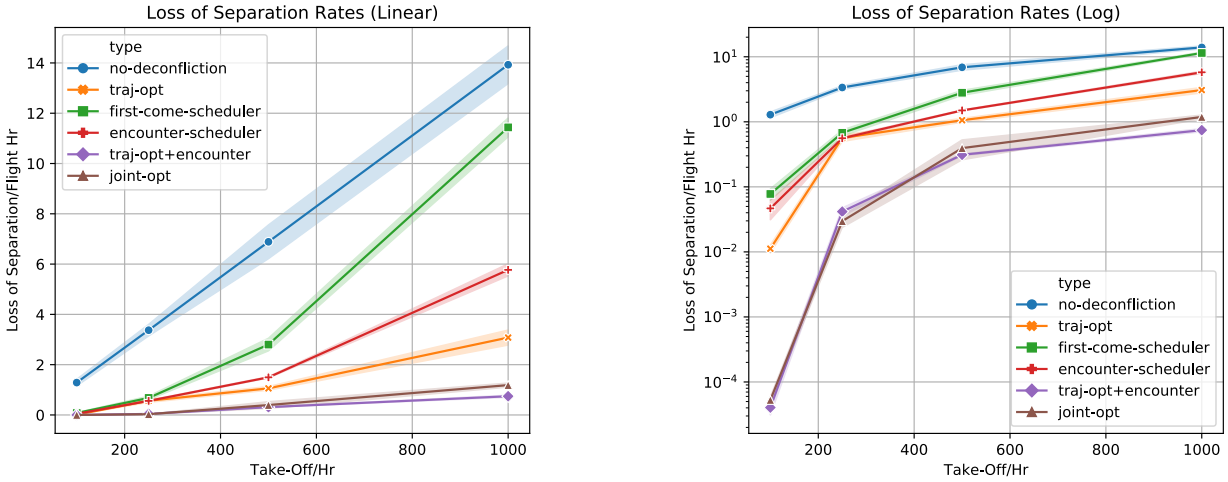

Figure 9. Loss of separation rates for the multi-modal region with linear (left) and log (right) scaled axis. The horizontal axis represents the vehicle take-off rate per hour for the whole region.

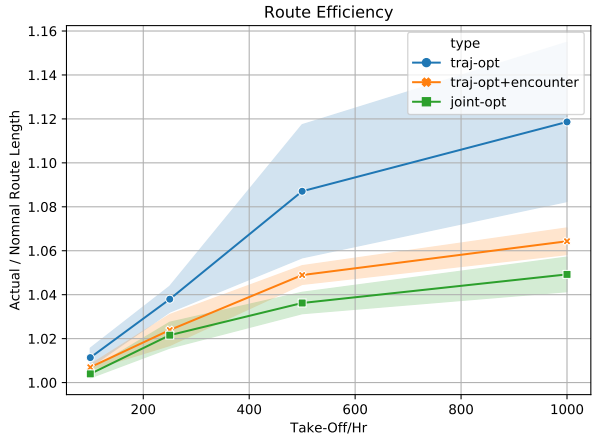

Figure 10. Route efficiencies for the multi-modal region at varying take-off rates. Efficiencies are represented by the ratio of actual / nominal route lengths.

important in assessing airspace complexity. Here, we are able to observe the primary difference between the joint optimization and the encounter aware methods. Because the joint optimization method is not attempting to manage the encounters in the airspace, we see that it actually has a positive encounter shift at 1000 take-offs/hr. This implies that the airspace managed by the joint optimization planner has more complex encounters than the one with no deconfliction at all. It should also be noted that the encounter aware trajectory optimization planner has the lowest encounter shift, signifying it is able to drive the encounter distribution towards simple encounter scenarios. This would explain the gap in safety performance between the encounter aware and the joint optimization approaches.

Conclusion

A robust and efficient distributed planning method is presented in this paper. The approach combines 4D trajectory optimization and an encounter aware scheduling heuristic, and is shown to outperform existing methods for distributed flight-planning. As part of this work, we also provide a number of numerical benchmarks that compare our approach to a other distributed planning algorithms and a centralized joint optimization approach in a high-fidelity airspace simulation environment. We show that our approach is able to consistently perform more safely and efficiently than other distributed approaches at all the operational densities considered in this work. We show that by reducing the fraction of many-vehicle encounters in the airspace, our approach is not only able to achieve better safety performance but is also more efficient than traditional planning approaches.

While this work provides a starting point, a number of questions still remain about traffic management in the unmanned sky. In particular, one big challenge lies in understanding the requirements of UTM. It is not clear how demand for UAS will evolve in the next 20 years, making it difficult to create requirements

Metric	CAS	Take-off Rates (1 / hr)			
		100	250	500	1000
Encounter Shift	No Deconfliction (ref)	-	-	-	-
	Traj Opt	-0.02 ± 0.01	-0.04 ± 0.03	-0.08 ± 0.03	-0.14 ± 0.05
	Encounter Scheduler	-0.10 ± 0.02	-0.22 ± 0.08	-0.47 ± 0.16	-0.84 ± 0.11
	Encounter Traj Opt	**-0.11 ± 0.01**	**-0.29 ± 0.09**	**-0.63 ± 0.21**	**-1.02 ± 0.14**
	Joint Opt	0.01 ± 0.02	-0.12 ± 0.04	-0.02 ± 0.05	0.13 ± 0.04
Multi-threat Encounter Percent	No Deconfliction	5.6 ± 1.3	19.5 ± 2.1	31.8 ± 2.5	51.4 ± 3.3
	Traj Opt	4.8 ± 1.5	17.1 ± 3.7	29.3 ± 3.2	54.5 ± 3.9
	Encounter Scheduler	**0.1 ± 0.1**	**4.3 ± 1.7**	**21.9 ± 5.4**	**36.8 ± 3.2**
	Encounter Traj Opt	0.2 ± 0.1	4.9 ± 1.2	23.6 ± 2.9	41.4 ± 2.8
	Joint Opt	5.3 ± 0.9	20.8 ± 2.7	34.6 ± 3.2	58.9 ± 3.8

Table I. Encounter metrics for varying take-off rates and different planning algorithms

and a functional system that can manage unmanned traffic, and future work will examine how flexible our approach is to a wide range of requirements. We also plan to examine other methods beyond the penalty method trajectory optimization [37], as it is known to have difficulties in convergence which could impact the overall quality of the flight-plans it produces. Another important aspect will be in evaluating how well the flight-planning approaches outlined in this work can integrate into the current airspace. Of particular importance is the ability to interface with human decision makers in ATC [39]. Lastly, we plan to examine other approaches to inferring the encounter shift. Such as the ones that have been applied to predict delays in air traffic networks [40]. One thing is certain. To enable safe and efficient operation of UAS, a traffic management system is needed that is robust, reliable, and scalable. The work presented in this paper is a small step towards that goal.

Acknowledgements

The authors would like to thank Richard Golding and Karthik Balakrishnan for their insights and guidance throughout this work. A special thanks also goes out to everyone at Airbus UTM, who in one way or another helped contribute to this paper.

References

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

[2] D. Jenkins, B. Vasigh, C. Oster, and T. Larsen, *Forecast of the Commercial UAS Package Delivery Market*. Embry-Riddle Aeronautical University, 2017.

[3] J. Holden and N. Goel, "Uber elevate: Fast-forwarding to a future of on-demand urban air transportation," *Uber Technologies, Inc., San Francisco, CA*, 2016.

[4] B. A. Hamilton. (2018). Urban air mobility market study, [Online]. Available: https : / / go . nasa . gov / 2MVSbth.

[5] P. Kopardekar, J. Rios, T. Prevot, M. Johnson, J. Jung, and J Robinson, "Unmanned aircraft system traffic management (utm) concept of operations," in *AIAA Aviation Forum*, 2016.

[6] H. Ushijima. (2017). Utm project in japan, [Online]. Available: https://bit.ly/2UMzDyw.

[7] SESAR. (2017). U-space blueprint, [Online]. Available: https://bit.ly/2UKf8SX.

[8] L. Janson, E. Schmerling, A. Clark, and M. Pavone, "Fast marching tree: A fast marching sampling-based method for optimal motion planning in many dimensions," *The International journal of robotics research*, vol. 34, no. 7, pp. 883–921, 2015.

[9] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on intelligent vehicles*, vol. 1, no. 1, pp. 33–55, 2016.

[10] Y.-b. Chen, G.-c. Luo, Y.-s. Mei, J.-q. Yu, and X.-l. Su, "Uav path planning using artificial potential field method updated by optimal control theory," *International Journal of Systems Science*, vol. 47, no. 6, pp. 1407–1420, 2016.

[11] C. Tomlin, G. J. Pappas, and S. Sastry, "Conflict resolution for air traffic management: A study in multiagent hybrid systems," *IEEE Transactions on automatic control*, vol. 43, no. 4, pp. 509–521, 1998.
[12] A. Chakrabarty, V. Stepanyan, K. S. Krishnakumar, and C. A. Ippolito, "Real-time path planning for multi-copters flying in utm-tcl4," in *AIAA Scitech 2019 Forum*, 2019, p. 0958.
[13] S. Balachandran, A. Narkawicz, C. Muñoz, and M. Consiglio, "A path planning algorithm to enable well-clear low altitude uas operation beyond visual line of sight," in *Twelfth USA/Europe Air Traffic Management Research and Development Seminar (ATM2017)*, 2017.
[14] J. Lee, H. Shin, and D. H. Shim, "Path-planning with collision avoidance for operating multiple unmanned aerial vehicles in same airspace," in *2018 AIAA Information Systems-AIAA Infotech@ Aerospace*, 2018, p. 1460.
[15] Z. Liu and R. Sengupta, "An energy-based flight planning system for unmanned traffic management," in *Systems Conference (SysCon), 2017 Annual IEEE International*, IEEE, 2017, pp. 1–7.
[16] J. Rios. (2018). Strategic deconfliction: System requirements, [Online]. Available: https://utm.arc.nasa.gov/docs/2018-UTM-Strategic-Deconfliction-Final-Report.pdf.
[17] J. T. Betts, "Survey of numerical methods for trajectory optimization," *Journal of guidance, control, and dynamics*, vol. 21, no. 2, pp. 193–207, 1998.
[18] N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa, "Chomp: Gradient optimization techniques for efficient motion planning," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, IEEE, 2009, pp. 489–494.
[19] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal, "Stomp: Stochastic trajectory optimization for motion planning," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, IEEE, 2011, pp. 4569–4574.
[20] J. Schulman, J. Ho, A. X. Lee, I. Awwal, H. Bradlow, and P. Abbeel, "Finding locally optimal, collision-free trajectories with sequential convex optimization.," in *Robotics: science and systems*, Citeseer, vol. 9, 2013, pp. 1–10.
[21] A. Klein, M. D. Rodgers, and K. Leiden, "Simplified dynamic density: A metric for dynamic airspace configuration and nextgen analysis," in *Digital Avionics Systems Conference, 2009. DASC'09. IEEE/AIAA 28th*, IEEE, 2009, pp. 2–D.
[22] P. Kopardekar, A. Schwartz, S. Magyarits, and J. Rhodes, "Airspace complexity measurement: An air traffic control simulation analysis," in *7th USA/Europe Air Traffic Management R&D Seminar, Barcelona, Spain*, 2007.
[23] G. Sharon, R. Stern, M. Goldenberg, and A. Felner, "The increasing cost tree search for optimal multi-agent pathfinding," *Artificial Intelligence*, vol. 195, pp. 470–495, 2013.
[24] K.-H. C. Wang and A. Botea, "Mapp: A scalable multi-agent path planning algorithm with tractability and completeness guarantees," *Journal of Artificial Intelligence Research*, vol. 42, pp. 55–90, 2011.
[25] R. Luna and K. E. Bekris, "Push and swap: Fast cooperative path-finding with completeness guarantees," in *IJCAI*, 2011, pp. 294–300.
[26] I. Rish *et al.*, "An empirical study of the naive bayes classifier," in *IJCAI 2001 workshop on empirical methods in artificial intelligence*, vol. 3, 2001, pp. 41–46.
[27] L. Ren, M. Castillo-Effen, H. Yu, Y. Yoon, T. Nakamura, E. N. Johnson, and C. A. Ippolito, "Small unmanned aircraft system (suas) trajectory modeling in support of uas traffic management (utm)," in *17th AIAA Aviation Technology, Integration, and Operations Conference*, 2017, p. 4268.
[28] P. Sachs. (2018). A quantitative framework for uav risk assessment, [Online]. Available: http://bit.ly/altiscopetr008.
[29] L. Xiao and S. Boyd, "Optimal scaling of a gradient method for distributed resource allocation," *Journal of optimization theory and applications*, vol. 129, no. 3, pp. 469–488, 2006.
[30] C. A. Uribe, S. Lee, A. Gasnikov, and A. Nedić, "Optimal algorithms for distributed optimization," *arXiv preprint arXiv:1712.00232*, 2017.
[31] E. Robinson, H. Balakrishnan, M. Abramson, and S. Kolitz, "Optimized stochastic coordinated planning of asynchronous air and space assets," *Journal of Aerospace Information Systems*, pp. 10–25, 2017.
[32] R. Golding. (2018). Metrics to characterize dense airspace traffic, [Online]. Available: http://bit.ly/altiscopetr004 (visited on 02/03/2019).
[33] M. Rufli, D. Ferguson, and R. Siegwart, "Smooth path planning in constrained environments," in *IEEE*

International Conference on Robotics and Automation, 2009. ICRA'09, Eidgenössische Technische Hochschule Zürich, 2009, pp. 3780–3785.
[34] J. Van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, IEEE, 2008, pp. 1928–1935.
[35] D. Silver, "Cooperative pathfinding.," *AIIDE*, vol. 1, pp. 117–122, 2005.
[36] M. Phillips and M. Likhachev, "Sipp: Safe interval path planning for dynamic environments," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, IEEE, 2011, pp. 5628–5635.
[37] E. Polak, *Optimization: algorithms and consistent approximations*. Springer Science & Business Media, 2012, vol. 124.
[38] C. Ippolito, K Krishnakumar, V Stepanyan, A Chakrabarty, and J Baculi, "An autonomy architecture for high-density operations of small uas in low-altitude urban environments," in *2019 AIAA Modeling and Simulation Technologies Conference. San Diego, CA. Jan*, vol. 2109, 2019.
[39] P. D. Vascik, H. Balakrishnan, and R. J. Hansman, "Assessment of air traffic control for urban air mobility and unmanned systems," 2018.
[40] K. Gopalakrishnan and H. Balakrishnan, "A comparative analysis of models for predicting delays in air traffic networks," ATM Seminar, 2017.