Rolling Horizon with K-Position Search Method for Strategic Deconfliction of Package Delivery UAS

Priyank Pradeep [*], Gautam Sai Yarramreddy [†], and Nicholas Amirsoleimani [‡]
Universities Space Research Association, Moffett Field, CA, 94035, USA.

Alexey A. Munishkin [§], Robert A. Morris [¶], Krishna M. Kalyanam [‖], and Min Xue [**]
NASA Ames Research Center, Moffett Field, CA, 94035, USA.

Kenny Chour [††]
Metis Technology Solutions, Moffett Field, CA, 94035, USA.

In this research, the strategic deconfliction of unmanned aircraft systems for an urban package delivery environment with two depots and multiple drop-off locations is studied. This research aims to formulate a mathematical model to compute both the departure sequence and scheduled time of departure for each unmanned aircraft system at a depot, considering temporal constraints at en-route crossing waypoints and depots for strategic deconfliction. However, the problem formulation results in an NP-hard mixed-integer nonlinear programming problem for the global optimal solution, so instead, a "rolling horizon with k-position search" heuristic method is developed. The simulation studies show that an increase in the value of k (the parameter used to determine the size of the local neighborhood) reduces the average ground delay at the cost of an increase in the computation time for a given problem size. The study also shows an order of magnitude increase in the maximum number of flights scheduled with the integration of rolling horizon (time decomposition) compared to those without the integration of rolling horizon in the heuristic algorithm for a given computation time cut off.

I. Introduction

Small unmanned aircraft systems (UAS) are envisioned to provide socio-economic benefits to the public by revolutionizing operations related to package delivery, precision agriculture scouting, surveillance, supporting first responders, and inspection of critical infrastructure like railroads and bridges [1, 2]. These UAS are anticipated to operate in the following manner: i) much closer to each other (higher traffic density) than conventional aircraft, ii) exclusively in low-altitude airspace, i.e., less than 400 ft above ground level (AGL) [2], and iii) beyond visual line of sight (BVLOS) in the National Airspace System (NAS) [3–5]. Therefore, UAS traffic management (UTM) aims to create a system that can safely and efficiently integrate low-altitude airspace UAS BVLOS operations safely and efficiently into NAS.

A. Background

1. Conflict Management Model

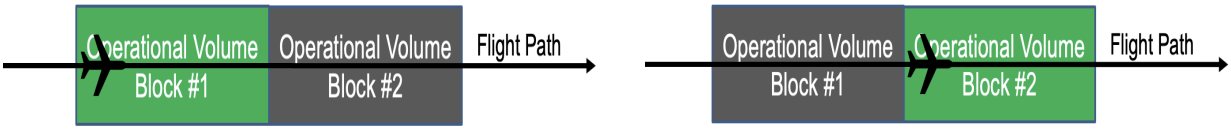
UAS traffic management (UTM) has been envisioned to have multiple layers of a conflict management model to ensure the safe, efficient, and scalable operations of UAS. These layers of the conflict management model are strategic deconfliction, tactical separation assurance, and collision avoidance [4]. At each layer, conflicts are resolved through a series of maneuvers compatible with the operational environment. The objective of the first layer of the conflict

[*]Aerospace Engineer, Universities Space Research Association, NASA Ames Research Center, AIAA Senior Member.
[†]Research Intern, Universities Space Research Association, NASA Ames Research Center.
[‡]Research Intern, Universities Space Research Association, NASA Ames Research Center.
[§]Aerospace Engineer, Aviation Systems Division, NASA Ames Research Center, AIAA Member.
[¶]Senior Research Scientist, Intelligent Systems Division, NASA Ames Research Center.
[‖]Aerospace Research Engineer, Aviation Systems Division, NASA Ames Research Center, AIAA Associate Fellow.
[**]Aerospace Research Engineer, Aviation Systems Division, NASA Ames Research Center, AIAA Senior Member.
[††]Senior Software Engineer, Metis Technology Solutions, NASA Ames Research Center, AIAA Member.

management model, i.e., strategic deconfliction, is to i) minimize the likelihood of airborne conflicts between UAS operations and ii) maximize the airspace usage by adjusting the departure times of UAS [3, 4]. The strategic deconfliction may involve re-planning routes in some scenarios. The tactical separation assurance layer consists of executing one or more maneuvers (speed change, altitude change, and path-stretch) to avoid an airborne conflict promptly when strategic deconfliction is ineffective due to uncertainties [6–8]. Finally, the last layer of protection is the onboard detect and avoid (DAA) system, i.e., collision avoidance system [3].

2. Trajectory-Based Operational Intent and Operational Volume Blocks

In the UTM ecosystem, a UAS operator planning to fly BVLOS is required to share the trajectory-based operational intent with other UAS operators/airspace users via the UAS Supplier Service (USS) network [3, 4]. As shown in Figure 1, the trajectory-based operational intent includes a sequence of 4D (spatiotemporal) operational volume blocks (OVBs) that make up the intended flight profile [3, 9, 10]. An OVB gets activated when a UAS enters it and deactivated when it exits as shown in Figure 1 [3]. In this research, each OVB is assumed to be fixed in space and has specified entry and exit times for the UAS of an operator per NASA and the FAA's UTM concept of operations [3, 4, 9]. In the operation planning phase, prior to departure, the UAS operator or operator's USS checks the OVBs against other operations for any 4D conflicts. If any spatiotemporal overlapping of OVBs is detected, then negotiation and replanning of the operational intent of the UAS are performed [3, 11]. Whenever there is a 4D spatiotemporal overlapping of two operational intents of different UAS operators, for example, at a crossing waypoint, then deconfliction of overlapping OVBs can be performed via temporal separation at that waypoint [2, 3].



(a) Lateral view of active operational volume block # 1 **(b) Lateral view of active operational volume block # 2**

Fig. 1 Activation (Green Color) and Deactivation (Grey Color) of Operational Volume Block (OVB) as a UAS Enters and Exits [5]

3. Departure Sequencing and Scheduling

The departure scheduling problem is to find a sequence of aircraft departure times that optimize an objective such as average ground delay or aggregate ground delay of all aircraft or makespan (scheduled time of departure of the last aircraft) [12, 13]. Mixed-integer linear programming (MILP) and mixed-integer nonlinear programming (MINLP) are powerful tools to model and solve combinatorial optimization problems such as departure sequencing and scheduling. MILP and MINLP problems cannot be solved deterministically in polynomial time (NP-hard) [14], so the computation time for finding an exact solution is exponential in the number of UAS. Therefore, heuristic methods for MILP/MINLP are of high interest for real-time solutions in conventional air traffic management (ATM) and traffic management of new entrants like UTM [12, 13, 15, 16].

4. Local Neighborhood Search

Local neighborhood search is a heuristic algorithm for local-optimal real-time solutions. Unlike global search methods that explore the entire solution space, local neighborhood search algorithms focus on making incremental changes to improve a current solution until they reach a satisfactory solution [12]. This approach is practical when the vast solution space makes an exhaustive search impractical in real-time [13].

5. Rolling Horizon

The rolling horizon scheduling approach involves decomposing the MINLP problem into multiple time slots, which are solved sequentially as sub-problems [17]. Figure 2 illustrates the concept of rolling horizon [17]. The rolling horizon mechanism delivers the scheduling plan periodically for every planning horizon. As shown in Figure 2, the time horizon of prediction (T) is defined as the time period from the beginning of the planning horizon to the end of the current UAS

traffic prediction. In each planning horizon (Figure 2), the scheduled time of departure (STD) of UAS with the earliest time of departure (ETD) in the current planning horizon are computed considering predicted conflicts at depots and en-route crossing waypoints due to UAS from current and previous planning horizons. Once the STDs are computed for the current planning horizon, they are frozen [18]. Then, STDs are calculated for UAS that have ETDs falling in the next planning horizon. The rolling horizon process is repeated until the sequencing and scheduling for overall traffic (multi-stage optimization problem) is computed [17].

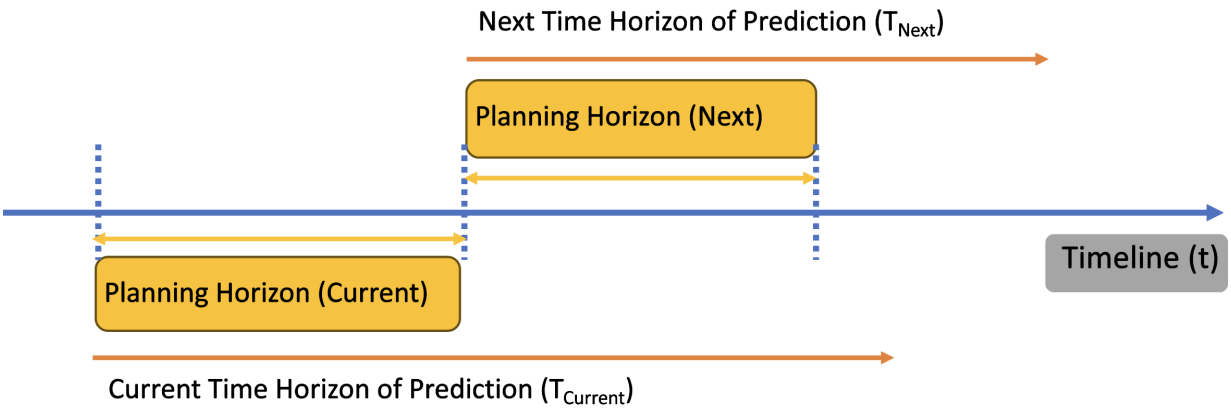


Fig. 2 Illustration of Rolling Horizon Concept for Strategic Deconfliction

B. Motivation

The current research aims to extend strategic deconfliction using OVBs at a single crossing waypoint [5] to multiple crossing waypoints. In conventional ATM research, the k-Position search (KPS), also known as insertion and local search (ILS), and constrained position shift (CPS) have been applied for departure and arrival sequencing and scheduling problems [12, 13]. Given the small flight range and flight time for UAS package delivery operations (in minutes) compared to conventional ATM (in hours), the KPS can be extended for en-route strategic deconfliction along with departure sequencing and scheduling of UAS operations. In this research, the simulated network structure of routes with depots and corresponding drop-off locations, are considered hub-and-spoke type [3, 19, 20], therefore, arrival sequencing and scheduling has been ignored. Since the departure sequencing and scheduling problem is non-deterministic polynomial-time (NP) hard, one way of overcoming scalability issues is to break down the departure sequencing and scheduling timeline at depots into subproblems using the concept of the rolling horizon [17, 18] and then apply KPS to each rolling horizon. Therefore, this research aims to understand the scalability of rolling horizon with the KPS heuristic algorithm for departure sequencing and scheduling with en-route strategic deconfliction in the UTM package delivery environment.

II. Problem Formulation

The strategic deconfliction of UAS at an en-route crossing waypoint, depot A (departure port), and depot B (departure port) has been formulated as a MINLP problem. The minimum temporal separation between UAS of different operators at an en-route crossing waypoint has been imposed based on temporal separation equations derived in [5] for strategic deconfliction. The strategic deconfliction at departure ports (depots) and en-route crossing waypoints is performed by adding ground delays (GDs) to the STDs of UAS. Assuming the flight time (FT) of a UAS to various crossing waypoints from a depot remains constant. Therefore, for each scenario, adjustment to the scheduled times of arrival (STAs) at various crossing waypoints of a UAS is performed by adjusting the STD. The variables and constants used in the MINLP problem are shown in Table 1.

The departure sequence, STDs, and STAs to various waypoints of UAS from all previous planning horizons are treated as constant while in the current planning horizon for departure sequencing and scheduling with en-route strategic deconfliction. Therefore, departure sequencing and scheduling are computed only for UAS in the current planning horizon (Figure 2).

In this section, for ease of explanation, the variables of interest are defined for the i^{th} UAS. Still, without loss of generality, similar formulas exist for the j^{th} and other UAS.

Table 1 Definition of Variables and Constants Used in the MINLP Problem

Terminology	Definition	Characteristic	Unit
GD	Ground delay of a UAS	Objective variable	Seconds
n	Total number of flights scheduled to depart from a depot in a planning horizon	Constant	Number of Flights
ODT	On-demand service request time to deliver a package	Poisson distribution	Seconds
Δt_{prep}	Flight preparation time	Constant	Seconds
Δt_{sep}	Minimum departure time separation	Constant	Seconds
Δt_{CW}	Minimum temporal separation at the crossing waypoint	Constant	Seconds
ETD	Earliest time of departure of the UAS considering ODT and flight preparation time	Poisson distribution	Seconds
STD	Scheduled time of departure	Decision variable	Seconds
FT	Flight time of a UAS to travel from the depot to the crossing waypoint	Constant	Seconds
STA	Scheduled time of arrival of a UAS to the crossing waypoint	Function of STD and FT	Seconds
$\omega(i, j)$	Decides which UAS will sequence the waypoint first: i^{th} UAS or j^{th} UAS	Binary decision variable	N/A

1. Objective Function

In this research, the objective function of the MINLP problem is the average ground delay defined as follows [5]:

$$\text{Minimize} \quad \frac{\sum_{i=1}^{n_A} \text{GD(i)} + \sum_{j=1}^{n_B} \text{GD(j)}}{n_A + n_B} \tag{1}$$

where GD(i) and GD(j) are the ground delays of the i^{th} UAS of the operator A, and j^{th} UAS of the operator B, respectively. Here, n_A and n_B are the total number of flights scheduled to depart from depot A and depot B in the

current planning horizon, respectively. The average ground delay is computed and minimized in each planning horizon.

The ground delay of the i^{th} UAS is defined as the difference between its STD and the earliest time of departure (ETD) as follows:

$$\text{GD(i)} = \text{STD(i)} - \text{ETD(i)} \tag{2}$$

ETD(i) of the i^{th} UAS is the sum of on-demand service request time for a package delivery (ODT(i)), and the flight preparation time (Δt_{prep}) [5]:

$$\text{ETD(i)} = \text{ODT(i)} + \Delta t_{\text{prep}} \tag{3}$$

Δt_{prep} is the time required for preparing the UAS flight, which includes time for on-demand order processing, packaging of order, loading of delivery package to the assigned UAS, and pre-flight checks. It is assumed to be a constant 3600 seconds.

2. Temporal Constraints at Departure Depots

The ODT in Equation 3 is simulated using a Poisson distribution [5]. The STD of each UAS is separated from its ODT by at least Δt_{prep} at each departure port [5]:

$$\text{STD(i)} \geq \text{EDT(i)} \tag{4}$$

At the departure port of UAS operator A and UAS operator B, STD of two consecutive UAS are separated by at least the minimum departure time separation (Δt_{sep}) [5]:

$$\text{STD(i+1)} \geq \text{STD(i)} + \Delta t_{\text{sep}} \tag{5}$$

3. Temporal Constraints at En-Route Crossing Waypoints

The minimum temporal separation (Δt_{CW}) constraint between two UAS of different UAS operators at the crossing waypoint is as follows [21]:

$$(\text{STA(j)} - \text{STA(i)})\omega(i,j) + (\text{STA(i)} - \text{STA(j)})(1 - \omega(i,j)) \geq \Delta t_{\text{CW}} \tag{6}$$

$$\omega(i,j) = \begin{cases} 1, & \text{if } i^{th} \text{ UAS sequences before } j^{th} \text{ UAS,} \\ 0, & \text{if } j^{th} \text{ UAS sequences before } i^{th} \text{ UAS,} \end{cases} \tag{7}$$

The STA of the i^{th} UAS to the crossing waypoint is given by:

$$\text{STA(i)} = \text{STD(i)} + \text{FT(i)} \tag{8}$$

where FT(i) is the flight time of the i^{th} UAS to the crossing waypoint.

The minimum temporal separation (between two flights) at an en-route crossing waypoint to avoid overlapping of active OVBs is defined based on the length and width of the OVBs, incoming crossing angle, and groundspeed of UAS [5]. The minimum temporal separation is shown in Equation 9, Equation 10, and Table 2. A UAS operator is assumed to not have real-time position and velocity information of a UAS of another UAS operator. However, the UAS operator would have real-time position and activation and deactivation times information of OVBs of another UAS operator. As stated earlier, STAs to various waypoints of UAS already airborne and planned in one of the previous planning horizons remains the same (Figure 2).

- For an incoming crossing angle > 90 deg, the minimum temporal separation (Δt_{CW}) at the crossing angle is defined using equations from [5] as follows:

$$\Delta t_{\text{CW}} = \max\left(\frac{2l_B}{V_B} + \frac{W_B}{2V_A \sin\theta} - \frac{W_A \cos\theta}{2V_A \sin\theta},\ \frac{2l_B}{V_B} + \frac{W_A}{2V_A \tan\theta_A}\right) \tag{9}$$

- For an incoming crossing angle ≤ 90 deg, the minimum temporal separation (Δt_{CW}) at the crossing angle is defined using equations from [5] as follows:

$$\Delta t_{\text{CW}} = \max\left(\frac{2l_B}{V_B} + \frac{W_B \sin\theta}{2V_A},\ \frac{l_B}{V_B} + \frac{l_A}{V_A} - \frac{W_A}{2V_A \tan\theta_A}\right) \tag{10}$$

Table 2 Definition of Terms Used in Temporal Separation Equations [5]

Terminology	Definition
V_A, V_B	Groundspeed of the UAS of operator A or B (assumed to be constant).
l_A, l_B	Length of the active operational volume block of UAS operator A or B.
W_A, W_B	Width of the active operational volume block of UAS operator A or B.
θ	Incoming crossing angle at the waypoint.

The optimization problem discussed earlier has a linear objective function (1) and linear constraints (4)-(5) except for minimum temporal separation constraints (6) at crossing waypoints. In this research, the Big-M method [22] has been utilized on the minimum temporal separation constraint at crossing waypoints (Equation 6) to convert the quadratic constraint into two linear constraints as shown in the following equations. One for if ith UAS sequences before jth UAS:

$$(\text{STA(j)} - \text{STA(i)}) + M(1 - \omega(i, j)) \geq \Delta t_{\text{CW}} \tag{11}$$

and the other for if jth UAS sequences before ith UAS:

$$(\text{STA(i)} - \text{STA(j)}) + M\omega(i, j) \geq \Delta t_{\text{CW}} \tag{12}$$

where in both equations, i.e., Equation 11 and Equation 12 have the Big-M value, ($M \gg 0$) which is a large positive constant. Ideally the Big-M value approaches infinity but one usually has to employ a trial-and-error methodology, i.e. randomly pick values of $M \gg 0$ and run an optimizer to see results then repeat as needed to find a suitable value. Now with the introduction of the Big-M value, the problem has been formulated as a MILP problem.

III. Rolling Horizon with k-Position Search for Departure Sequencing and Scheduling with En-Route Strategic Deconfliction

Given the difficulty of finding the exact solution to the NP-hard MINLP problem, a heuristic algorithm called rolling horizon with K-Position Search (KPS) developed in this research is shown in Figure 3 and Figure 4. The rolling horizon methodology decomposes the MINLP problem into multiple time slots, which are solved sequentially as sub-problems [17, 18]. The KPS algorithm developed in this research is based on the ILS algorithm described for single runway scheduling of commercial air traffic by Malik and Jung [12, 16]. As shown in Figure 3 and Figure 4, in a given planning horizon of the rolling horizon methodology [17], the KPS starts with the initial guess for the UAS departure sequence using the first-come-first-served (FCFS) order based on the ETDs. The term "position" means the position of a UAS in the departure sequence.

1. K-Position Search at Single Depot

As shown in Figure 3, an instance of a list of UAS with package delivery request is processed using the concept of rolling horizon [17], where the problem of computing the departure sequencing and scheduling of the instance of a list of UAS is decomposed into multiple subproblems based on ETDs of UAS as discussed earlier. At each planning horizon (or subproblem), as shown in Figure 4, the sequencing and scheduling of UAS with ETDs falling in the current planning horizon are computed using KPS [12]. In a given planning horizon, applying the KPS method at a depot involves the following steps:

- Fixing the i^{th} position in the departure sequence involves a local neighborhood search starting from the i^{th} position to the $(i + k - 1)^{th}$ position. For example, as shown in Figure 5, fixing the 1^{st} position in the departure sequence involves a local neighborhood search starting from the 1^{st} position to the 3^{rd} position, i.e., $\{A_1, A_2, A_3\}$ when k=3.

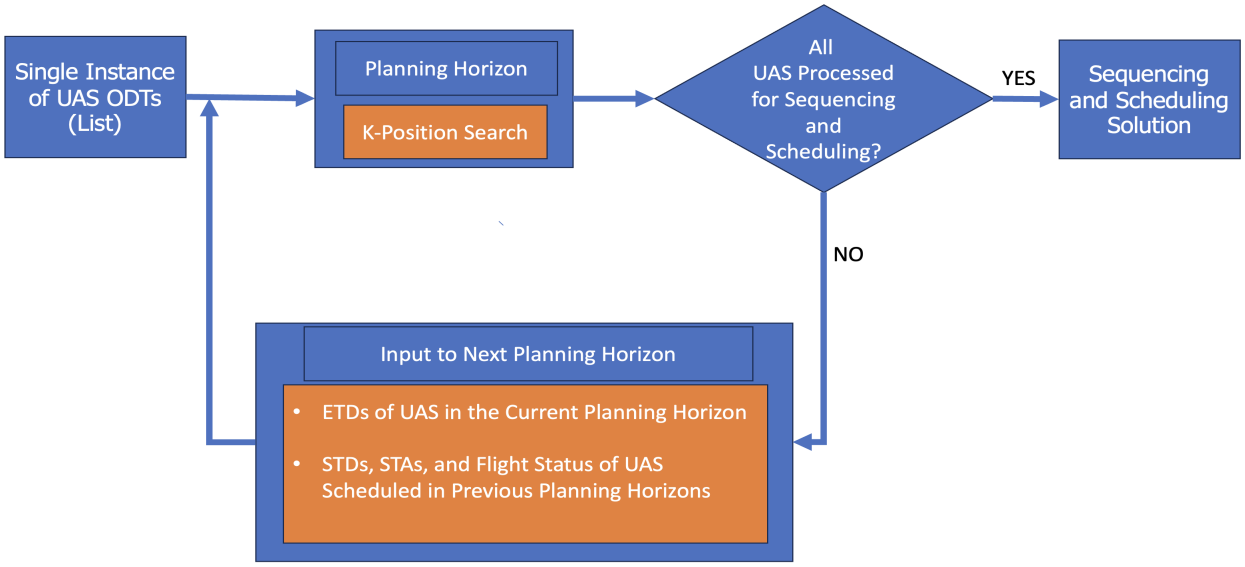


Fig. 3 Rolling Horizon with k-Position Search Methodology

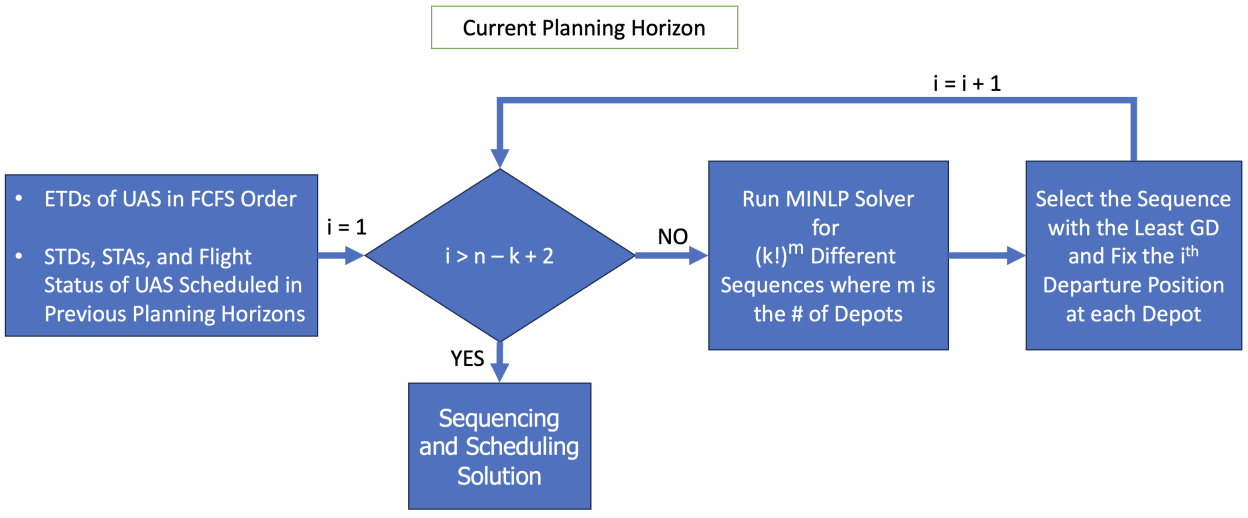


Fig. 4 Illustration of K-Position Search Algorithm for a Current Planning Horizon

- There are $k!$ possible sequences (permutations) of UAS for fixing a position. Therefore, local optimization is carried out $k!$ times to pick the preferred local sequence and fix the i^{th} position. For example, as shown in Figure 5, there are 3! possible sequences (permutations) of UAS for fixing the 1^{st} position. The local optimization is carried out 3! times to pick the preferred local sequence and fix the 1^{st} position.
- During the local neighborhood search, the preferred sequence has the least objective value among all the objective values in the $k!$ sequences. Hence, for fixing the i^{th} position, the UAS positions from the 1^{st} to the $(i-1)^{th}$ are considered fixed whereas the positions from i^{th} to the n^{th} are considered free. For example as shown in Figure 5, once 1^{st} and 2^{nd} positions are fixed with UAS $\{A_3, A_1\}$, the positions from 3^{rd} until 5^{th} are considered free.
- After fixing the i^{th} position, the local neighborhood search window advances by one position, and the process is repeated until fixing the $(n-k+1)^{th}$ position. At the end of fixing the $(n-k+1)^{th}$ position, the UAS positions at $n+2-k, ..., n$ are fixed based on the preferred sequence for the $(n-k+1)^{th}$ position in the departure sequence [16].

Fig. 5 Applying K-Position Search Method for Departure Sequencing and Scheduling at Single Depot (Depot A) Whereas Depot B Maintains FCFS Departure Order in a Given Planning Horizon

2. Simultaneous K-Position Search at Two Depots

In a given planning horizon, the total number of UAS expected to takeoff from depot A and depot B is assumed to be the same, i.e., n is the number of UAS in the planning horizon and k is the number of free (moving window) UAS involved in local neighborhood search at each depot. The process of applying the KPS method at depots A and B simultaneously involves the following steps:

- Fixing the i^{th} position in the departure sequence involves a local neighborhood search starting from the i^{th} position to the $(i+k-1)^{th}$ position at both depots. For example, as shown in Figure 6, fixing the 1^{st} position in the departure sequence involves a local neighborhood search starting from the 1^{st} position to the 2^{nd} from each depot, i.e., $\{(A_1, A_2), (B_1, B_2)\}$ when $k = 2$.
- As shown in Figure 4, there are $(k!)^2$ possible sequences (permutations) of UAS for fixing a position simultaneously at each depot. A local optimization is performed $(k!)^2$ times to pick the preferred local sequence and fix the i^{th} position. The $(2!)^2$ possible permutation in the example from Figure 6 are as follows: $\{(A_1, A_2), (B_1, B_2)\}$, $\{(A_2, A_1), (B_1, B_2)\}$, $\{(A_1, A_2), (B_2, B_1)\}$, and $\{(A_2, A_1), (B_2, B_1)\}$.
- During the local neighborhood search, the preferred sequence has the least objective value among all the objective values in the $(k!)^2$ sequences. This preferred sequence is chosen and the i^{th} positions are fixed. For example, in Figure 6, the 1^{st} positions for Depot A and Depot B have been fixed by UAS A_2 and B_1 respectively.
- After fixing the i^{th} position at each depot, the local neighborhood search window advances by one position for

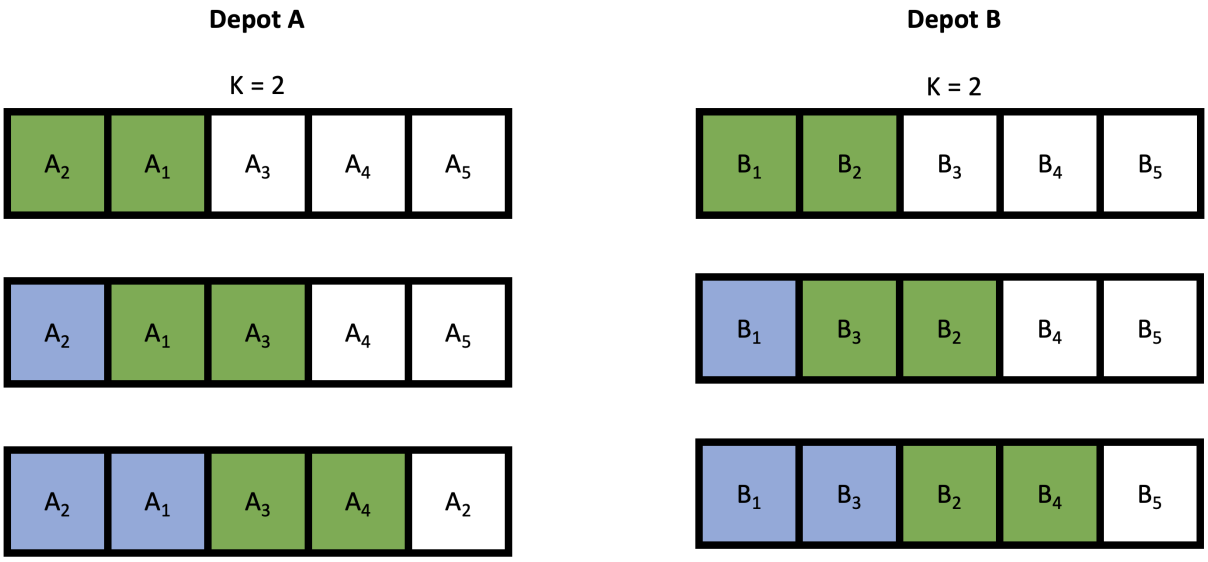


Fig. 6 Applying K-Position Search Method for Departure Sequencing and Scheduling at Depot A and Depot B Simultaneously in a Given Planning Horizon

each depot, and the process is repeated until fixing the $(n-k+1)^{th}$ position. At the end of fixing the $(n-k+1)^{th}$ positions, the UAS positions at $n+2-k, ..., n$ are fixed based on the preferred sequence for the $(n-k+1)^{th}$ position in the departure sequence [16].

3. Analysis of Computation Time

The computation time (CT) of the proposed algorithm, i.e., rolling horizon with K-Position Search can be written as follows:

$$\text{CT} = \frac{n}{r} * (\text{runtime to fix departure positions in a planning horizon}) \tag{13}$$

where n is the total number of flights, and r is the number of flights per planning horizon. Therefore, n/r is the total number of rolling horizon windows required to schedule all flights. The runtime of a computer to solve the UAS sequencing and scheduling problem is a complex function of various factors, such as, type of solver, type of processor, computer memory, type of processing, k! (local neighborhood search), total number of flights (n), number of flights per rolling horizon window (r), number of depots, and number of crossing waypoints [23, 24]. However, in this research, only the impact of k! (local neighborhood search), and total number of flights (n) on the computation time is explored. The MINLP runtime is:

$$\text{MINLP runtime} = O(C^{\text{v}} * C^{\text{c}}) \tag{14}$$

where C is a constant factor dependent on factors such as the solver, type of processor, computer memory, etc; v is the number of variables, and c is the number of constraints. In the worst case scenario, the number of variables and number of constraints both are $O(r^2)$ in the problem formulation; therefore, MINLP runtime is:

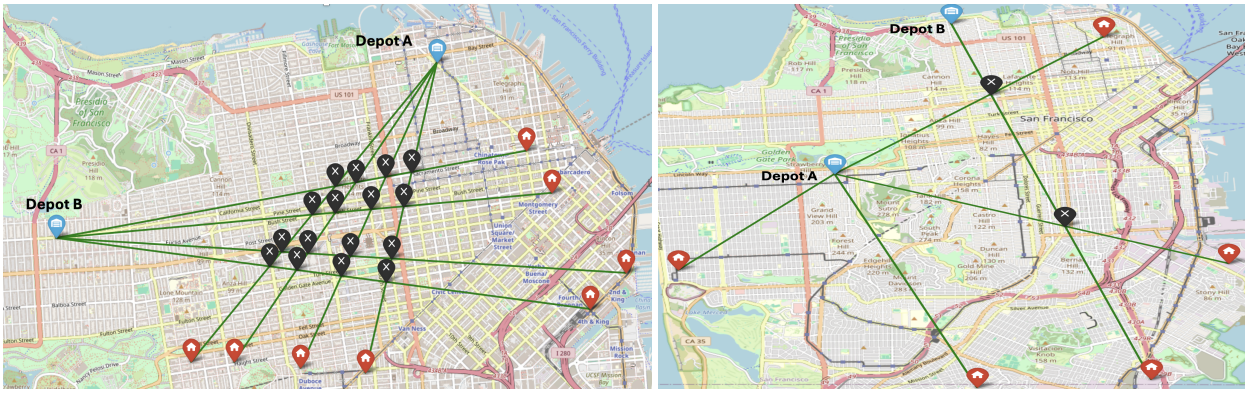
$$\text{MINLP runtime} = O(C^{r^2}) \tag{15}$$

In a series processing to fix a departure position at a depot, each sequence (permutation) of a local neighborhood search is solved one after another; therefore, the runtime to fix a departure position in a planning horizon should be multiplied by O(k!). Hence, the rolling horizon with K-Position Search algorithm has the computation time as follows:

$$\text{CT} = O(\frac{n}{r} * k! * (r-k+1) * C^{r^2}) \tag{16}$$

$$\tag{17}$$

Therefore, the algorithm with $k > 1$ has $k!$ times higher computation complexity than $k = 1$, i.e., FCFS (departure order at both depots).



(a) Map View of Simulated Scenario-1 **(b) Map View of Simulated Scenario-2**

Fig. 7 Depiction of Hub-and-Spoke Network Structure of Routes in UTM Environment (Courtesy OpenStreetMap)

IV. Simulation Studies

A. 4D Trajectories for Simulation Studies

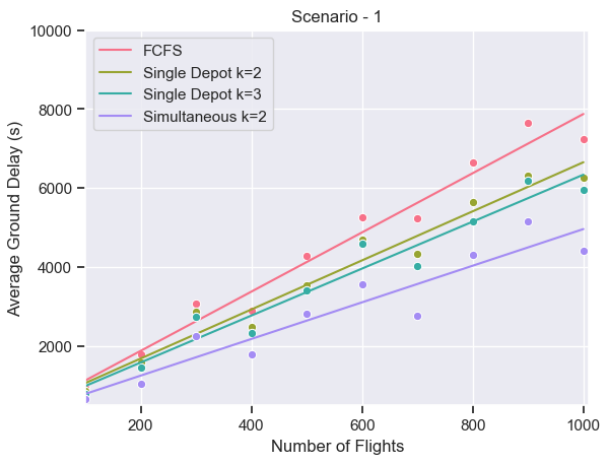
To compute flight time (FT) of a UAS to various crossing waypoints from different depots (Figure 7), the 4D trajectories of the UAS are generated using flight kinematics, flight dynamics, quadrotor performance data, and path constraints from [5, 25] using multiphase optimal control approach [26]. The formulated multiphase optimal control problem is solved using a direct collocation method for the generation of 4D trajectories [27]. A pseudospectral method is a direct collocation method that transcribes a multiphase optimal control problem to a large sparse nonlinear programming (NLP) problem [27, 28]. Finally, the corresponding NLP is then solved using Interior Point OPTimizer (IPOPT) software library [29]. Table 3 lists the parameters relevant to simulation studies.

Table 3 Parameters Relevant to Simulation Studies

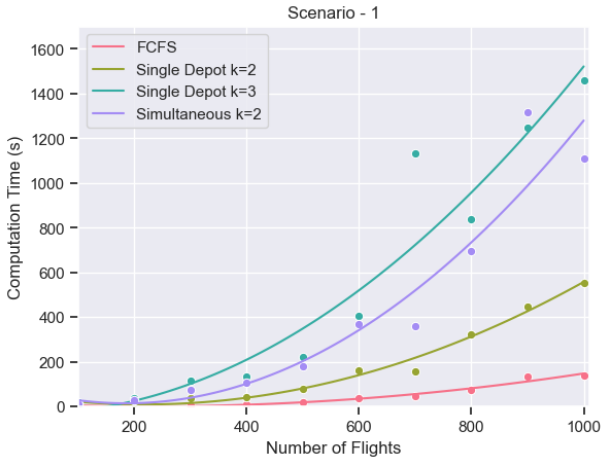
Parameter	Value(s)
Δt_{prep}	3600 sec
Δt_{sep}	60 sec
Poisson demand rate λ	1/60 (1/sec)
Length and width of operational volume blocks	500 m and 200 m
Cruise groundspeed and cruise altitude MSL	20 m/s and 121 m (400 ft)
Climb and descent flight path angle	10 deg
Rolling horizon window	300 sec
Local neighborhood search (k)	1, 2, and 3

B. Simulated Network Structure of Routes

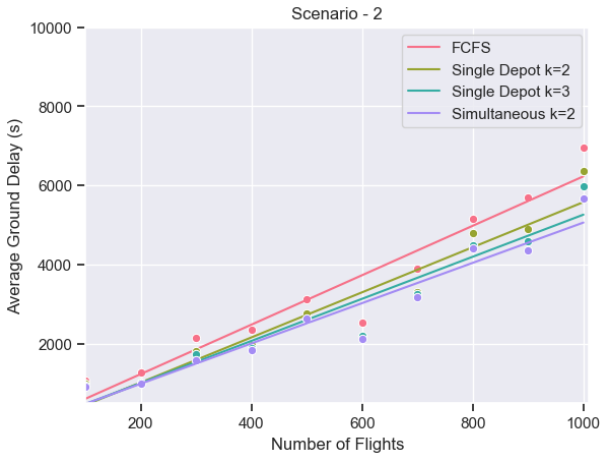
In simulation studies, as shown in Figure 7, two scenarios are considered, i.e., i) Scenario-1 consists of two depots and four drop-off locations per depot with sixteen en-route crossing waypoints, and ii) Scenario-2 consists of two depots and four drop-off locations for depot A and one drop-off location for depot B with two en-route crossing waypoints. Therefore, given the higher number of en-route crossing waypoints in Scenario-1, it is more complex than Scenario-2 [30]. The simulated network structure of routes, i.e., depot and corresponding drop-off locations, are considered hub-and-spoke type [3, 19, 20] in the San Francisco metropolitan area.



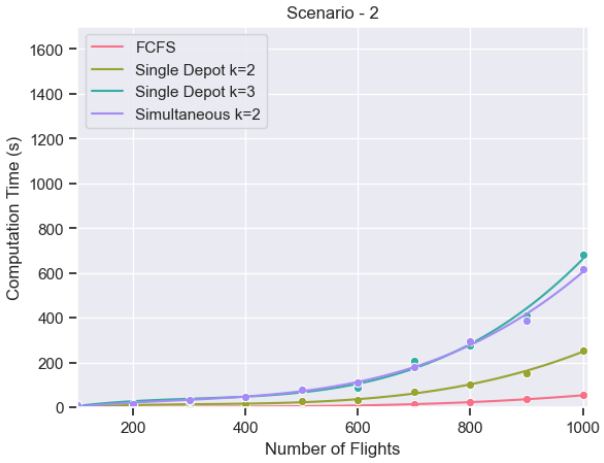
(a) Average Ground Delay (Scenario-1)



(b) Computation Time (Scenario-1)



(c) Average Ground Delay (Scenario-2)



(d) Computation Time (Scenario-2)

Fig. 8 Average Ground Delay and Computation Time as a Function of Number of Flights for the Heuristic Algorithm Using MacOS, M2 Pro for Simulated Scenario-1 and Scenario-2

C. UAS Traffic Simulation

The UAS departure traffic at both depots is simulated using a Poisson distribution with an on-demand service request rate (λ). The Poisson distribution has been used to model the on-demand service request at two depots because each demand request is assumed to be discrete, independent, and mutually exclusive, but at an average rate (λ) when viewed as a group for a set of demands.

D. Computing Resources

The UAS traffic simulation algorithm and heuristic algorithm (rolling horizon with the KPS) are implemented in Python 3.11 on a MacOS-based laptop with a 2.6 GHz 12-Core Apple M2 Pro processor and 16 GB of memory. The MINLP optimization problem is implemented in the PuLP modeling library [31, 32]. The simulations for UAS traffic and the heuristic algorithm for strategic deconfliction are run on the above mentioned MacOS-based laptop.

V. Results

As stated in Simulation Studies section IV and Table 3, the UAS departure traffic at depot A and depot B are simulated using Poisson distribution with an average departure separation of 60 seconds. Each UAS is randomly assigned one of the routes with an origin as a given depot. The strategic deconfliction for various simulations of scenario-1 and

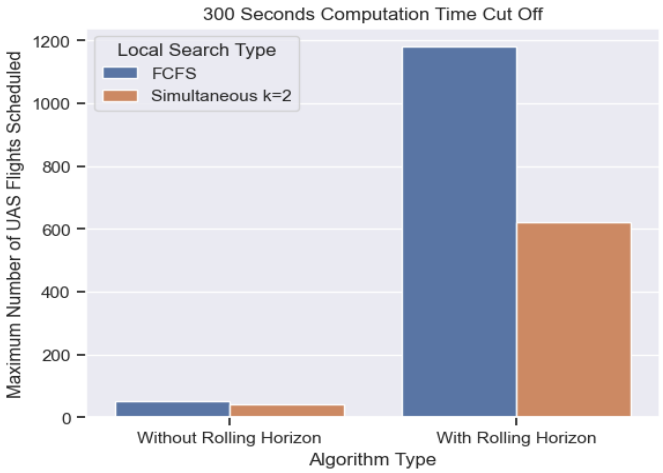


Fig. 9 Maximum Number of UAS Flights Scheduled in 300 Seconds Computation Time Cut Off

scenario-2 is carried out using the heuristic algorithm (rolling horizon with the KPS) with different values of k (1, 2, and 3) and applied to either a single depot or both depots. The timeline of a simulation is decomposed into planning horizons of 300 seconds. The simulations are categorized based on the number of UAS flights in the scenario.

In Figure 8, the average ground delay and computation time are averaged over all simulations for a given scenario, size of the rolling horizon window, and total number of flights. The following can be observed from Subfigures of Figure 8:

- The average ground delay (ground delay per UAS flight) increases with the total number of flights for a given size of the rolling horizon window, Poisson departure rate, number of depots and routes.
- The simultaneous KPS with k=2 has the lowest average ground delay compared to the FCFS, single depot with k=2, and single depot with k=3 approaches for both Scenario-1 and Scenario-2.
- The FCFS departure order at depots has the lowest computation time, i.e., the departure order is the same as the service request order. The low computation time for the FCFS compared to single depot with k=2, and single depot with k=3 can be attributed to the fact that computation complexity of local neighborhood search algorithms are known to be function of k! as shown in Equation 16, which has a value of 1 for the FCFS.
- The average ground delay reduces at the cost of an increase in the computation time with an increase in the value of k (local neighborhood search) in the KPS for a given number of UAS, rolling horizon window, and number of depots involved in the local neighborhood search. Therefore, computation time is highly sensitive to any large factor of k!. Hence, for feasibility, k must be kept small, i.e., 1, 2, or 3.

In Figure 9, the average computation time are averaged over all simulations for a given scenario, size of the rolling horizon window, and total number of flights. As shown in Figure 9, the maximum number of UAS flights scheduled in computation time cut-off of 300 seconds increased at least by an order of magnitude with rolling horizon compared to without rolling horizon for $k = 2$ (simultaneous) and $k = 1$.

VI. Conclusions

This research focused on the strategic deconfliction of unmanned aircraft systems for an urban package delivery environment with two depots and multiple drop-off locations. A heuristic algorithm called rolling horizon with k-position search was used to compute both the departure sequence and scheduled time of departure of each unmanned aircraft system at a depot, considering temporal constraints at en-route crossing waypoints and depots for strategic deconfliction. The simulation studies showed that an increase in the value of k (the parameter used to determine the size of the local neighborhood) reduces the average ground delay at the cost of an increase in the computation time for a given number of flights, size of the rolling horizon window, and number of depots involved in the local neighborhood search. The study also showed at least an order of magnitude increase in the maximum number of flights scheduled with the integration of rolling horizon (time decomposition) than without the integration of rolling horizon in the heuristic algorithm for a given computation time cut off. In the future, during the strategic deconfliction uncertainties such as

wind, departure time, and lateral path will be considered.

VII. Acknowledgements

The material is based upon work supported by the National Aeronautics and Space Administration under Contract Number NNA16BD14C, managed by the Universities Space Research Association (USRA). Authors thank Mr. José Ignacio de Alvear Cárdenas, Dr. Gano Chatterji, Dr. Hyo-Sang Yoo, and Mr. Andrew Cone for their constructive feedback regarding the algorithm and research paper.

References

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

[2] Federal Aviation Administration, "Advisory Circular: 107-2, Small Unmanned Aircraft Systems (sUAS)," , 2016. URL www.faa.gov/documentlibrary/media/advisory_circular/ac_107-2.pdf.

[3] "UTM ConOps Version2," *Federal Aviation Administration NextGen Office*, 2020. URL https://www.faa.gov/sites/faa.gov/files/2022-08/UTM_ConOps_v2.pdf.

[4] Rios, J. L., Homola, J., Craven, N., Verma, P., and Baskaran, V., "Strategic Deconfliction Performance: Results and Analysis from the NASA UTM Technical Capability Level 4 Demonstration," 2020.

[5] Pradeep, P., Munishkin, A. A., Kalyanam, K. M., and Erzberger, H., "Strategic Deconfliction of Small Unmanned Aircraft Using Operational Volume Blocks at Crossing Waypoints," *AIAA SciTech 2023 Forum*, 2023, p. 1654.

[6] Erzberger, H., Paielli, R. A., Isaacson, D. R., and Eshow, M. M., "Conflict detection and resolution in the presence of prediction error," *1st USA/Europe Air Traffic Management R&D Seminar, Saclay, France*, Citeseer, 1997, pp. 17–20.

[7] Lauderdale, T. A., Pradeep, P., Edholm, K.-M., and Bosson, C. S., *Separation at Crossing Waypoints Under Wind Uncertainty in Urban Air Mobility*, 2021. https://doi.org/10.2514/6.2021-2351, URL https://arc.aiaa.org/doi/abs/10.2514/6.2021-2351.

[8] Erzberger, H., Lauderdale, T., and Chu, Y., "Automated conflict resolution, arrival management, and weather avoidance for air traffic management," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of aerospace engineering*, Vol. 226, No. 8, 2012, pp. 930–949.

[9] Verma, S. A., Monheim, S. C., Moolchandani, K. A., Pradeep, P., Cheng, A. W., Thipphavong, D. P., Dulchinos, V. L., Arneson, H., Lauderdale, T. A., Bosson, C. S., Mueller, E. R., and Wei, B., "Lessons Learned: Using UTM Paradigm for Urban Air Mobility Operations," *2020 AIAA/IEEE 39th Digital Avionics Systems Conference (DASC)*, 2020, pp. 1–10. https://doi.org/10.1109/DASC50938.2020.9256650.

[10] Kim, J., and Atkins, E., "Airspace Geofencing and Flight Planning for Low-Altitude, Urban, Small Unmanned Aircraft Systems," *Applied Sciences*, Vol. 12, No. 2, 2022, p. 576.

[11] Johnson, M., and Larrow, J., "UAS Traffic Management Conflict Management Model," 2020. URL https://www.nasa.gov/sites/default/files/atoms/files/2020-johnson-nasa-faa.pdf.

[12] Malik, W., and Jung, Y. C., *Exact and Heuristic Algorithms for Runway Scheduling*, 2016. https://doi.org/10.2514/6.2016-4072, URL https://arc.aiaa.org/doi/abs/10.2514/6.2016-4072.

[13] Balakrishnan, H., and Chandran, B. G., "Algorithms for scheduling runway operations under constrained position shifting," *Operations Research*, Vol. 58, No. 6, 2010, pp. 1650–1665.

[14] Karp, R. M., *Reducibility among combinatorial problems*, Springer, 2010.

[15] Berthold, T., "Primal heuristics for mixed integer programs," Ph.D. thesis, Zuse Institute Berlin (ZIB), 2006.

[16] Pradeep, P., and Wei, P., "Heuristic Approach for Arrival Management of Aircraft in On-Demand Urban Air Mobility," *Journal of Aerospace Information Systems*, 2020, pp. 1–12.

[17] Samà, M., D'Ariano, A., and Pacciarelli, D., "Optimal aircraft traffic flow management at a terminal control area during disturbances," *Procedia-Social and Behavioral Sciences*, Vol. 54, 2012, pp. 460–469.

[18] Kleinbekman, I. C., Mitici, M., and Wei, P., "Rolling-horizon electric vertical takeoff and landing arrival scheduling for on-demand urban air mobility," *Journal of Aerospace Information Systems*, Vol. 17, No. 3, 2020, pp. 150–159.

[19] Evans, A. D., Egorov, M., Anand, A., Campbell, S. E., Zanlongo, S., Young, T., and Sarfaraz, N., "Safety Assessment of UTM Strategic Deconfliction," *AIAA Scitech 2023 Forum*, 2023, p. 0965.

[20] Roche, C., Oakes, S., Bender, K., Magyarits, S., and Homola, J. R., "Unmanned Aircraft System Traffic Management (UTM) Research Transition Team (RTT) Concept Working Group-Concept & Use Cases Package# 2 Addendum: Technical Capability Level 3," Tech. rep., 2018.

[21] Rogovs, S., Nikitina, V., and Gerdts, M., "A novel mixed-integer programming approach for the aircraft landing problem," *Frontiers in Future Transportation*, Vol. 3, 2022, p. 30.

[22] Cococcioni, M., and Fiaschi, L., "The Big-M method with the numerical infinite M," *Optimization Letters*, Vol. 15, No. 7, 2021, pp. 2455–2468.

[23] Balakrishnan, H., and Chandran, B., "Scheduling aircraft landings under constrained position shifting," *AIAA guidance, navigation, and control conference and exhibit*, 2006, p. 6320.

[24] Prevot, T., and Lee, P. U., "Trajectory-based complexity (TBX): A modified aircraft count to predict sector complexity during trajectory-based operations," *2011 IEEE/AIAA 30th Digital Avionics Systems Conference*, IEEE, 2011, pp. 3A3–1.

[25] Pradeep, P., Park, S. G., and Wei, P., "Trajectory optimization of multirotor agricultural UAVs," *2018 IEEE Aerospace Conference*, IEEE, 2018, pp. 1–7.

[26] Bryson, A. E., *Applied optimal control: optimization, estimation and control*, Routledge, 2018.

[27] Becerra, V. M., "Solving complex optimal control problems at no cost with PSOPT," *2010 IEEE International Symposium on Computer-Aided Control System Design*, 2010, pp. 1391–1396. https://doi.org/10.1109/CACSD.2010.5612676.

[28] Garg, D., Patterson, M., Hager, W. W., Rao, A. V., Benson, D. A., and Huntington, G. T., "A unified framework for the numerical solution of optimal control problems using pseudospectral methods," *Automatica*, Vol. 46, No. 11, 2010, pp. 1843–1851.

[29] Wächter, A., and Biegler, L. T., "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, Vol. 106, 2006, pp. 25–57.

[30] Xue, M., and Do, M., "Scenario Complexity for Unmanned Aircraft System Traffic," *AIAA Aviation 2019 Forum*, 2019, p. 3513.

[31] Mitchell, S., OSullivan, M., and Dunning, I., "PuLP: a linear programming toolkit for python," *The University of Auckland, Auckland, New Zealand*, Vol. 65, 2011.

[32] Mitchell, S., et al., "An introduction to pulp for python programmers," *Python Papers Monograph*, Vol. 1, No. 14, 2009, p. 2009.