Demand-Capacity Balancing Algorithms for Urban Air Mobility Operations

Kushal A. Moolchandani
Universities Space Research Association (USRA)
at *NASA Ames Research Center*
Moffett Field, USA
kushal.a.moolchandani@nasa.gov

Hanbong Lee
Aviation Systems Division
NASA Ames Research Center
Moffett Field, USA
hanbong.lee@nasa.gov

***Abstract*—Demand-Capacity Balancing is one of strategic air traffic management methods that can be applied for safe and conflict-free operations. This paper proposes demand-capacity balancing algorithms specifically for application to Urban Air Mobility operations. NASA previously developed and simulated a demand-capacity balancing algorithm for urban air mobility operations that resolved predicted demand-capacity imbalances at vertiports by assigning pre-departure delays to aircraft. In contrast, the algorithms proposed in this paper resolve imbalances at enroute waypoints in addition to vertiports and make use of aircraft speed changes in addition to pre-departure delay as an imbalance resolution mechanism. These new algorithms are implemented in a new fast-time simulation tool that has been developed for NASA's urban air mobility airspace service development research. The implementation of these algorithms along with results of their simulations for a set of traffic scenarios is presented, followed by a discussion of insights derived and proposals for development of future demand-capacity balancing services for urban air mobility.**

Keywords—urban air mobility, demand-capacity balancing

I. Introduction

Urban Air Mobility (UAM) is an upcoming aviation market intended to provide air transportation services for people and cargo in and around metropolitan areas. Being a novel aviation market, UAM may need new airspace services that are different from those used for commercial aviation and developed and certified to support the envisioned state of high-density and highly autonomous UAM operations being managed with a collaborative, federated framework those does not rely on ATC services [1]. NASA has been running simulations in support of advancement of the essential aircraft and airspace service technologies that will be needed for mature UAM operations. These simulations fill the data availability gaps that result from the absence of real-world operations and their related performance data. Recent simulations, described in [2], demonstrated a UAM operational state where airspace services developed by different independent service providers were integrated and operated within a common simulation environment. These simulations focused on the evaluation of Demand-Capacity Balancing (DCB) services developed independently by industry partners and integrated with Providers of Services for UAM (PSUs) for multiple operators sharing airspace resources in a common simulation environment for test traffic scenarios.

In air traffic management operations, Demand-Capacity Balancing (DCB) is one of the seven interdependent components that are integrated to provide safe and efficient air transportation services [3]. DCB service is one of a set of airspace services that provide strategic conflict management capabilities, which aim to reduce the need for enroute separation provision services. DCB is applied whenever the demand for an airspace resource exceeds its capacity to serve that demand. DCB service evaluates the state of air traffic flows to determine where and when flights operate.

Reference [4] describes a DCB algorithm that resolves predicted demand-capacity imbalances at vertiports by assigning pre-departure delays only. While the proposed algorithm is guaranteed to work, the lack of resolution mechanisms besides pre-departure delay means that the solution obtained can lead to unacceptably large delays. This is because, regardless of which of the flight's enroute resources has a demand-capacity imbalance, such a DCB algorithm will delay flights on ground. This, in turn, will likely lead to network-wide inefficiencies because, while an aircraft waits on the ground, it occupies a vertiport that could be used by another flight for departure or arrival. Thus, delay assigned to one aircraft affects another aircraft planning to use the same vertiport.

To accommodate larger volume of traffic at midterm to mature stages of urban air mobility market development while minimizing the need for tactical separation, more advanced DCB algorithms need to be investigated. More advanced DCB algorithms can play a role in promoting operational efficiency. Besides pre-departure delay, demand can be moderated by altering aircraft speed on a route segment or flying a longer path to mitigate congestion. However, while enroute maneuvers can provide network-wide benefits in form of improved operations throughput and resource capacity utilization, additional energy needed for enroute maneuvers, reserves, and flight to alternate airport should be planned for during flight planning prior to flight.

This paper proposes two new DCB algorithms that build upon the algorithm that was described in [4]. The details of the algorithms and the results obtained by simulating them for a common traffic scenario are presented. The DCB algorithms are compared against one another and the simulation results examined to obtain insights for the future development of more advanced DCB algorithms. These algorithms are implemented in a new, fast-time simulation tool developed for modeling

UAM operations; the next section presents a brief overview of this tool.

II. Fast-time Simulation Tool Architecture

As mentioned, NASA has been conducting a series of simulation activities in support of development of new airspace services and structures for UAM operations [5]. To aid research on new supporting airspace services for UAM, there was a need for a fast-time simulation tool with customized implementation of the various components of the UAM system. A simulation tool can aid research by allowing for rapid prototyping and testing of new services and algorithms designed to support UAM operations. The UAM Simulation Tool for Airspace services Research (USTAR) fulfills this need as a lightweight simulation tool that can be quickly configured to match different airspace services system architectures and simulate a range of traffic scenarios. Since, USTAR is a fast-time simulation tool that provides speed and flexibility of implementation and execution, it has been used for various use cases such as observing the results of simulating different traffic scenarios, exploring the performance of various system architectures, and testing the behavior of newly developed algorithms for airspace services. The latter is also the use case presented in this paper. The following is a brief description of the tool; detailed description of USTAR will be documented in future technical reviews that are currently in writing [6].

A. USTAR System Design

Fig. 1 shows the current USTAR architecture, along with interactions among various objects. USTAR includes a Simulation Engine, Fleet Operator, Provider of Services for UAM (PSU), Resource Planning Service (RPS), and Simulation Data Collector classes. The main classes within the tool are:

1. *Simulation Engine:* This class contains details of the simulation setup, including names of input and output files, simulation length, and related metadata. This class also includes a "run" function that controls the flow of the simulation. During execution, the "run" function steps in one-second time increments for the duration of the traffic scenario.

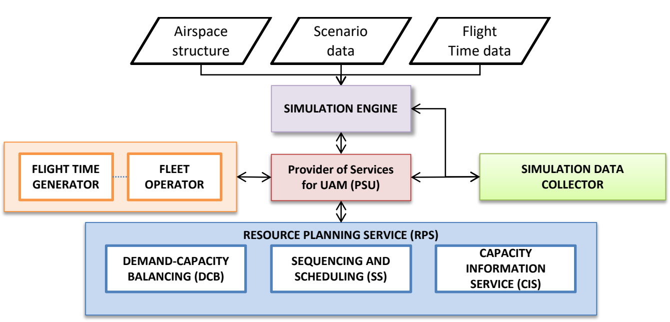


Fig. 1. Architecture of USTAR.

2. *Fleet Operator*: This class defines a UAM fleet operator that takes traffic scenario as input and requests flights for departure. This class includes a "Flight Time Generator" function which returns the flight time between waypoints. Since USTAR currently lacks a trajectory generation algorithm, the "Flight Time Generator" block in Fig. 1 returns requested flight times from a pre-defined lookup table. During execution, the fleet operator controls and flies all aircraft over the entire route network.

3. *Provider of Services for UAM (PSU)*: A PSU serves as the interface between the fleet operator(s) and all airspace services. It receives flight requests from an operator and, depending on the logic of flight scheduling, calls the relevant airspace services to check strategic deconfliction, schedule flights, and get approvals. In its current implementation, the PSU calls the DCB function, which is a sub-class of the RPS class, and receives, in return, information on the flight maneuvers such as pre-departure delay assignment or speed change to be applied to resolve any potential demand-capacity imbalances.

4. *Resource Planning Service (RPS)*: This class includes various airspace services as sub-classes, including the DCB service, the sequencing and scheduling service (SS), and the capacity information service (CIS). The DCB sub-class includes different functions of each of the imbalance resolution algorithms proposed in this paper. This sub-class receives the complete trajectory information for a set of flights, compares the estimated times of arrival at all waypoints along the route of each flight, and returns the maneuver to be performed to avoid any predicted imbalance. The current implementation of sequencing and scheduling sub-class ensures that consecutive aircraft using a resource are separated by the specified minimum time separation. Finally, the CIS sub-class returns the available capacity at each constrained resource at the requested time.

5. *Simulation Data Collector*: An object of this class stores all scenario and simulation data for post-simulation analysis. All other objects in the simulation interface with this object to save their respective output data.

To set up a simulation, USTAR requires a pre-defined airspace structure, flight time data, and a traffic scenario definition as inputs. Airspace structure data includes the list of all vertiports and waypoints along with their latitude, longitude, and altitude values. The current version of USTAR includes the airspace information for Dallas/Fort Worth metropolitan region. For each simulation run, a subset of vertiports included within the airspace are selected as origins and destinations for the flights in the simulation. Thus, for each simulation run, a route network is defined connecting the vertiports by paths through waypoints. All vertiports and a few selected waypoints are considered to be constrained resources with finite capacities, specified as number of operations within a time interval; the capacities remain fixed for the simulation. Flight time data includes the time taken to fly each segment of pre-defined routes in the given route network. Finally, traffic scenario information includes the list of flights to be simulated along with their callsign, origin, destination, and desired departure times. In Sec. IV, we describe these inputs specific to our simulations.

III. Advanced Demand-Capacity Balancing Algorithms for UAM Operations

Demand-Capacity Balancing (DCB) aims to reduce the need for tactical separation provision, which is the next air traffic management layer following strategic conflict management for safe flight operations. Reference [4] described the DCB procedure for balancing vertiport demand with capacity with pre-departure delay, which we will refer to as *DCB-Vertiport* algorithm in this paper. The disadvantage of using pre-departure delays as the exclusive resolution mechanism is the potential for very large delays assigned to some flights. Another disadvantage is that an aircraft may continue to occupy the vertipad while it waits, potentially delaying other aircraft in the network. Large delays also make UAM an unattractive option for passengers and impose economic penalties such as higher operating costs, further providing an incentive to develop traffic management concepts that more efficiently allocate delays.

Building upon the DCB algorithm developed in Ref. [4], two new algorithms are proposed for mitigating demand-capacity imbalances at all constrained trajectory waypoints by employing speed control. The DCB service receives full trajectory information for a set of flights in a given time step. For each flight, the DCB service calls a specified algorithm that predicts and resolves imbalances at each of the constrained waypoints along the route. The order of selection of flights for scheduling can be based on different criteria such as the desired departure time of flights, the time at which the flight was requested, or some other prioritization mechanism. In all simulations presented in this paper, flights are selected in the chronological order of their proposed departure times. If multiple flights are proposed to depart at the same time, they are randomly selected. Fig. 2 shows the outer-loop logic common to all DCB algorithms. All DCB algorithms presented consider one flight at a time. Each call to the DCB algorithm returns the amount of delay assigned to the current flight and the logic will always find a feasible solution regardless of the amount of total delay assigned. In the following two sub-sections, we describe two new DCB algorithms.

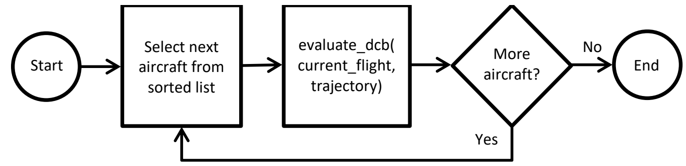


Fig. 2. DCB service logic to evaluate each flight at each constrained waypoint.

A. DCB at Enroute Waypoints and Vertiports with Pre-departure Delays Only

The first DCB algorithm proposed resolves imbalances at enroute waypoints in addition to vertiports; this algorithm is referred to as *DCB-Waypoint*. *DCB-Waypoint* is a superset of *DCB-Vertiport*, that is, if the *DCB-Waypoint* algorithm were to be constrained to consider only vertiports, it will represent *DCB-Vertiport*. Like *DCB-Vertiport*, *DCB-Waypoint* selects one flight at a time and resolves imbalances at one node at a time, where a node represents a constrained resource along the given route, including the vertiports and enroute waypoints. *DCB-Waypoint* resolves imbalances starting from the origin vertiport followed by the constrained waypoints in their sequential order along the route and, finally, the destination vertiport.

Like *DCB-Vertiport*, *DCB-Waypoint* also resolves imbalances by assigning pre-departure delays only. However, because *DCB-Waypoint* considers more constrained waypoints, i.e., enroute waypoints in addition to the vertiports, it will assign more delays as compared to *DCB-Vertiport*. On the other hand, *DCB-Waypoint* algorithm will ensure that demand remains at or below capacity at all waypoints along a flight route and not just the vertiports, and could reduce the need of tactical separation in the air. In the results presented below, performance of *DCB-Waypoint* is compared with *DCB-Vertiport* in terms of metrics such as number of flights delayed and amounts of delay assigned.

Fig. 3 shows the flowchart of the proposed *DCB-Waypoint* algorithm. Note that the resolution logic presented in Fig. 3 receives a single flight at a time including its full trajectory information. Starting with the origin vertiport, this algorithm selects one constrained resource at a time and compares to see if the current flight causes a demand-capacity imbalance. If it does cause an imbalance, the logic of resolution delays the departure time of the aircraft to the start of the next time bin. Then, using the new departure time, the algorithm calculates the new arrival times and the associated time bins at all waypoints on the route. The algorithm then calculates the amount of delay to be assigned and returns it to the calling function.

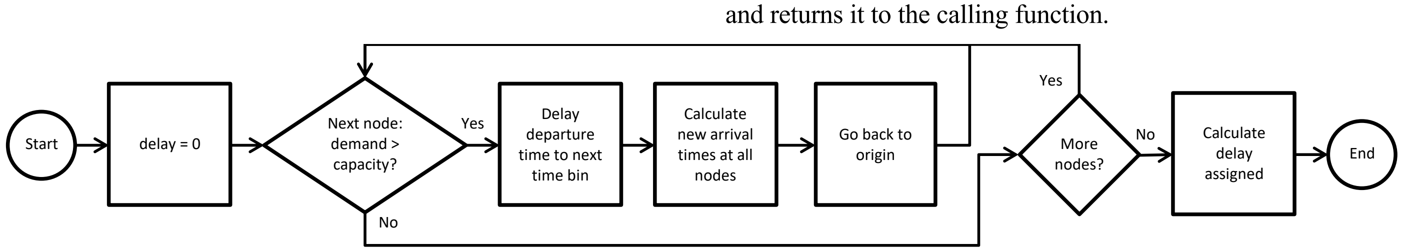


Fig. 3. Flowchart of the *DCB_Waypoint* algorithm to resolve imbalances.

B. *DCB Applied at Enroute Waypoints and Vertiports with Pre-Departure and Enroute Delays*

Although *DCB-Waypoint* resolved imbalances at all constrained waypoints, it still continued to use pre-departure delay as the sole imbalance resolution algorithm. With more constrained resources to deconflict, the pre-departure delays to will tend be larger than those assigned by *DCB-Vertiport,* further increasing potential network-wide inefficiencies. The next DCB algorithm, called *DCB-Enroute-Delay,* assigns enroute delays, in addition to delays at vertiports. This algorithm uses information on flight speeds to ensure that the assigned delays are feasible, meaning that the delays are such that variations in flight speeds are within range of feasible aircraft speeds. Fig. 4 shows the flowchart of the *DCB-Enroute-Delay* algorithm.

Like the previous two DCB algorithms, *DCB-Enroute-Delay* receives full trajectory information for a single flight with predicted demand-capacity imbalance. Unlike the previous algorithms, for each waypoint where an imbalance is predicted, the algorithm assigns delay at that waypoint until it finds a feasible solution. This is equivalent to issuing a speed reduction command to the aircraft; this speed reduction is applied only to the flight segment immediately preceding the waypoint under consideration. The algorithm then compares the new reduced flight speed to a prescribed lower speed limit, e.g., the aircraft stall speed, to ensure that the speed lies within a feasible range. If the speed is below the lower flight speed limit, the aircraft is not assigned airborne delay and instead the imbalance is resolved using pre-departure delay, which is similar to the approach of *DCB-Waypoint*.

IV. DCB Algorithm Evaluation

In this section, two traffic scenarios are simulated with the three DCB algorithms and the results are compared. The first traffic scenario is a short scenario of just five operations that is used to verify the algorithms' behavior. Next, a longer traffic scenario defined in Ref. [7] was simulated and the performance of the algorithms was evaluated both in the presence and absence of uncertainty.

A. *Experiment Setup*

A subset of seven vertiports plus four waypoints in the Dallas/Fort Worth area were selected as the nodes of the route network used in the following simulations. These 11 nodes are connected by 10 routes; Fig. 5 shows the route network used in the simulations. A single route connects any pair of vertiports, so that identifying the origin and destination of a flight uniquely defines the route that an aircraft will fly.

Together, these 11 nodes are the constrained resources that the DCB algorithms evaluate to ensure that the demand at any time remains at or below capacity. Node capacities are specified as input to the simulation and defined in terms of the number of operations within fixed time intervals. At the vertiports, the sum of takeoffs and landings taken together count as the number of operations, while at the enroute waypoints, it is the number of flights passing through within the time interval. The time intervals are 12-minute durations and referred to as "time bins." A limit of two operations is set per time bin at all constrained resources.

Flight time information obtained from one of NASA's trajectory simulation runs was input to the simulation. This flight time information is used to estimate the arrival times at each waypoint and destination vertiport of the flights. In the simulations presented in this paper, a single aircraft model is used for all routes of the network. Therefore, all aircraft fly at the same speed on all segments of the path. A single operator handles all flights over the entire network.

Table I shows the short five-flight scenario that we used for verification of the algorithms, as an example of traffic scenario specification. Each row in this table corresponds to a single flight and specifies the flight's origin and destination vertiports, and the desired departure time in seconds from the start of simulation from the origin vertiport.

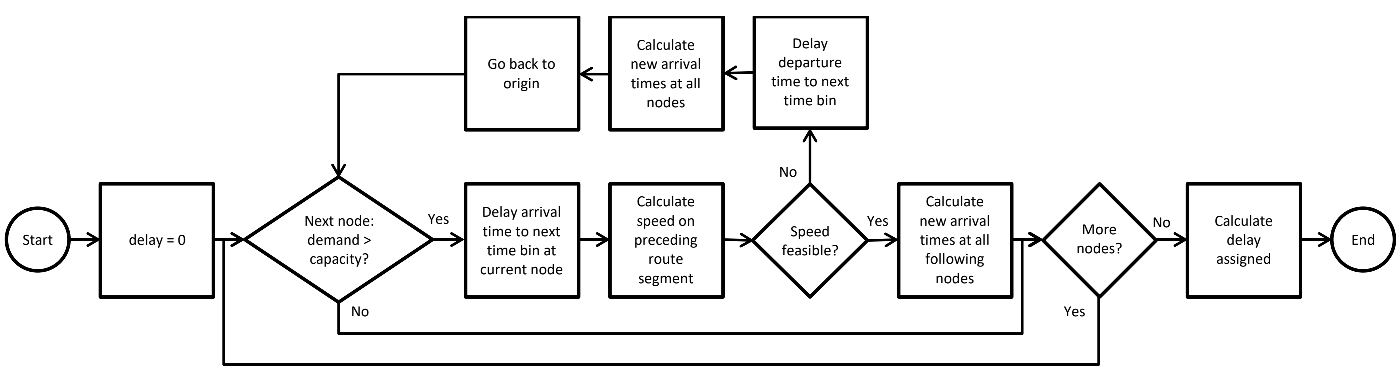


Fig. 4. Flowchart of the *DCB_Enroute_Delay* algorithm to resolve imbalances.

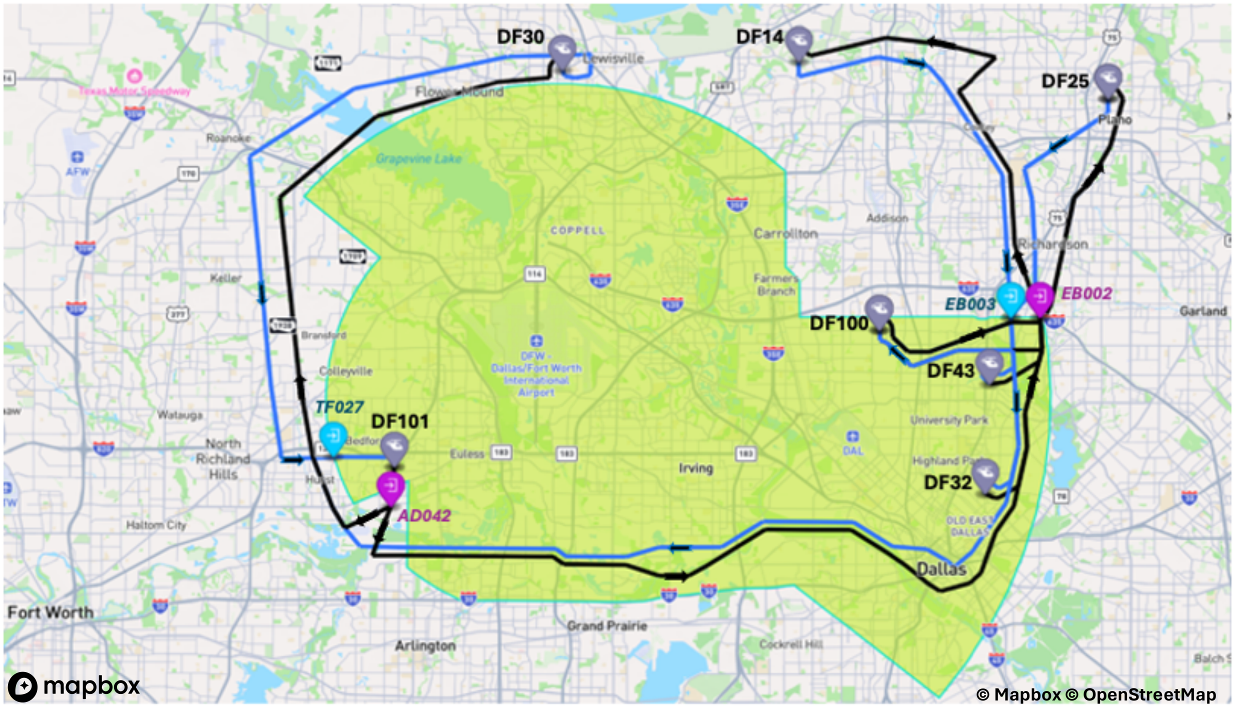


Fig. 5. Route network in Dallas/Fort Worth urban area.

TABLE I. TRAFFIC SCENARIO

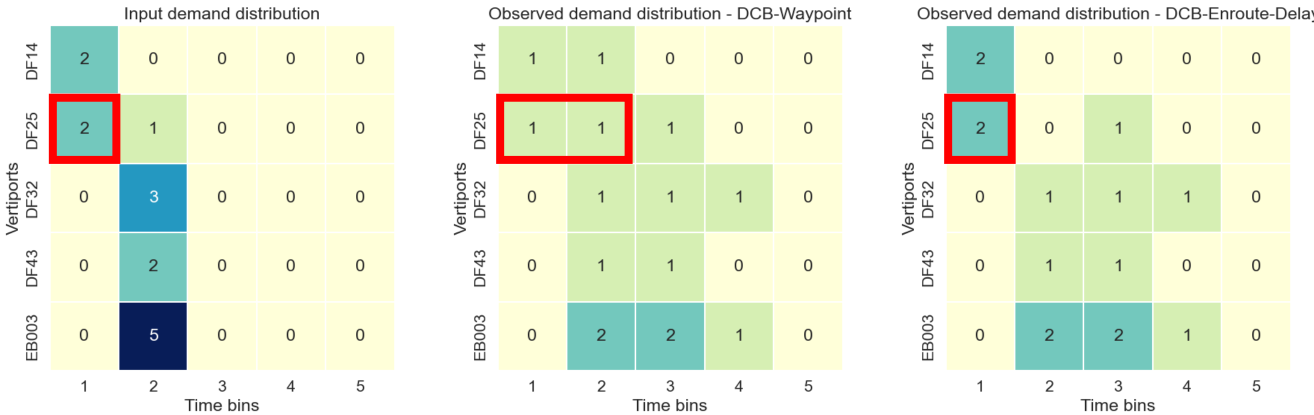
No.	Origin	Destination	Departure Time (sec)
1	DF25	DF32	420
2	DF14	DF43	540
3	DF25	DF32	660
4	DF14	DF43	660
5	DF25	DF32	720

B. *Verification of DCB Algorithms Resolution Capabilities*

To verify that the proposed DCB algorithms behave in the expected manner, a simulation is set up with just two routes that cross each other at a single waypoint in a traffic scenario with five flights. In this scenario, DF14 and DF25 are the two origins of the two routes and DF43 and DF32 are the destinations corresponding to the two origins, respectively. EB003 is the waypoint common to the two routes (see Fig. 5).

The demand in each time bin, i.e., the occupancy of the five nodes of this network can be visualized using a heatmap. Fig. 6(a) shows the occupancy specified as input in the traffic scenario; the numbers in each cell indicate the number of flights proposed at each of the five nodes in all time bins. This scenario was intentionally designed with excess demand over capacity at two locations: vertiport DF32 and crossing waypoint EB003 both have more than two operations planned in time bin labeled "1."

From Figs. 6(b) and 6(c), it can be observed that both algorithms successfully resolve all imbalances. However, the difference lies in the amount of assigned pre-departure delay. *DCB-Waypoint* delays three flights, one at DF14 and two at DF25, so that all time bins at these two vertiports have an occupancy of one. *DCB-Enroute-Delay*, on the other hand, also delays three flights but resolves two out of the three predicted imbalances by slowing down the aircraft in flight instead of pre-departure delay, thereby assigning lower ground delay. More specifically, with *DCB-Waypoint*, one of the two flights scheduled in time bin "1" at DF25 (shown by the red box in Fig. 6(a)) is delayed to the next time bin (shown by the expanded red box in Fig. 6(b)), to resolve the imbalance at EB003. In comparison, *DCB-Enroute-Delay* delays the same flight using airborne delay instead of ground delay, as shown by the red box in Fig. 6(c). Fig. 7 shows this more clearly, where *DCB-Waypoint* assigns pre-departure delay to three aircraft, while *DCB-Enroute-Delay* assigns pre-departure delay to just one aircraft and two additional aircraft are delayed while airborne, so that they takeoff at the originally desired time.



(a) (b) (c)

Fig. 6. Heatmap of observed demand: (a) Input demand defined by the traffic scenario, (b) Using the *DCB-Waypoint* algorithm, (c) Using the *DCB-Enroute-Delay* algorithm.

From Figs. 6(b) and 6(c), it can be observed that both algorithms successfully resolve all imbalances. However, the difference lies in the amount of assigned pre-departure delay. *DCB-Waypoint* delays three flights, one at DF14 and two at DF25, so that all time bins at these two vertiports have an occupancy of one. *DCB-Enroute-Delay*, on the other hand, also delays three flights but resolves two out of the three predicted imbalances by slowing down the aircraft in flight instead of pre-departure delay, thereby assigning lower ground delay. More specifically, with *DCB-Waypoint*, one of the two flights scheduled in time bin "1" at DF25 (shown by the red box in Fig. 6(a)) is delayed to the next time bin (shown by the expanded red box in Fig. 6(b)), to resolve the imbalance at EB003. In comparison, *DCB-Enroute-Delay* delays the same flight using airborne delay instead of ground delay, as shown by the red box in Fig. 6(c). Fig. 7 shows this more clearly, where *DCB-Waypoint* assigns pre-departure delay to three aircraft, while *DCB-Enroute-Delay* assigns pre-departure delay to just one aircraft and two additional aircraft are delayed while airborne, so that they takeoff at the originally desired time.

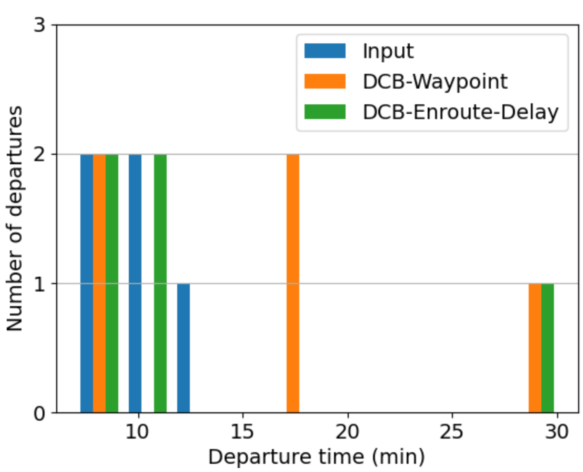


Fig. 7. Departure time distribution for traffic scenario in Table I.

C. *Simulating New DCB algorithms for the Dallas/Fort Worth Route Network*

The traffic scenario in Ref. [7] is comprised of forty flights operating over all ten routes shown in Fig. 5. The simulation setup is the same as that used in Ref. [7] simulations, with two crucial differences: all vertiports and constrained waypoints have a capacity of two operations per time bin and no temporal spacing requirement. In contrast, the waypoints had a higher capacity of four operations in each time bin in the simulations described in Ref. [7]. Reduced capacities of waypoints is necessary to force *DCB-Waypoint* and *DCB-Enroute-Delay* to act. Also, in Ref. [7], consecutive flights crossing any waypoint were required to be separated by two minutes. This requirement can impose additional delays on flights, as demonstrated in Ref. [7]. This requirement was removed for the following simulation because we wanted to focus on the effect of new DCB algorithms, which do not have their own separation requirement, on assigned delays in isolation from other airspace services.

Fig. 8 shows the heatmap of input traffic scenario for 40 flights, which means that 40 departures and 40 arrivals use the vertiport resources. The naming convention used in the heatmap is that the vertiports are named starting with "DF" and all others are waypoints. As can be seen from the heatmap, all vertiports and waypoints have demand-capacity imbalance in at least one time bin, i.e., have occupancy of more than 2 operations per 12 minutes.

Fig. 9 shows the resulting heatmap of observed traffic using *DCB-Vertiport* algorithm, i.e., the actual occupancy after the DCB service moderates demand, where the unresolved imbalances at waypoints EB002 (one in time bin "2" and two in time bin "5") and EB003 (one each in time bins "2" and "3"), are as expected. Figures 10 and 11 show the resulting heatmaps employing *DCB-Waypoint* algorithm and *DCB-Enroute-Delay* algorithm, respectively. Both algorithms resolved all demand-capacity imbalances. For the latter two algorithms, while all time bins are at or below their capacities, differences can be observed at the vertiports DF14 and DF32.

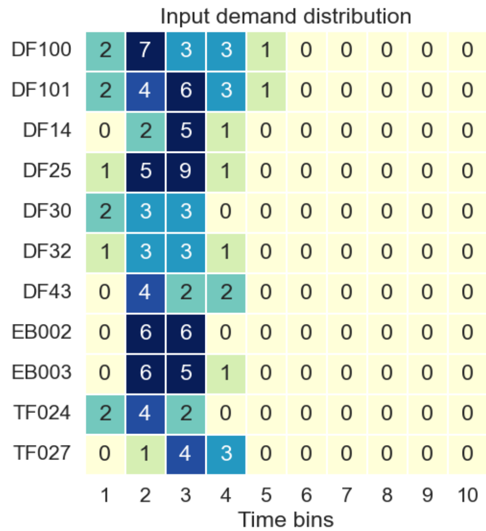


Fig. 8. Heatmap of input demand distribution.

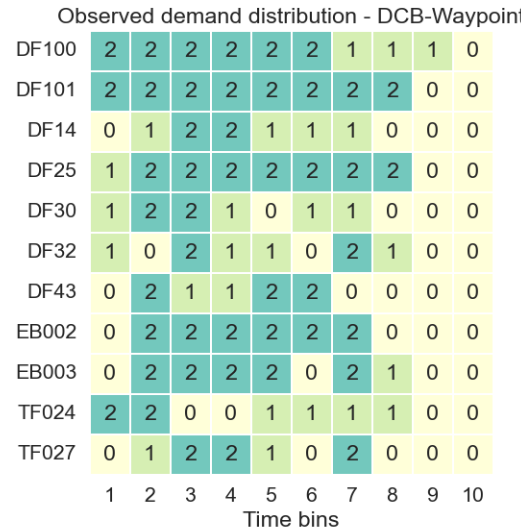


Fig. 10. Heatmap of post-DCB demand– *DCB-Waypoint*.

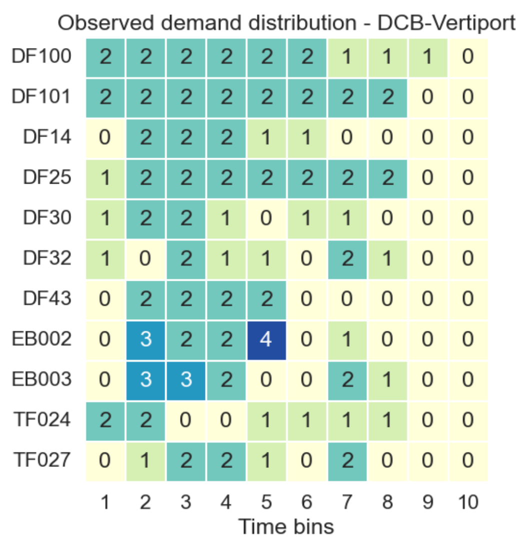


Fig. 9. Heatmap of post-DCB demand– *DCB-Vertiport*.

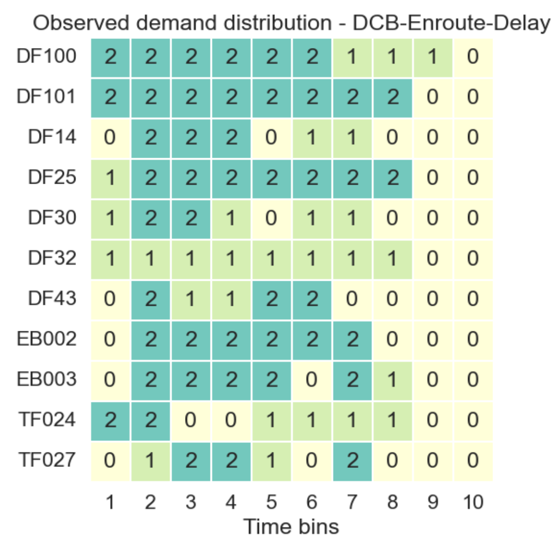


Fig. 11. Heatmap of post-DCB demand– *DCB-Enroute-Delay*.

The *DCB-Vertiport* algorithm delays 25 aircraft on the ground but leaves some waypoint demand-capacity imbalances unresolved. In total, this algorithm assigns 695 minutes of delay, with an average delay of 27.79 minutes per delayed flight. The *DCB-Waypoint* algorithm delays 28 aircraft, thereby resolving all predicted imbalances, including at waypoints. As a result, the total delay increases to 796 minutes, with the average delay of 28.44 minutes. The higher total delay is needed for moderating traffic to comply with the capacity constraints of all the resources in the network.

Table II shows the statistical differences between the results obtained from the three DCB algorithms. The benefit of *DCB-Enroute-Delay* algorithm is evident from the third column of Table II. Like *DCB-Vertiport*, *DCB-Enroute-Delay* delays only 25 aircraft on the ground, even though it resolves all imbalances, including those at waypoints. This algorithm also manages to lower the total ground delay to 747 minutes, although with a slightly higher mean delay of 29.86 minutes per flight.

There were eight instances of airborne delay assigned by this algorithm. Three flights were assigned no ground delay and their predicted imbalances were resolved by speed reduction command. Together, these three flights saved 35.37 minutes of pre-departure delay. The remaining five flights were delayed on the ground – due to imbalance at their origin vertiport – but were assigned additional airborne delay, which totaled 14.4 minutes.

Thus, a total of 49.77 minutes of pre-departure delay was reduced by the *DCB-Enroute-Delay* algorithm, as compared to the *DCB-Waypoint* algorithm.

TABLE II. TRAFFIC SCENARIO

Statistic	DCB-Vertiport	DCB-Waypoint	DCB-Enroute-Delay
Number of pre-departure delays	25	28	25
Total ground delay (min)	694.63	796.38	746.61
Mean delay (min)	27.79	28.44	29.86
Std. Deviation Delay (min)	15.69	15.09	14.74
Max delay (min)	58	58	58
Number unresolved imbalances	5	0	0
Number of resolution maneuvers	33	38	38

D. *Comparing DCB Algorithms in Presence of Departure Time Uncertainty*

In real world operations, airspace services will need to be robust to uncertainties. In this section, the results of the simulation of all three proposed DCB algorithms in the presence of departure time uncertainty are presented. At the vertiports, uncertainties inherent to activities involved in aircraft turnaround such as aircraft charging or refueling times, passenger boarding or de-boarding, etc. may cause the actual departure times may differ from the scheduled times. We conducted the following simulations to understand the effects of such uncertainties on system-level performance, especially in terms of number of aircraft delayed and the amount of delays assigned.

Using the same traffic scenario presented in Sec. IV.C, each of the three DCB algorithms were simulated with departure time deviations ranging from ±1 minute to ±5 minutes. The variations in the departure time from the scheduled time of the 40 flights in the scenario were randomly drawn with uniform distribution. Each combination of a DCB algorithm and departure time deviation was run 10 times for a total of 150 simulation runs (= 3 algorithms x 5 deviation cases x 10 runs/case). The results presented below are averages of all simulations for each combination.

Fig. 12 shows the average occupancy in the time bins for simulations with each of the three algorithms for all five cases of departure time deviation. The first observation is that even in the presence of uncertainty both *DCB-Waypoint* and *DCB-Enroute-Delay* perform better. While the desired occupancy was two operations / time bin, the occupancy exceed 10 operations / time bin for all five cases of departure time deviation for simulations with *DCB-Vertiport* while for the other two DCB algorithms, it ranged between 6 – 9 operations / time bin. When the demand-capacity balance is ensured at waypoints in addition to vertiports, there is added spacing between flights due to additional delays assigned and this leads to the lower occupancy observed in Fig. 12.

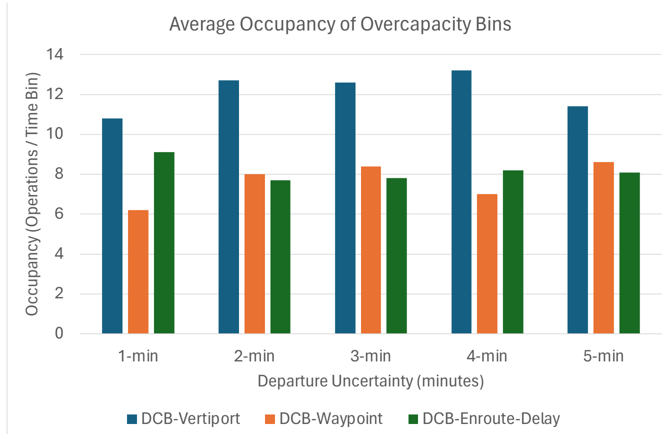


Fig. 12. Average occupancy in time bins in presence of uncertainty.

However, the trend between *DCB-Waypoint* and *DCB-Enroute-Delay* is not clear, even though *DCB-Enroute-Delay* leads to lower occupancy in three of the five cases. This is because of DCB resolutions are strategically applied once when the flights are proposed, whereas the uncertainty modeled in this study occurs at actual departure time. If the DCB services could be run even after takeoff, we can expect to see lower occupancy.

V. DISCUSSION AND RECOMMENDATIONS FOR DCB SERVICES

The main advantage of the two new DCB algorithms presented in this paper over the previous DCB algorithm in Ref. [4] is that they managed the balance of demand versus capacity at waypoints in addition to the vertiports. Quantitative analysis of managing demand at waypoints showed reduced over-utilization of airspace resources and, with *DCB-Enroute-Delay*, reduced pre-departure delay. The latter provides throughput and efficiency benefits by increased utilization of vertiports.

The authors in Ref. [4] did a comparative study of the effect of varying time bin size on the amount of assigned delays, and found that smaller time bins lead to more delays. However, using a longer time bin size reduced frequency of tactical separation required and did not account for periods of high traffic density at specific locations and times. At present, there is no industry consensus or standard for the length of a time bin for defining resource capacities. There needs to be a more rigorous study to fill this gap. Specifically, decisions need to be made about the length of time bins and whether these lengths remain the same across the entire network at all times. Time bins can also be defined to be "rolling" which means that the start and end times could be dynamically changed. For these modifications, new DCB algorithms will be required that account for changes in the definition of time-bound capacities. DCB services will need to work along with a "Capacity Information Service" that will dynamically determine capacities of airspace resources as opposed to the fixed values used in this paper.

An even more advanced DCB algorithm may issue additional maneuvers in addition to speed reduction. Assigning only delays leads to inefficiencies in the system in form of underutilized resource capacities. To ensure that the demand does not exceed capacity, throughput needs to be lowered by changing the time of arrival at the capacity constrained resource, i.e., successive arrivals need to be temporally spaced. Underutilization can be expected with excessive delays although it may improve safety of operations. Further, uncertainties such as those due to departure delays and weather can also cause underutilization. Hence, only the needed amount of delay should be applied to balance between safety and resource utilization. A balance between ground and airborne delay needs to be done to prevent underutilization, where, in a dynamic airspace, airborne delay may not be needed if capacity becomes available.

Yet another command can be a speed-up command to an aircraft that prepones the arrival time at constrained waypoints. The logic of such a DCB algorithm can build on that of the *DCB-Enroute-Delay* algorithm. Whereas, *DCB-Enroute-Delay* attempts to resolve the imbalance by assigning a delay at the waypoint under consideration, failing which it assigns pre-departure delay, this new algorithm will attempt resolve predicted imbalances by first issuing a speed reduction command, failing which it will issue a speed increase command, and, if that fails too, will assign pre-departure delay. This new DCB algorithm may lead to even fewer pre-departure delays since it has an additional degree of freedom to assign maneuvers. However, issuing speed up commands to aircraft may lead to other issues such as limits on aircraft performance and increase energy consumption, which will need to be considered. A study that simulates the DCB algorithm with the able to issue a speed-up command against the *DCB-Enroute-Delay* algorithm will test whether using speed increase as a resolution mechanism provides any benefits.

In summary, there are many potential advancements to the DCB logic that can be investigated in the future. DCB services need better integration with other airspace services thereby enabling better handling of resource capacity and demand information, especially when the values for capacities could be dynamically changing. Another avenue for advanced DCB algorithms is incorporating optimization within the imbalance resolution logic for improving capacity utilization and reducing delays. All of the presented DCB algorithms were restricted to considering one flight at a time. Better delay distribution may be obtained if multiple aircraft are deconflicted simultaneously. Considering multiple aircraft at a time can also help address the question of fairness in delay assignment among multiple aircraft and operators.

VI. Conclusions and Future Work

This paper proposes a set of new demand-capacity balancing algorithms for application to UAM operations. Building on a DCB algorithm that resolved predicted imbalances at vertiports only using pre-departure delay as the sole resolution mechanism, two new algorithms proposed extend the capabilities by resolving imbalances at waypoints in addition to vertiports and by assigning airborne delays with speed changes. These algorithms are expected to provide benefits of reduced pre-departure delays along with increased throughput and operational efficiency.

The proposed algorithms were implemented in a new fast-time simulation tool developed for simulating UAM operations. The algorithms were verified using a test scenario and then used in simulations using common traffic scenarios to exercise their capabilities and identify the potential benefits that they can provide. The algorithms were found to provide reduced ground delays even in the presence of departure time uncertainties. Although the presented DCB algorithms, which are heuristic in nature, provided benefits, additional research is needed to provide optimization capabilities and better integrate with existing and new airspace services.

References

[1] Federal Aviation Administration (FAA), "Urban Air Mobility: Concept of Operations v2.0," version 2.0, April 2023.

[2] A. Cheng, K. E. Witzberger, D. R. Isaacson, S. A. Verma, H. Arneson, H. Lee, C. H. Seah, I. Levitt, "National Campaign (NC)-1 Strategic Conflict Management Simulation (X4) Final Report," NASA Technical Memorandum, NASA/TM-2022-0018159, December 2022.

[3] International Civil Aviation Organization (ICAO), "Global Air Traffic Management Operational Concept," ICAO, Doc 9854, First Edition, 2005.

[4] H. Lee, K. A. Moolchandani, and H. Arneson, "Demand and Capacity Balancing at Vertiports for Initial Strategic Conflict Management of Urban Air Mobility Operations," 41st Digital Avionics Systems Conference (DASC), Portsmouth, VA, September 2022.

[5] H. Lee, "NASA's Simulation Activities for Evaluating UAM Concept of Operations," HorizonUAM Symposium 2021, September 2021.

[6] K. Moolchandani, "UAM Simulation Tool for Airspace services Research: A Fast-time Simulation Tool for Urban Air Mobility Operations," unpublished, in writing.

[7] H. Lee, A. Lee, C. H. Seah, C. H. Lee, and K. A. Moolchandani, "Provider of Services for Urban Air Mobility (PSU) Prototype Simulation (X5) Report," NASA Technical Memorandum, in writing.