A Simulation Study of Bin-and-Sort Policies in a Distributed System for Flights Scheduling

Alexander V. Sadovsky* and Robert D. Windhorst†
NASA Ames Research Center, Aviation Systems Division, Branch AFH, Moffett Field, CA 94035-0001

James D. Phillips‡
University Space Research Association, Moffett Field, CA 94035-0001

Yung-Cheng Chu§
University Space Research Association, Moffett Field, CA 94035-0001

This is a report on the results of a simulation experiment conducted on an automated distributed system for scheduling flights in a shared airspace. The scheduling is carried out by computer systems owned by the private operators of the flights, which interface through a centralized independent server. The scheduling requires determining which of the group of flights at hand (whether owned by the same operator or different ones) are to have the advantage of being scheduled before the others. A chosen method of grouping and prioritizing flights must: have the ability to schedule a flight promptly, prevent scheduling conflicts between pairs of overlapping flight routes, and be computationally feasible. Scheduling flights one by one is the First Come First Served policy. Scheduling too many at a time is infeasible. The type of scheduling policy proposed herein, called *bin-and-sort*, is an attempt to reach a compromise between these two extremes. Bin-and-sort allows for each operator to choose its criteria for prioritizing its flights, and for the centralized component to use different arbitration criteria. A number of these criteria are used in the simulation presented herein. The results of each criterion choice are reported and analyzed.

I. Introduction

Arrivals of commercial flights in the current National Airspace System (NAS) are scheduled by a centralized system, Time Based Flow Management (TBFM), used by the Federal Aviation Administration (FAA). TBFM assigns each inbound flight a scheduled time of arrival at its arrival meter fix , which is located in the vicinity of the airport and on the boundary of the Center and TRACON airspaces. For each flight, the scheduled time of arrival and a speed advisory are displayed to the air traffic controllers responsible for the airspaces upstream of the arrival meter fix. The controllers regulate the flights so that they cross their arrival fixes at times as close as possible to their scheduled times of arrival.

TBFM is a centralized closed system. Under this system, airlines have few, if any, ways to influence the scheduled times of arrival that are assigned to their flights. Such a system has a number of disadvantages. One is the risk of single point of failure. Another is that the centralized nature of the system limits scalability if there is a need to add more users; e.g., if there are new entrants to the aviation market. Furthermore, the stakeholders (airlines) are left with little influence the scheduled times of arrival. Some such influence is enabled through Collaborative Trajectory Options Programs (CTOP) [5, 6], which, in particular, allow an airline to provide reroute options for its flights and flight swapping. CTOP, however, rely on a centralized system to execute the scheduling algorithms and require an airline to disclose information that it may be reluctant to disclose.

To address these and other issues, future systems being proposed for managing air vehicles are distributed systems, offering operators influence to schedule and negotiate resources for their flights, and having each operator use or implement the software paid for by its private owner. One example of such a system, designed for managing Unmanned Aerial Systems (UAS), is the UAS Traffic Management (UTM) [2].

*Aerospace Engineer, Aviation Systems Division, Mail Stop 210-6, alexander.v.sadovsky@nasa.gov.
†Aerospace Engineer, Aviation Systems Division, Mail Stop 210-6.
‡Software Engineer.
§Software Engineer.

The alternative to TBFM assumed here is a distributed system (DS), analogous to that of UTM. A DS consists of one central unit called Resource Schedule (RS), owned by an party external to the airlines (e.g., by the government), and multiple units, each termed a Flight Planning and Scheduling Service (FPSS), owned by a private fleet operator and called upon to schedule flights for that airline. (Here and in what follows, a flight is said to be "owned" by an operator or by an FPSS, even though an operator may be serving multiple airlines. This is different from the general colloquial understanding that a flight is owned by the airline.). A DS is intended for scheduling flights pre-departure (as opposed to modifying a flight plan that is already being executed). The specific type of a DS used here is the Collaborative Seamless Management of Airspace Resources and Traffic (CSMART); see Ref. [8]. Thus, under CSMART, the FPSS make no direct disclosures to each other, communicating instead through the RS. Each FPSS carries the computational load of computing its own schedules.

Past research introducing CSMART demonstrated the capability of the system using only one FPSS. This paper expands upon this prior research by implementing and exploring arbitration between two FPSSs. The main content of this paper is a report on and analysis of the results of a simulation experiment conducted with an instance of CSMART that has two FPSS. The following two paragraphs describe this instance and the supplementary software and data in more detail.

The RS serves to keep a record, for each resource, of the time windows when this resource is unavailable, and to enable the FPSS schedule their flights consistently with these evolving availability constraints. An overview of the problematics of scheduling arrivals and the rationale for and definition of the bin-and-sort policy*, used in this simulation, are given in Sec. II.

The two FPSS own two mutually complementary subsets of a pre-selected set of 54 flights, fixed throughout the simulation. Each FPSS acts in its own business interests, ranking by priority (aka *prioritizing*) the flights that it owns and knowing nothing of the flights owned by the other FPSS. When two flights from two different FPSS contend for a departure time slot, the system uses the RS to resolve the contention by what is termed here *arbitration*. Thus, the FPSS prioritize, and the RS provides the arbitration. A detailed specification of the arrival operation scenario used in this simulation is given in Sec. III.

The simulation experiment consists of 17 simulation runs. The last of these is not a DS: it involves no FPSS and has all 54 flights owned by a centralized entity. In most runs of the simulation, the DS was used together with another software system, Autoresolver Simulator (ARSim) (see, e.g., Ref. [3] and references therein), whose role is to simulate the execution of the flights and the interaction between pilots and Air Traffic Controllers (ATC). The first 16 runs differ in: the criteria of intra-FPSS prioritization and of inter-FPSS arbitration. The details of the individual runs are given in Table 1 in Sec. IV.

The two metrics used to study the experimental results are *departure delay*–i.e., the difference between a flight's, (a), actual and, (b), scheduled times of departure–and *airborne delay*–i.e., the difference between, (a), the flight's duration, from takeoff to touchdown, according to the schedule and, (b), the flight's actual duration. Their detailed definitions are given in Sec. VI.A.

The key questions studied in this paper are as follows:

- Q1 How do different choices of prioritization affect the statistics of departure delay?
- Q2 How do different choices of prioritization affect the statistics of airborne delay?
- Q3 Does centralized ownership and scheduling of all flights do better to reduce the departure delay among the flights than does the distributed system with multiple FPSS?
- Q4 Does centralized ownership and scheduling of all flights do better to reduce the airborne delay among the flights than does the distributed system with multiple FPSS?

The results of the experiment are presented and analyzed in Sec. VI.

II. Bin-and-Sort Policies of Flight Scheduling

As in Refs. [4, 7, 8], a *flight schedule* here refers to a sequence of the format

(departure resource, time window), (resource 2, time window), ..., (arrival resource, time window);

for each resource needed by the flight, a time window is specified for the flight's use of that resource.

*This policy bears no relation to the *bin sort* algorithm taught in courses on algorithms; e.g., in [1], section 8.5.

A. Considerations From First Principles

With the scheduling of flights being carried out by CSMART (described in Sec. I), the flights in need of scheduling emerge asynchronously and at times that the RS cannot easily predict. Furthermore, if a flight is to be scheduled, its schedule must not conflict with the already existing flight schedules. The constraints a flight schedule must meet fall into two broad categories:

- The constraints involving one flight only; e.g., the flight's performance envelope or a resource being out of service.
- The constraints involving pairs of flights; e.g., the time intervals in which a needed resource is reserved for use by a previously scheduled flight.

These constraints are described in detail in Refs. [7, 8].

The RS must decide when to let each flight's FPSS schedule that flight. If two flights emerge, in some order, in need of scheduling, they are not necessarily to be scheduled in that order. The "order of need" (i.e., the "first-come" order) must generally allow overriding by an order defined by some other criteria. Therefore, to assign such an order to flights, one must evaluate the flights against the criteria. This implies that the high-level approach to scheduling must include each of the following steps:

1) Gathering a number of flights into a group.
2) Ordering the flights in the group by some set of criteria.
3) Letting the FPSS of each flight in the group, in the order instituted in the previous step, schedule that flight without conflicting with any of the already scheduled flights.

With this approach, a flight's actual time of departure or arrival may well be later than the corresponding preferred time.

One extreme special case is the group having size one. This would result in scheduling the flights in the First Come First Serve (FCFS) order. The other extreme case would be a group of infeasibly many (for processing) flights. It follows that if flights continue to enter the distributed system at unpredictable times and need scheduling, and that if they are to be processed in groups, then each group needs to have feasible size and to be ordered by suitable criteria. The next Section introduces a type of scheduling policy that processes flight scheduling requests in groups, hence generally differs from FCFS.

B. Bin-and-Sort Policies

The type of scheduling policy proposed in this Section will henceforth be called *bin-and-sort policies*. It fits into the general category of policies described in Sec. II.A and proceeds in repeating, nonverlapping cycles. Each cycle consists of three steps that can be thought of as a specific implementation of those described in Sec. II.A.

A bin-and-sort policy relies on the concepts termed here *priority* and *arbitration*, defined as follows. Each FPSS is free to order the flights it owns by its own criteria. This set of criteria for intra-FPSS flight ordering is the FPSS-specific *priority*. It is possible that two different FPSS each request to schedule a flight, and the two flights are of equal intra-FPSS priority and request schedules that conflict. In this case, the decision as to which of these two flights to schedule first, rests with the RS and is called *arbitration*.

The FPSS-specific definitions of priority, and a pre-defined set of arbitration criteria for the RS, together determine a *bin-and-sort policy*, which carries out the aforementioned steps 1-3 as follows:

1′. The RS allocates an empty list, called the *bin*, to be populated with flights. The RS allocates a time window and set a maximal bin size. Each flight entering the system during this time window is added to the bin. Once the window ends or the maximum number of flights is reached, proceed to the next step.
2′. The RS uses arbitration to order the flights that contend for a turn to be scheduled.[†]
3′. In the obtained order, have the RS contact the FPSS of each flight in the bin:
   (a) Give the FPSS all the time windows of availability at each of the resources needed by the flight.
   (b) Let the FPSS compute and send to the RS a schedule for the flight.
   (c) Verify that the schedule does not conflict with any of the previously scheduled flights.

Difference choices of criteria for priority and arbitration lead to different bin-and-sort policies.

We note that there are two orderings of flights involved here: the order in which the FPSS requests to schedule flights are received by the RS and added to the bin, and the order in which those flight schedule requests end up being processed. Priority and arbitration refer to the latter ordering. Thus, in each cycle of the bin-and-sort policy, the bin is processed as follows:

[†]The resulting order of the flights in the bin preserves the relative priority of every two flights owned by the same FPSS. However, if two flights, 1 and 2, are owned by the same FPSS and are consecutive in the corresponding priority, it is possible that another FPSS flight is scheduled after flight 1 and before flight 2.

- Schedule all top priority flights, one from each FPSS involved. If two or more flights request conflicting schedules, then use arbitration to decide which of these get scheduled before the others.
- Schedule all second priority flights, one from each FPSS involved. If two or more flights request conflicting schedules, then use arbitration to decide which of these get scheduled before the others.

  ⋮

An Example of a Bin-and-Sort Policy

Suppose there are only two FPSS, labeled 1 and 2, and:

- FPSS 1 owns flights 1.A, 1.B, 1.C, listed here in the internal priority order of FPSS 1;
- FPSS 2 owns flights 2.A, 2.B, 2.C, 2.D, listed here in the internal priority order of FPSS 2;
- The arbitration criteria of the RS are as follows: 1.A < 2.A, 1.B > 2.B, and 1.C < 2.C.

Assume the maximal bin size is 7 or more flights. Since each FPSS will submit its flights to the bin in the order of that FPSS internal priority, the resulting order in the bin of the aforelisted 7 flights, if they are all submitted within the allotted time window, will be: 2.A, 1.A, 1.B, 2.B, 2.C, 1.C, 2.D. This is the order in which the flights will be scheduled. One can similarly construct examples with more that 2 FPSS.

C. The Flights Scheduled Earlier Under Bin-and-Sort Get an Advantage

The later in the bin a flight is scheduled, the less time window availability it is likely to have at its resources. This is because the previously scheduled flights, if they need any of those resources, have already had their time windows reserved at those resources.

D. The FPSS with Smaller Fleets Under Bin-and-Sort Get an Advantage

It follows from Sec. II.B that if a bin contains N flights from one FPSS, and $M > N$ flights from another, then the lowest-priority $(M - N)$ flights from the latter FPSS will not begin to be scheduled until all flights of the former FPSS are scheduled. In the example at the end of Sec. II.B, we have $N = 3$ and $M = 4$. Indeed, the lowest-priority flight of FPSS 2–namely, flight 2.D–is not scheduled until all flights of FPSS 1 are.

III. The Scenario Used in the Experiment

A. The Totality of All Flights Used

The scenario is closely based on historical data for the following set of arrival flights:

- total number of arrival flights: 54 (in 90 minutes);
- arrival airport: EWR;
- landing runway: 54 flights landed on 22L;
- actual traffic date and landing time: 2018-04-26, 18:30 - 20:00 UTC;
- the departure times were adjusted to attain increased demand on the landing runway (see Fig. 1) so as to stress the system by raising the demand on the runway above the capacity, which is limited by the time separation requirement of 75 seconds;
- the list of the 54 arrival flights is split into two parts, each owned by an operator.

As part of the initial settings of the simulation, this list of flights has been partitioned into two subsets, each to be handled by its own FPSS. The key desired characteristic of such a partition is “maximal stress on the system”; i.e., to result in the maximal possible load on the FPSS to compute the schedule and to coordinate with the RS. Stressing the system by having demand exceed the capacity is necessary for manifesting the capabilities of CSMART. Consistently with these considerations, the partition was chosen by the following method, using a quantity we call *arrival competition*.

B. The Method of Partitioning the Set of 54 Flights into Two Subsets, One Per FPSS

The Estimated Times of Arrival (ETA) for all 54 flights are fixed. Each of the FPSS will be serving multiple airlines, but each airline’s flights will be owned by only one FPSS; this was ensured by using the actual callsigns of the historical flights that served as the basis for the scenario.

For each possible partition of the 54 flight set into two subsets F_1, F_2, introduce the following quantities and notation:

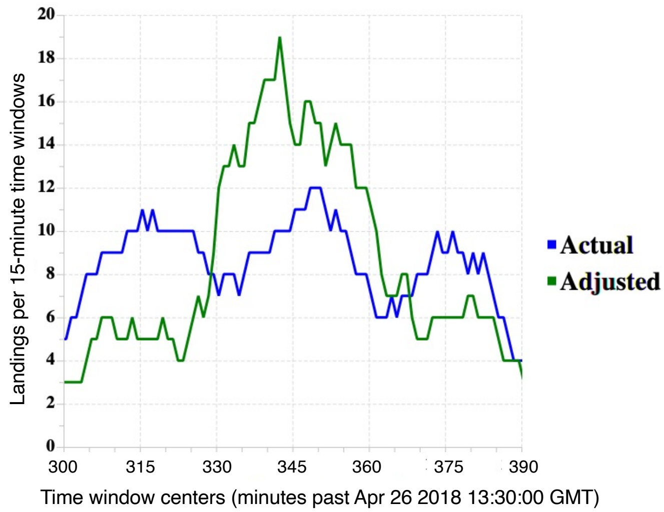

Fig. 1 Actual and adjusted arrival rates at EWR.

- The mean μ_1 and standard deviation σ_1 of the Estimated Times of Arrival (ETA) of the flights in F_1. Let A_1 denote the interval $[\mu_1 - \sigma_1, \mu_1 + \sigma_1]$.
- The mean μ_2 and standard deviation σ_2 of the Estimated Times of Arrival (ETA) of the flights in F_2. Let A_2 denote the interval $[\mu_2 - \sigma_2, \mu_2 + \sigma_2]$.

Only those partitions were considered where each airline's flights were handled by only one of the FPSS. In other words, an airline was not allowed to have some of its flights handled by one of FPSS, and its other flights by the other FPSS.

Let us define the quantity of *arrival competition* for the partition (F_1, F_2) by:

$$\text{arrival competition for } (F_1, F_2) = \frac{\text{length of } (A_1 \cap A_2)}{\text{length of } (A_1 \cup A_2)}. \tag{1}$$

The practical intent behind this definition can be seen from examining the following two extreme cases:

1) If the ETAs of the flights in F_1 are far from those in F_2, then the two subsets do not compete with each other's operations. This results in minimal computational load on the system (the opposite of what is wanted here). Quantitatively, this results from the length of $(A_1 \cap A_2)$ (which is the numerator in Eq. (1)) being close to zero, the smallest possible arrival competition.
2) If the ETAs of the flights in F_1 and in F_2 are close (fill the same time interval), then the two subsets substantially interfere with each other's operations. This results in the desired high computational load on the system. Quantitatively, this results from $(A_1 \cap A_2)$ being (nearly) equal to each of A_1, A_2, hence to $(A_1 \cup A_2)$, resulting in the maximal arrival competition of (nearly) 1.

The partition F_1, F_2 used in this experiment was the one attaining the highest arrival competition value. This value, rounded to the 2nd place after the decimal point, turned out to be 0.95. In this partition, one FPSS ended up owning 31 flights; the other FPSS, 23 flights.

IV. The Experiment

The experiment consisted of simulating the above scenario under a number of different bin-and-sort policies, corresponding to different choices of criteria for priority and arbitration. The simulation runs corresponding to these policies are listed in Table 1. In each run except 1, CSMART was run to compute the flight schedules, and ARSim was run to simulate the execution of the scheduled flights. In run 1, an older version of the code was used and, while there were two FPSS, in effect their flights were combined and, in totality, ordered by route length. Thus, effectively, the result of the scheduling was identical to what it would have been with one FPSS handling all 54 flights.

To get insight into the effect of the prioritization criteria on the metrics defined in Sec. VI.A, and to reflect the fact that an airline's business objectives are likely proprietary and inaccessible, at least in full, to outside parties, each FPSS in runs 2 through 6 uses a prioritization that is a random permutation of its flights. In intuitive language, the random

Table 1 The experiment matrix used in this work.

Run	With ARSim?	With CSMART?	Prioritization scheme for FPSS 1	Prioritization scheme for FPSS 2	Arbitration
1	No	Yes	N/A	N/A	Flight route length
2-6	Yes	Yes	A new random permutation for each of the 5 runs.	A new random permutation for each of the 5 runs.	Flight route length
7-11	Yes	Yes	The same flight orderings were used for runs 7 through 11 as for runs 2 through 6, respectively.	The same flight orderings were used for runs 7 through 11 as for runs 2 through 6, respectively.	Earliest scheduled departure time
12-16	Yes	Yes	The same flight orderings were used for runs 12 through 16 as for runs 2 through 6, respectively.	The same flight orderings were used for runs 12 through 16 as for runs 2 through 6, respectively.	Earliest estimated landing time
17	Yes	Yes	N/A	N/A	Earliest estimated landing time

permutations simulate the fact that, to an outside observer, an FPSS way of prioritizing flights may look like a random shuffle.

For the sequence of the five runs 7 through 11, as well as for 12 through 16, each FPSS uses the same five random permutations of the flights as it used for runs 2 through 5. The last row (run 17) corresponds to a centralized system that owns and schedules all of the flights.

V. A brief summary of the implementation of the distributed system for arrival scheduling

To conduct the experiment, the distributed system described above has been implemented in Python 3, as follows. RS and FPSS have each been implemented as a class. A simulation uses one instance of RS and multiple instances of FPSS. Step 3′, described at the end of Sec. II, is carried out via the a protocol of message exchange between the RS and each FPSS.

The instances of class FPSS (representing individual operators of flights) "enter the system" at randomized times, to simulate the functionality of a real system, where flight operators, being autonomous federates, contact the RS asynchronously and attempt to work with the RS to schedule the flights under their ownership.

VI. Analysis of Experimental Results

The next three sections, Secs. VI.A, VI.B, and VI.C, give, respectively, the metrics used to analyze the results of the experiment, the statistics computed for the experimental results and figures illustrating the data for some of the runs. An analysis of all these is offered in Sec. VI.D.

A. The Metrics Used

In what follows, the Scheduled Time of Arrival (STA) is that selected by the FPSS for a particular resource. It must satisfy the spacing constraint imposed at the resource. The FPSS were designed to compute the STA using Meyn's algorithm [4], which results in STA as close as possible to the ETA; i.e., the STA that minimize delay. However, as Meyn's algorithm computes the STA in compliance with the transit time constraints, the resulting STA may not take the next open interval on the resource timeline.

- Departure Delay = (Actual - Scheduled) Departure Time
- Touch Down Delay = (Actual time of touch down) - (STA at arrival runway)
- Airborne Delay = (Touch Down Delay) - (Departure Delay)

B. The Statistics

The statistics calculated for the results of the experiment, listed in Table 2, include the means and standard deviations of delay by the run and, for runs 1-16, by the FPSS.

C. A Description of the Figures

- Figs. 2 through 6 for scatter plots of departure delay for runs 1, 2, 7, 12, 17. Whenever the horizontal axis variable is a time variable (all runs except 2 through 6), the units of time are **minutes past Thu Apr 26 2018 13:30:00 GMT**. For runs 1, 2, 7, 12, the data points corresponding to FPSS 1 are denoted by marker +, colored blue; those corresponding to FPSS 2, by marker ×, colored brown. For run 17, the data points for all 54 flights are denoted by marker +. In all of the above Figures, the size of the marker is proportional to the flight's priority. Shown at the top of each of these Figures are:
  - The total data set (all 54 flights) correlation coefficient, corresponding to the two variables plotted.
  - The average, μ, and standard deviation, σ, of the departure delays incurred by the flights under each FPSS, and the correlation between departure delay and priority (`del.-pri.  corr.`) in runs 1 through 16.
  - The average, μ, and standard deviation, σ, of the departure delays, and the correlation between departure delay and priority (`del.-pri.  corr.`) for all 54 flights in run 17.
- Figs. 7, 8 show the bar-and-whisker plots for departure delay statistics in runs 1, 2, 7, 12. The corresponding statistics for run 17 are shown in Fig. 9.
- Figs. 10, 11 show the bar-and-whisker plots for airborne delay statistics in runs 1, 2, 7, 12. The corresponding statistics for run 17 are shown in Fig. 12.

D. Observations and Insights

Tables 3 and 4 characterize qualitatively the overall behavior of the statistics computed for the results of the experiment. A list of more detailed observations follows.

In what follows, the references Q1, Q2, Q3, Q4 are to the questions listed and so labeled at the end of Sec. I. Each of the following observations is accompanied by a reference to one or more of these questions.

1. Without preferential treatment based on the number of flights owned, the FPSS with more flights take more departure delay (question Q1)

In the scatter plots corresponding to runs 1 through 16 (of these, runs 1, 2, 7, 12, 17 are shown in Figs. 2 through 5), the flights owned by the FPSS with more (31) flights have clearly taken more departure delay than those owned by the other FPSS. To see why this is expected, apply the considerations described in Sec. II.D: here we have $M = 31$ and $N = 23$. Thus, the lowest-priority $(31 - 23) = 8$ flights of FPSS 1 are the last ones to be scheduled, hence will have less choice as to the time slots for using resources, hence will take more delay.

2. In run 1 (Table 1), the criterion of arbitration is highly correlated with the departure delay (question Q1)

This is indicated by the correlation coefficient, -0.44, shown in Fig. 2.

3. With two FPSS, the choice of criteria for prioritization and arbitration seems to have little effect on the departure and airborne delay statistics (questions Q1, Q2)

The departure delay and airborne delay statistics for runs 1 through 16 of departure delay and airborne delay are similar (see the four rightmost columns in Table 2, as well as Fig. 7). The similarity is even more pronounced by comparing the two rows in the Table in each of the following pairs: runs 2 and 7, runs 3 and 8, runs 4 and 9, etc.. Fig. 8) for departure delay, and Fig. 11) for airborne delay, each shows similar and and medians (the 2nd quartile).

4. The negative minima in the airborne delay box-and-whisker plots result from ARSim shortening a flight's transit time (question Q2)

In each of the runs 1 through 17, at least one flight had a negative airborne delay, an indication that the flight's actual transit time from origin to destination turned out shorter than that originally estimated. (Figs. 10 through 12 illustrate this for the runs shown.) This is expected: ARSim has the capability to shorten a flight's transit time by modifying the trajectory.

Table 2 Some of the statistics of the experimental results. All the means and standard deviations are given in minutes.

	FPSS 1, departure delay statistics			FPSS 2, departure delay statistics			All FPSS, departure delay statistics			All FPSS, airborne delay statistics	
Run no.	mean	st.dev.	correlation with priority	mean	st.dev.	correlation with priority	correlation with the arbitration criterion	mean	st.dev.	mean	st.dev.
1	12.49	17.46	0.71	3.27	5.37	0.59	-0.44	8.56	14.43	0.67	0.97
2	13.08	17.81	0.74	5.83	12.08	0.55	-0.12	9.99	16.03	0.22	0.53
3	13.91	20.66	0.76	3.42	8.94	0.43	0.15	9.44	17.49	0.35	0.69
4	11.43	16.39	0.73	3.75	7.64	0.52	-0.02	8.15	13.91	0.73	1.07
5	12.84	18.78	0.65	3.66	10.07	0.44	0.05	8.93	16.32	0.33	0.95
6	13.85	20.59	0.78	5.22	12.42	0.55	0.3	10.17	18.09	0.33	0.65
7	12.77	17.88	0.71	5.66	11.34	0.53	0.24	9.75	15.83	0.26	0.68
8	13.91	20.66	0.76	3.42	8.94	0.43	0.06	9.44	17.49	0.35	0.69
9	11.43	16.39	0.73	3.75	7.64	0.52	0.19	8.15	13.91	0.73	1.07
10	14.63	19.92	0.66	3.8	10.05	0.44	0.1	10.02	17.3	0.48	0.9
11	14.51	21.44	0.78	5.74	13.25	0.57	-0.07	10.77	18.91	0.54	0.8
12	12.99	18.0	0.04	6.01	12.12	0.09	-0.04	10.02	16.14	0.21	0.53
13	13.91	20.66	-0.15	3.42	8.94	0.33	-0.07	9.44	17.49	0.35	0.69
14	11.43	16.39	0.13	3.75	7.64	-0.17	0.01	8.15	13.91	0.73	1.07
15	14.56	19.58	0.13	4.04	11.04	0.12	-0.06	10.08	17.29	0.46	0.83
16	14.47	21.4	-0.16	5.8	13.42	0.28	-0.09	10.77	18.92	0.44	0.76

Run 17:

departure delay statistics	
mean	4.86
st.dev.	4.05
correlation with priority	0.44
correlation with the arbitration criterion	0.31
airborne delay statistics	
mean	1.66
st.dev.	1.9

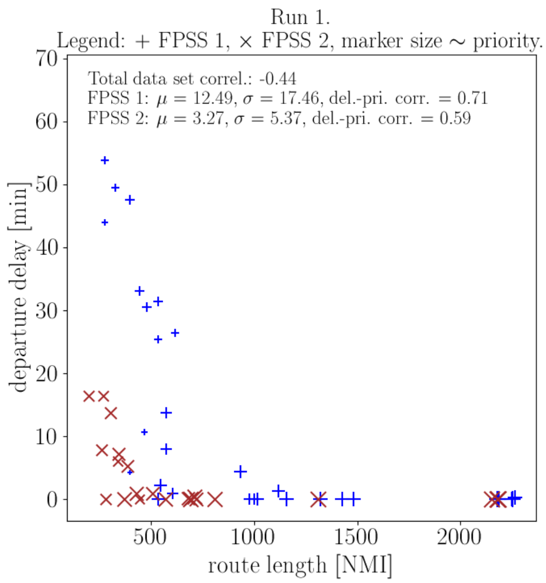

Fig. 2 A scatter plot of departure delay vs. earliest scheduled departure time, run 1. For each FPSS, departure delay is strongly correlated with priority.

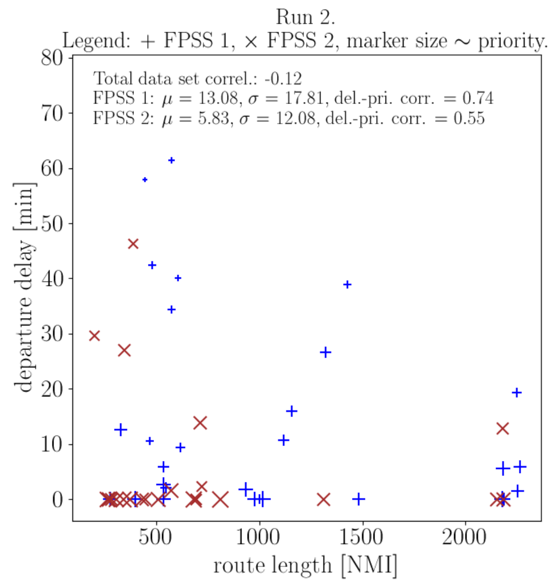

Fig. 3 A scatter plot of departure delay vs. route length, run 2. For each FPSS, departure delay is strongly correlated with priority.

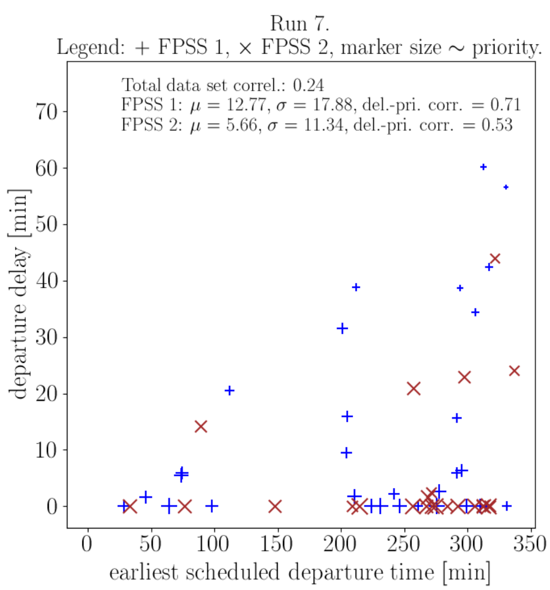

Fig. 4 A scatter plot of departure delay vs. earliest scheduled departure time, run 7. For each FPSS, departure delay is strongly correlated with priority.

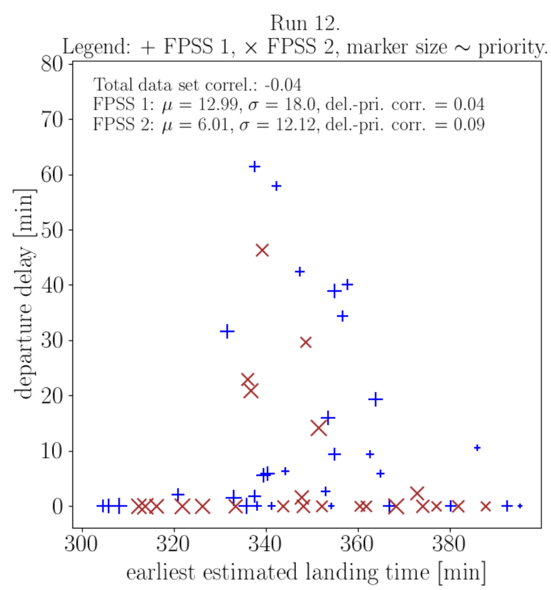

Fig. 5 A scatter plot of departure delay vs. earliest estimated landing time, run 12. For each FPSS, departure delay is weakly correlated with priority.

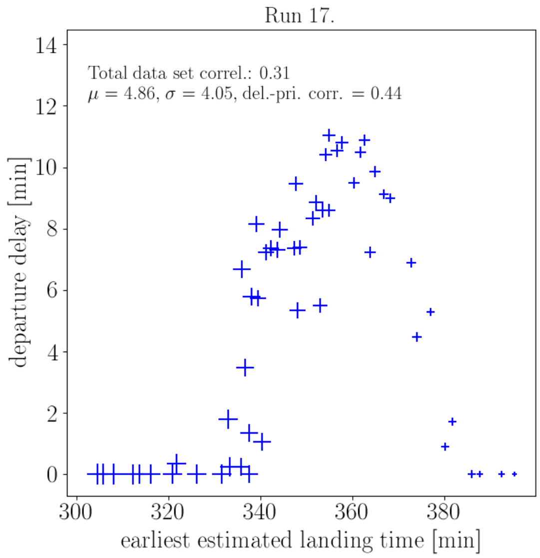

Fig. 6 Scatter plot of departure delay vs. earliest estimated landing time, run 17.

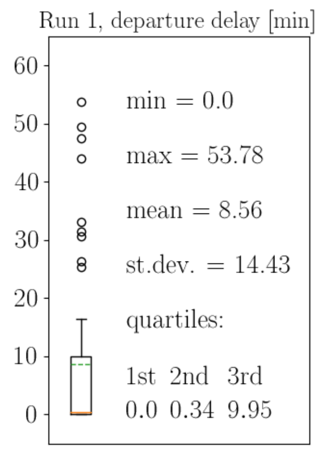

Fig. 7 Departure delay statistics: run 1.

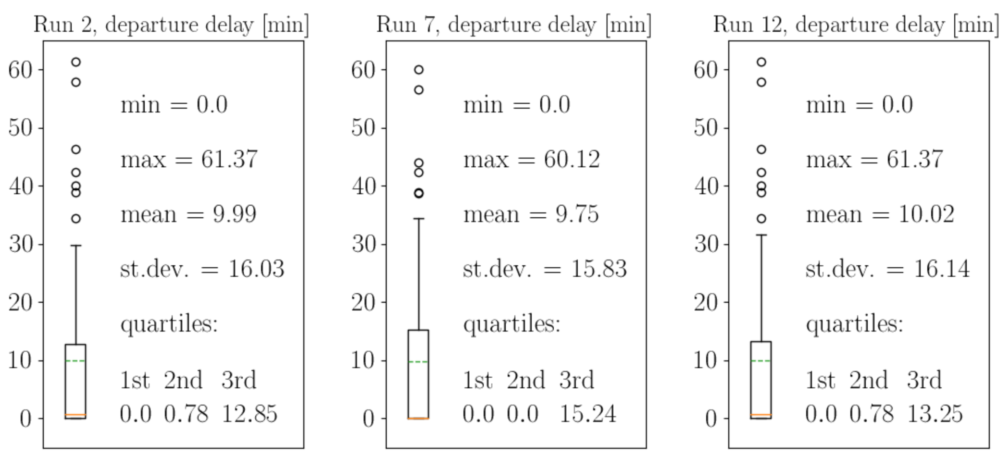

Fig. 8 Departure delay statistics: runs 2, 7, 12.

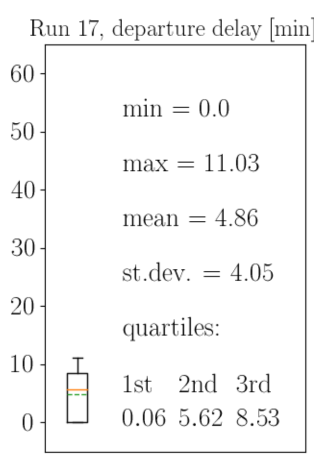

Fig. 9 Departure delay statistics: run 17.

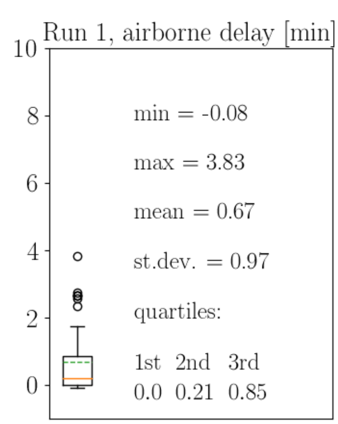

Fig. 10 Airborne delay descriptive statistics: run 1.

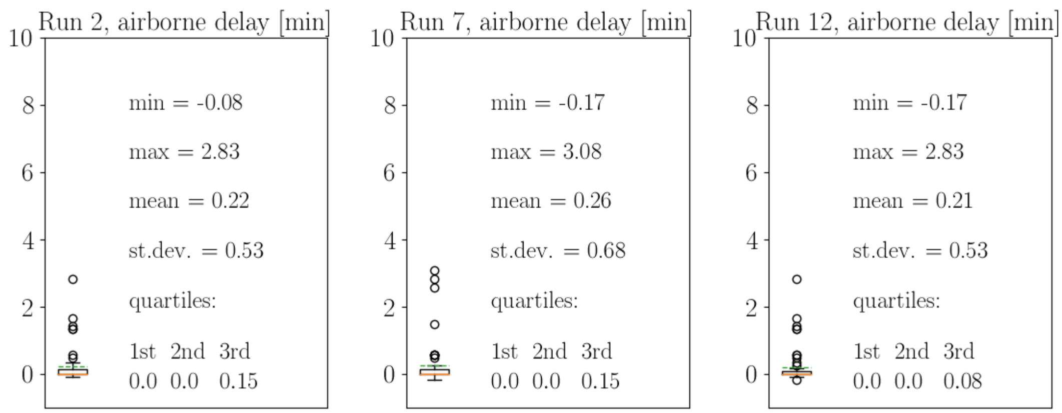

Fig. 11 Airborne delay statistics: runs 2, 7, 12.

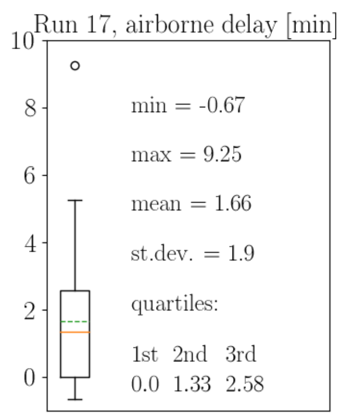

Fig. 12 Airborne delay descriptive statistics: run 17.

Table 3 The qualitative behavior of the means and standard deviations of departure delay.

	Across runs 1-16	For each of runs 1-16	Run 17
FPSS 1	Similar means and standard deviations of departure delay	Means and standard deviations of departure delay higher for FPSS 1 (the larger fleet) than for FPSS 2	Means and standard deviations of departure delay noticeably lower than those for either FPSS in runs 1-16
FPSS 2	Similar means and standard deviations of departure delay		

Table 4 A qualitative characterization of the correlation between departure delay and priority rank.

	Runs 1-6 (arbitration by flight route length)	**Runs 7-11 (arbitration by earliest scheduled departure time)**	**Runs 12-16 (arbitration by earliest estimated landing time)**	**Run 17 (arbitration by earliest estimated landing time)**
FPSS 1	Strong	Strong	Weak	Strong
FPSS 2	Strong	Strong	Weak	

5. *Under a centralized system (run 17), the departure times exhibit a period of congestion (question Q1) corresponding to a surge in demand*

This is seen in the clear unimodal behavior in Fig. 6: the flights scheduled between abscissa markers 340 and 370 experience much higher delay than the rest. This corresponds to the demand adjustments made as part of the scenario settings; see Fig. 1: CSMART introduced the departure delays to keep the demand on the landing runway within capacity. The data for runs 12 through 16 (shown for run 12 in Fig. 5) show a similar period of congestion between abscissa markers 340 and 370.

6. *Under a centralized system, departure delays become more globally optimal, and airborne delays less globally optimal (questions Q3, Q4)*

This is seen by comparing the means and standard deviations shown for run 17 in the bottom portion of Table 2 with the corresponding statistics shown in the same Table for runs 1 through 16.

In the box-and-whisker plots and statistics for departure delay in runs 1, 2, 7, 12 (Figs. 7, 8), the maximal delays are slightly under 62 minutes, the average delays slightly under 10 minutes, and the standard deviation slightly above 15 minutes. In run 17 (Fig. 9), one has a maximum of approximately 10 minutes, an average of approximately 5 minutes, and standard deviation of approximately 6 minutes.

In the box-and-whisker plots and statistics for airborne delay in runs 1, 2, 7, 12 (Figs. 10, 11), all but two maxima are under 3 minutes. The highest occurring maximum for runs 1 through 16 is 6 minutes (some of the data not shown). For run 17 (Fig. 12), the maximum airborne delay is almost 10 minutes (some of the data not shown), and the standard deviation is approximately twice or more than what it is for runs 1 through 16 (see Table 2).

This ability of the centralized system to minimize departure delay is expected: such a system has all the information needed for the minimization.

VII. Summary Conclusions

The results presented herein indicate, in general, without preferential treatment, the FPSS with the larger fleets will incur more departure delay, as expected (see Sec. II.D and Table 3). The priority rankings internal to each FPSS under bin-and-sort policies appear weak when the priority criterion is the flight's earliest estimated landing time and appear strong when the priority criterion is flight route length or earliest scheduled departure time (see Table 4).

The above results indicate that the use of a distributed system to schedule flights has merits and costs. The merits include:

- giving airlines more of a say (than they currently do) in the scheduling of their flights,
- allowing airlines to coordinate their schedules without direct disclosure of proprietary information, and

- having each airline carry its own expenses (for both hardware and software) of scheduling its flights, instead of having all the airlines' flights scheduled by a central entity relying on a single cloud server.

One of the costs, on the other hand, is that a distributed system has greater average departure delays than does a centralized system, as none of the participants in the distributed system has as much information as would a centralized system (see Table 3; compare the rightmost column, corresponding to a centralized system, to the others).

VIII. Future work

The first of the insights listed in Sec. VI implies that the more flights an FPSS owns compared to the other participating FPSS, the more departure delay will accrue to its flights. This may lead to a perception of unfairness. On the other hand, attempts to mitigate this inequity by giving higher preference to the FPSS with more flights may stimulate their growth–in terms of flight volume–even further, increasing their advantage and resulting in monopolies. We intend to analyze the tradeoff between the policies of various schemes of preference.

References

[1] Aho, A.V., Hopcroft, J.E., and Ullman, J.D., *Data Structures and Algorithms*, Addison-Wesley series in computer science and information processing, Addison-Wesley,1983.

[2] Rios, J. L., Smith, I. S., Venkatesan, P., Smith, D. R., Baskaran, V., Jurcak, S. M., Iyer, S. K. and Verma, P., *UTM UAS Service Supplier Development: Sprint 2 Toward Technical Capability Level 4*, NASA Technical Memorandum, 2018.

[3] Lauderdale, T., Bosson, C., Chu, Y.C. and Erzberger, H. *Autonomous Coordinated Airspace Services for Terminal and Enroute Operations with Wind Errors*, AIAA Aviation Forum, Atlanta, GA, 2018.

[4] Meyn, L., *A Closed-Form Solution to Multi-Point Scheduling Problems*, American Institute of Aeronautics and Astronautics (AIAA) Guidance, Navigation, and Control (GNC) Conference and Modeling and Simulation Technologies (MST) Conference, Toronto, Canada, August 2-5, 2010.

[5] Rodionova, O., Arneson, H., Sridhar, B. and Evans, A. *Efficient trajectory options allocation for the collaborative trajectory options program*, 2017 IEEE/AIAA 36th Digital Avionics Systems Conference (DASC), pages 1–10, 2017.

[6] Kulkarni, D. *Impact of Slot Substitution on CTOP Performance*, National Aeronautics and Space Administration, Ames Research Center, 2019.

[7] Sadovsky, A., and Windhorst, R., *A Scheduling Algorithm Compatible with a Distributed Management of Arrivals in the National Airspace System*, 38th Digital Avionics Systems Conference (DASC), San Diego, California, September 8-12, 2019.

[8] Phillips, J., and Sadovsky, A., *Progress in Scheduling Algorithms for a Collaborative Distributed System for Flight Planning*, NASA Technical Memorandum No. 20210000561.