

A Method for Rapid Prototyping and Benchmarking of Arrival Sequencing and Scheduling Algorithms

Jane Thippavong*, Todd Farley†, and Cedric Walker*
NASA Ames Research Center, Moffett Field, CA 94035

Leonard Bagasol§
Spectrum Software Technology, Berlin, NJ 08009

A method is introduced for rapid prototyping and benchmarking of arrival sequencing and scheduling algorithms for air traffic management. The method has three main components: 1) algorithm models 2) recorded air traffic simulation data, and 3) a free software library that interfaces the algorithm models with the simulation output database. By accessing simulation data via a relational database, the method makes it possible to quickly implement prototype algorithms and then to evaluate them without a direct interface to the airspace simulator code. To illustrate this method, a basic first-come, first-served arrival sequencing and scheduling algorithm was studied. Results are offered as benchmarks for comparison with future arrival sequencing and scheduling algorithms.

I. Introduction

IMPROVED resolution of trajectories will spawn a new generation of automation tools and operational concepts that will enable air traffic managers to make more efficient use of the airspace system. Already, new algorithms for arrival sequencing and scheduling are being proposed to support the Next Generation Air Transportation System¹ (NextGen) vision and the increased levels of traffic demand it is expected to serve. To support initial research of such algorithms, a way to rapidly prototype and assess candidate sequencing and scheduling algorithms is needed. High-fidelity simulations generally tend to require many precise inputs and longer lead times for development, which are not conducive to early prototyping and evaluation of potential algorithms. Fast-time simulation tools such as the Airspace Concept Evaluation System^{2,3} (ACES) and the Future ATM Concepts Evaluation Tool⁴ (FACET) were developed to expedite the process of prototyping and testing various ATM concepts. Both tools are stand-alone systems that can simulate aircraft flights across the National Airspace System (NAS). FACET and ACES include functionality that organizes various flight-related output data in a MySQL relational database. These airspace simulators have the necessary inputs stored in their databases for a candidate algorithm; however, incorporating the algorithm into the software requires more than basic programming skills. To address the objective of rapidly testing an algorithm in FACET, Optimal Synthesis Inc. developed a prototyping tool called the Configurable Airspace Research and Analysis Tool-Scriptable⁵ (CARAT#), which provides users direct access to modify FACET functionalities via MATLAB, a popular computing environment used among engineers.

In this paper, a more generic methodology is introduced which leverages the airspace-simulation capabilities contained in ACES and FACET without requiring a direct interface which is expected to reduce the effort of prototyping candidate algorithms. This method is illustrated using FACET with the objective of establishing a baseline against which to compare future sequencing and scheduling algorithms. A study was done to evaluate the behavior of time-based metering (TBM) at a Terminal Radar Approach Control (TRACON) airspace. A simple first-come, first-served TBM algorithm with miles-in-trail (MIT) restrictions was implemented in MATLAB. Then, three scenarios were examined: 1) the conservative use of MIT restrictions over the meter fixes, 2) minimum MIT restrictions over the meter fixes, and 3) the conservative use of MIT restrictions at waypoints upstream of the meter fixes. For each of these scenarios, the aircraft delays at each metering point and arrival throughput were compared. The metrics were also computed with increased demand.

* Research Engineer, Aviation Systems Division, Mail Stop 210-6, AIAA Member.

† Research Engineer, Aviation Systems Division, Mail Stop 210-6, AIAA Associate Fellow.

§ Software Engineer.

The next section details the specific components of the simulation method. A description of the experimental setup used to illustrate the method and to create some initial benchmarks for a common scheduling algorithm follows next. The paper ends with a discussion on further expansions and applications of the method.

II. Rapid Prototyping Method

In the present work, a method for rapid prototyping of air traffic management concepts for simulation—in particular, arrival scheduling and sequencing algorithms—has been developed. This method provides a way for researchers to quickly implement an algorithm, analyze it, and easily tune the algorithm as needed. The architecture contains three main components as depicted in Figure 1: 1) An airspace simulator such as FACET, which stores its output data in a MySQL database, 2) MATLAB, where the algorithm is implemented, and 3) mYm, a program that facilitates data exchange between the simulation output database and MATLAB. Each component is discussed below.

The airspace simulation is the source of key information about the sample of flights used to test the scheduling algorithm. FACET is capable of modeling airspace operations over the United States with accurate representation of Center/sector boundaries, airways, navigation aids /fixes and airports. It is also able to use real-world data such as Aircraft Situation Display to Industry (ASDI) as input. A core capability of FACET is the ability to rapidly model and simulate a variety of aircraft trajectories (i.e. filed flight plan routes, direct or wind-optimal routes) that are also tailored to the specific aircraft type. Results from the simulation are then recorded in a MySQL database. This includes all aircraft trajectory-related information, aircraft type, and anticipated origin and destination airports.

MATLAB provides the environment in which the algorithm is implemented. MATLAB is a widely used tool used among researchers and engineers and requires only a modest amount of programming experience. Also, its built-in graphing and data analysis tools are useful for evaluating and presenting results. These functions can be automated in MATLAB scripts, thereby further reducing the workload and turnaround time incurred when analyzing a battery of test cases.

A software library called mYm⁶ facilitates the interface between MATLAB and the airspace simulation output database. mYm is a free software package that is designed for users to access and manipulate multiple MySQL databases within MATLAB.

This setup allows the algorithms to interface with a standard relational database that is populated by the airspace simulation during execution, rather than to directly interface with the airspace simulation. The algorithms under test are implemented in software separate from the airspace simulation, and so the airspace simulation does not need to be executed with each algorithm under test. As a result, output data generated from a single execution of the airspace simulation may be reused after execution repeatedly and iteratively to test any number of algorithm improvements or permutations. These qualities enable researchers to prototype and assess candidate algorithms with less development time and simulation executions. They also enable researchers to readily repeat experiments with differing airspace simulation tools capable of writing its output to the same standard database format.

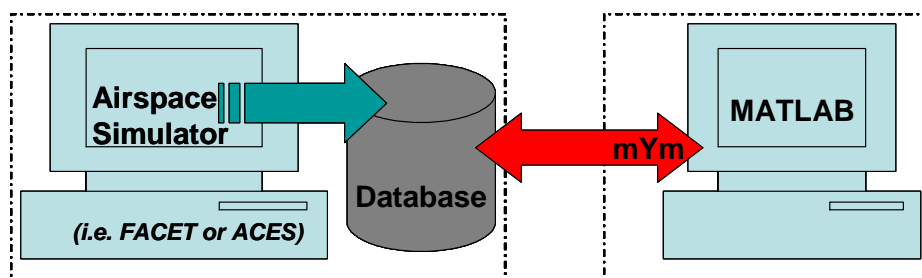


Figure 1. Components of the rapid prototyping method.

III. Experiment Design

An experiment was designed to illustrate the rapid prototyping method. A candidate sequencing and scheduling algorithm was analyzed to provide a set of baseline performance metrics against which new algorithms can be compared using the same rapid prototyping method. This section describes the details of the experiment, including the algorithm implemented in MATLAB, the inputs provided by FACET, and the test scenarios.

A. Sequencing and Scheduling Algorithm

A simple TBM sequencing and scheduling algorithm was scripted in MATLAB. TBM has been shown to improve the predictability of aircraft trajectories which could improve the throughput of airports.^{7,8} Three types of scheduling reference points were included: runways (RWY), meter fixes (MF), and outer points (OP), as illustrated in Figure 2. Each outer point fed a single meter fix; each meter fix served all runways in the simulated terminal area. For each aircraft, arrival times were estimated for the relevant OP-MF-runway reference points. These times are referred to as the estimated times of arrival, or ETAs. Using these ETAs, aircraft were then sequenced on a first-come, first-served basis, and they were spaced to conform to the applicable miles-in-trail (MIT) constraint(s). The OP, MF and runway crossing times computed after sequencing and spacing are referred to as the scheduled times of arrival, or STAs.

The inputs to the scheduling algorithm were obtained from the simulation output database, which was populated during execution of the airspace simulation (i.e., FACET). Using FACET's flight simulation capabilities, the aircraft departed at its proposed departure time and flew along its given flight plan. Based on FACET's flight simulation, aircraft groundspeeds, ETAs to the OPs, and times of flight between two points (the ETEs) were obtained. Details of the algorithm are as follows.

The ETAs to the outer points, ETA_{OP} , were first sorted sequentially. For the earliest aircraft in the queue, its STA_{OP} was set to its ETA_{OP} . Continuing with the aircraft with the next-earliest ETA to the OP, the STA at the meter point, STA_{MF} , was computed. Two constraints were taken into consideration when calculating an aircraft's STA_{OP} . First, the aircraft was scheduled no earlier than its ETA_{OP} . Second, the STA_{OP} took into account any MIT restriction that was imposed at the outer point by scheduling the aircraft no earlier than the STA_{OP} of the leading aircraft plus the necessary time constraint needed to adhere to the in-trail restriction, MIT_{OP} . Between these two constraints, the most restrictive time was retained. Thus,

$$STA_{OP} = \text{Latest}(ETA_{OP}, STA_{OP, \text{leading ac}} + MIT_{OP} / GS_{OP})$$

where GS_{OP} is the groundspeed (assumed to be constant) of the aircraft at the outer point.

After computing the STA to the outer point, this time was used to estimate the ETA at the meter fix, ETA_{MF} , by adding the travel time from the outer point to the meter fix, $ETE_{OP \text{ to MF}}$:

$$ETA_{MF} = STA_{OP} + ETE_{OP \text{ to MF}}$$

Then, the ETAs to the meter fixes were sorted by time, and the STA at the meter fix, STA_{MF} , was calculated likewise to the outer point, where the first aircraft's STA was set to its ETA, and the next-earliest aircraft's STA was computed as follows:

$$STA_{MF} = \text{Latest}(ETA_{MF}, STA_{MF, \text{leading ac}} + MIT_{MF} / GS_{MF})$$

Next, the ETAs at the runway, ETA_{RWY} , were calculated in a similar fashion, by adding the travel time from the meter fix to the runway, $ETE_{MF \text{ to RWY}}$, to the STA at the meter fix, and then sorted by time:

$$ETA_{RWY} = STA_{MF} + ETE_{MF \text{ to RWY}}$$

Finally, the STA of the first aircraft in line at the runway was set to its ETA, and the rest of the STAs were calculated taking into account the minimum separation requirements at the runway:

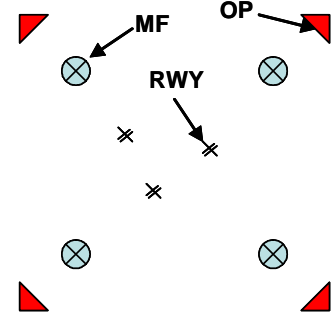


Figure 2. Scheduling reference points.

$$STA_{RWY} = \text{Latest}(ETA_{RWY}, STA_{RWY, \text{leading aircraft}} + MIT_{RWY} / GS_{RWY})$$

B. Metrics

To characterize the performance of the algorithm, two metrics were computed for each metering reference point: 1) aircraft total delay and 2) arrival rate. The arrival rate for a particular metering point was computed based on aircraft STAs. The aircraft delay at a point was defined to be the difference between an aircraft's ETA and STA. Thus, an aircraft's total delay was calculated by summing the individual delays:

$$\text{Total Delay}_{ac} = \text{Delay}_{OP} + \text{Delay}_{MF} + \text{Delay}_{RWY}$$

where

$$\begin{aligned} \text{Delay}_{OP} &= STA_{OP} - ETA_{OP} \\ \text{Delay}_{MF} &= STA_{MF} - ETA_{MF} \\ \text{Delay}_{RWY} &= STA_{RWY} - ETA_{RWY} \end{aligned}$$

C. Test Airspace Layout

The Philadelphia TRACON airspace was chosen for the study. The airspace was modeled with one runway at each of four airports within the Philadelphia TRACON airspace: Philadelphia International Airport (PHL), New Castle Airport (Wilmington, DE) (ILG), Trenton Airport (TTN), and Philadelphia Northeast Airport (PNE). All four arrival meter fixes were modeled: BUNTS, SPUDS, TERRI, and VCN (see Figure 3). Flights from the Midwest and West typically arrive via BUNTS. Traffic from the South and East typically arrives over TERRI and Cedar Lake (VCN), respectively. Traffic from the North typically arrives via SPUDS. Outer points were modeled in the simulation at DUNEE, PSB (Phillipsburg), and COFAX. DUNEE typically feeds westbound traffic into VCN, while COFAX and PSB typically funnel traffic into BUNTS. No outer points were modeled upstream of SPUDS or TERRI.

D. Test Scenarios

Three scenarios were designed to probe performance differences associated with various metering configurations. Scenario 1 modeled present-day traffic management techniques, applying constraints at the TRACON boundary by imposing typical MIT restrictions over the meter fixes. Scenario 2 modeled an idealized time-based metering operation in which in-trail restrictions were set to minimum values. Scenario 3 applied traffic management constraints further upstream by imposing MIT restrictions at waypoints along en-route airspace boundaries. The MIT restrictions used at the outer points, meter fixes, and runway for each scenario are specified in Tables 1 and 2.

Scenario 1. The first scenario was intended to model present-day operations in Philadelphia TRACON. Typical MIT restrictions were used at the meter fixes, as determined from field observations and interviews with subject-matter experts. Traffic management coordinators (TMCs) subjectively set these MIT restrictions based on their knowledge of inbound streams, intuition, and experience. This practice is used in almost all terminal areas that are not equipped with automated decision support technology.

Scenario 2. The second scenario was configured to model typical terminal-area TBM operations. In-trail restrictions were reduced to the separation standard minimum, and spacing was based on the inter-arrival wake vortex spacing standards for the TBM arrival sequence.

The restrictions manually set by TMCs are sometimes overly conservative and do not efficiently take into account merging streams. TBM, on the other hand, uses predicted 4-D trajectories to tailor the arrival time of each aircraft based on multiple streams into the airport. Thus, setting large miles-in-trail restrictions are unnecessary and instead, minimum aircraft separation requirements were set at the meter fixes for this simulation.

Scenario 3. The third scenario combined terminal-area TBM operations with upstream MIT restrictions in the transition airspace. In some TBM operations, MIT restrictions are placed at upstream waypoints to help reduce the

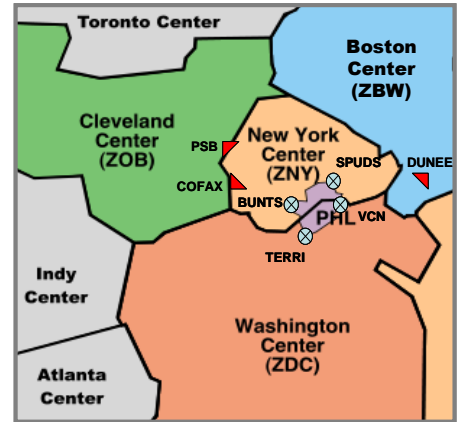


Figure 3. Map of Philadelphia TRACON, its meter fixes, and its outer points of interest.

demand inbound to the meter fixes. To model this, MIT restrictions commonly imposed at facility boundaries upstream of the Philadelphia TRACON were placed at the outer points. Spacing requirements inside the TRACON were modeled as in Scenario 2.

Table 1. MIT restrictions placed at the meter fixes and outer points for each simulation.

METER POINT	Scenario 1	Scenario 2	Scenario 3
BUNTS	10	5	5
TERRI	15	5	5
VCN	15	5	5
SPUDS	20	5	5
PSB	0	0	20
COFAX	0	0	20
DUNEE	0	0	20

Table 2. MIT restriction between two types of aircraft at the runway.

AC TYPE	Trailing					
	HJ	LJ	LT	ST	SP	57
Leading						
HJ	4.0	5.0	5.0	6.0	6.0	5.0
LJ	2.5	2.5	3.0	4.0	4.0	2.5
LT	2.5	2.5	2.5	4.0	4.0	2.5
ST	2.5	2.5	2.5	2.5	3.0	2.5
SP	2.5	2.5	2.5	2.5	2.5	2.5
57	4.0	4.0	4.0	5.0	5.0	4.0

where

- HJ = heavy jet
- LJ = large jet
- LT = large turbo
- ST = small turbo
- SP = small prop
- 57 = B757

These values based upon FAA regulations for minimum wake-vortex separation requirements at the runway.⁹

IV. Results

The three scenarios outlined above were executed for each of three demand sets: a present-day traffic load using ASDI traffic data recorded on March 23, 2006 and future-day traffic loads of approximately double and triple that of the nominal demand set. The results consist of performance metrics for average delay per aircraft at each metering reference point (OP, MF and runway) and arrival rate at each reference point. The results for the present-day and future-day demand sets are presented in the subsections that follow.

A. Present-Day Traffic Demand

The number of aircraft that passed through a particular metering point for each airport in Philadelphia TRACON is summarized in Table 3. Most of the arrivals were to PHL, whereas arrivals bound for smaller airports in the TRACON (i.e., ILG, TTN, and PNE) accounted for just about 10% of the total demand. About 40% passed through the outer points before heading to the meter fixes. More traffic arrived via the western and southern fixes than through the northern fix, as is consistent with real-world operations. The distributions of the demand (based on aircraft ETAs) for each metering point are displayed in Figure 4.

Table 3. Number of aircraft passing through a particular metering point.

AIRPORT	METER FIX						OUTER POINT			
	BUNTS	TERRI	VCN	SPUDS	OTHER	Total	PSB	COFAX	DUNEE	Total
PHL	255	179	171	58	56	719	28	202	64	294
ILG	6	15	5	0	3	29	1	4	1	6
TTN	4	10	2	0	21	37	0	4	8	12
PNE	7	2	2	1	20	32	1	3	2	6
Total	272	206	180	59	100	817	30	213	75	318

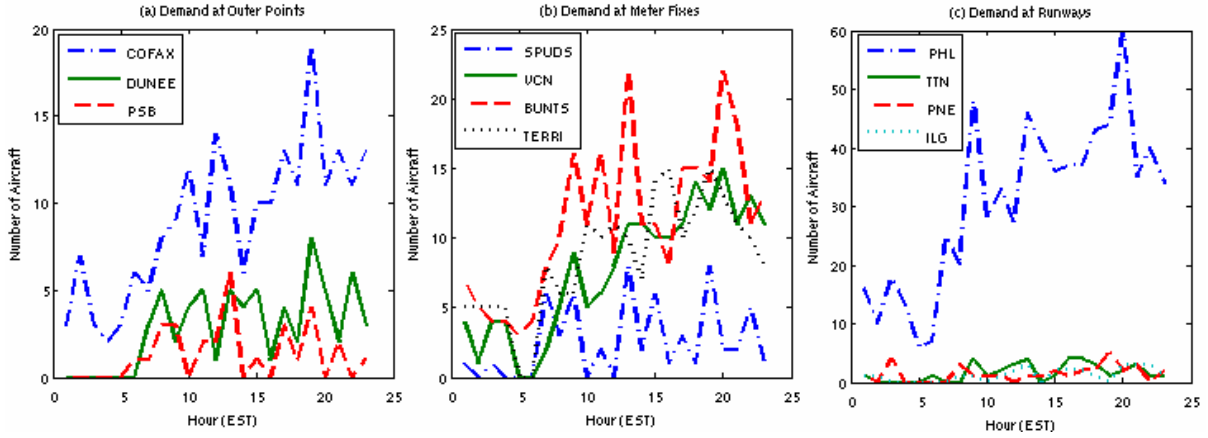


Figure 4. Distribution of the demand at each metering point.

The average aircraft delay on an hourly basis for all three scenarios is shown in Figures 5-7. Figure 8 shows the corresponding histogram of the aircraft total delay at PHL and TTN airports. These plots exhibit the difference in the way delays are distributed among the metering points.

Scenario 1. In Scenario 1, capacity constraints were applied at the meter fixes by imposing MIT restrictions typical of present-day operations at Philadelphia TRACON. Figure 5 contrasts the accumulation of delay at the meter fixes versus the runways. Figure 5a depicts high delays at the meter fixes, while Figure 5b shows only minor delays at the runways. The disproportion of delays at the meter fixes suggests that capacity was over-constrained at the meter fixes and under-constrained at the outer points and runways.

Figure 5a shows how delays were distributed across the meter fixes. In particular, the delays at VCN and TERRI increased rapidly in the second half of the day. This escalation of delay is characteristic of a first-come, first-served-tippy queuing system where demand has outstripped capacity over a sustained period. Note that although BUNTS received somewhat more demand than did VCN and TERRI, the average aircraft delay at BUNTS was low compared to VCN and TERRI. This is because Scenario 1 was configured with more capacity at BUNTS (where a 10-MIT restriction was in place) than at VCN (where a 15-MIT restriction was applied). In other words, the demand-capacity imbalance at VCN and TERRI was more acute than at BUNTS. Minimal restrictions were placed at the airports and outer points, reflected in Figure 5b by the low delays for those metering reference points.

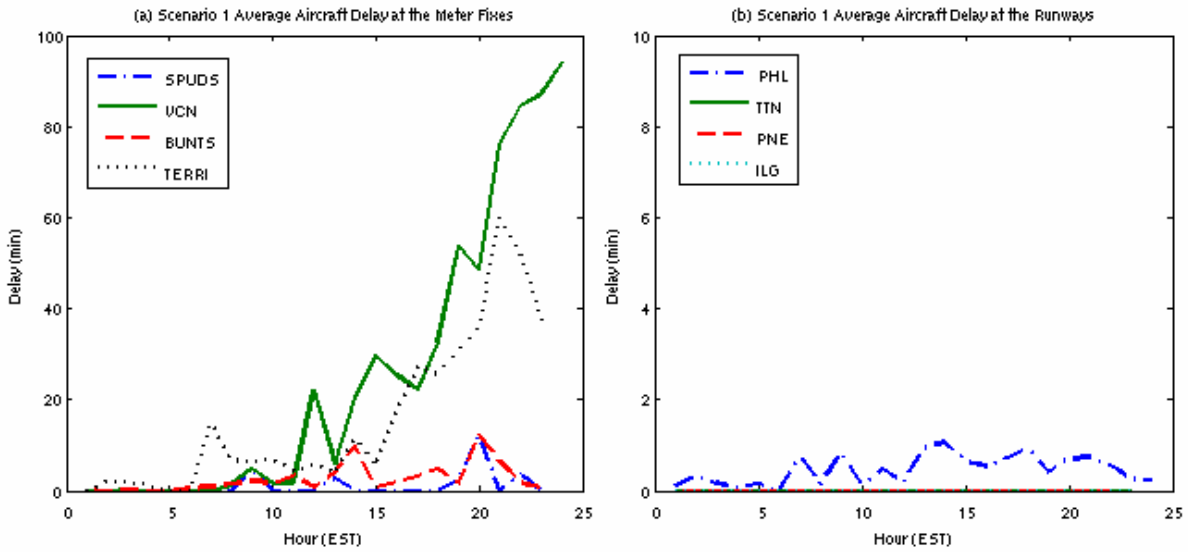


Figure 5. Average aircraft delay at each hour for Scenario 1.

Scenario 2. In the second scenario, capacity constraints at the meter fixes were reduced to the minimum (5 nmi.) and capacity was regulated dynamically based on the inter-arrival spacing of aircraft in sequence to the runways.

The results depicted in Figure 6a show considerably less delay at the meter fixes than Figure 5a from Scenario 1. This is expected, given the reduction in constraints there. However, the lower restrictions at the meter fixes resulted in increased demand at the runways, eroding the demand-capacity balance there and resulting in an increase in average aircraft delays, as seen in Figure 6b.

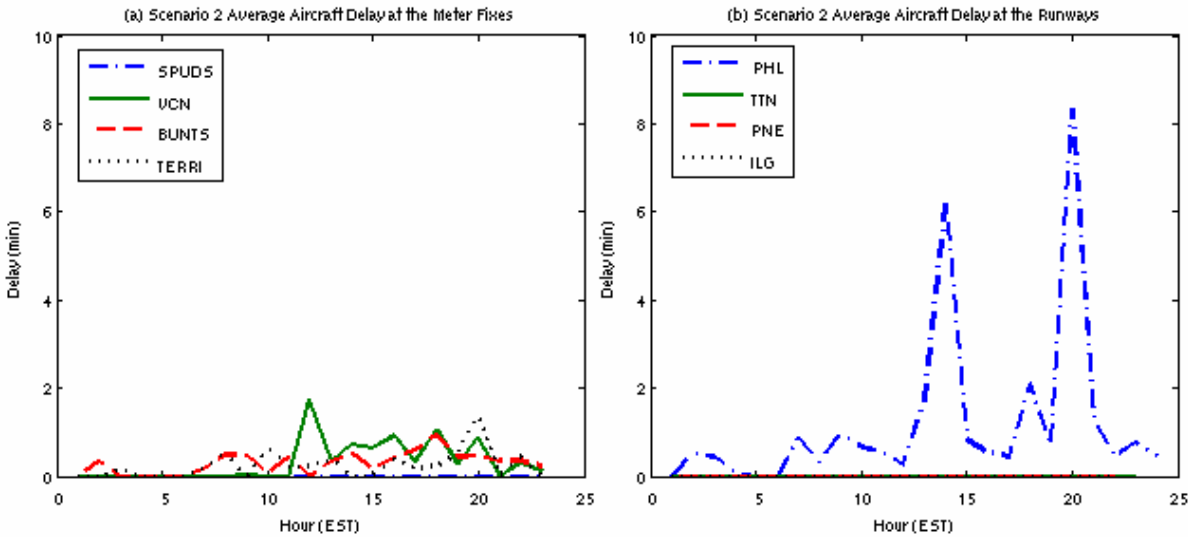


Figure 6. Average aircraft delay at each hour for Scenario 2.

Scenario 3. In Scenario 3, capacity constraints were applied at the outer points by imposing MIT restrictions at the facility boundaries upstream of Philadelphia TRACON. Accordingly, Figure 7a indicates that aircraft passing through COFAX assumed a large amount of delay due to the MIT restriction. Figures 7b and 7c indicate no appreciable reduction in delay at the meter fixes or runways, respectively relative to Scenario 2 results. Therefore, it can be concluded that the restriction at COFAX was not an efficient choice in terms of average per-aircraft delay. Only about a third of the total traffic to Philadelphia TRACON pass through the outer points and are affected by the MIT restriction, so the impact on delays aren't as prominent. With minimum restrictions at the meter fixes and

runway, the delays at these points (as seen in Figure 7b and 7c) are comparable to those seen in Scenario 2 (Figure 6).

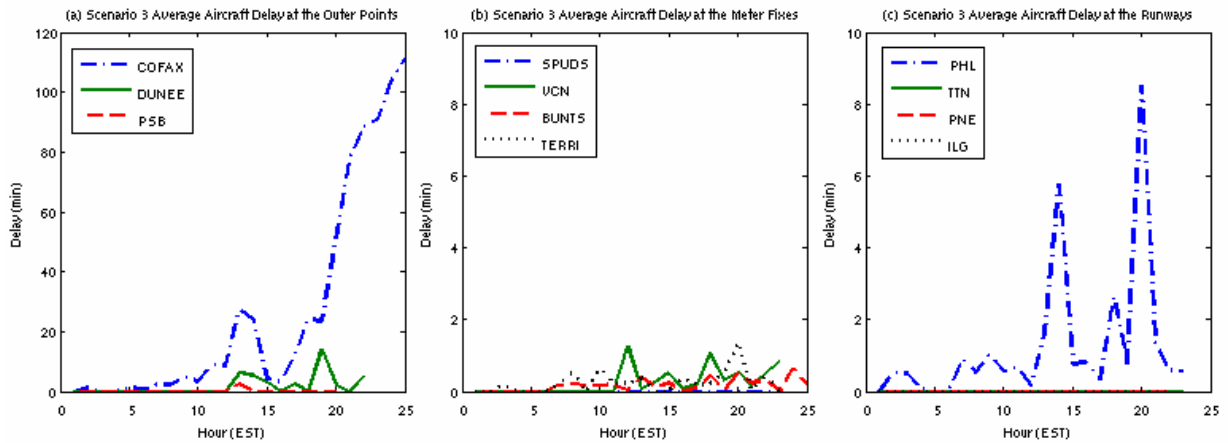


Figure 7. Average aircraft delay at each hour for Scenario 3.

Figure 8 compares the distribution of total arrival delay for Scenarios 1, 2 and 3. The histograms indicate that the metering configuration used in Scenario 2 was the most efficient in terms of arrival delay. Figure 8c shows that aircraft generally experienced less total delay under the metering configuration used in Scenario 3 as compared to that of Scenario 1, but some aircraft were assigned large delays due to the restrictions at the outer points. In all three scenarios, aircraft bound for the secondary airports in Philadelphia TRACON (i.e., ILG, TTN, and PNE) experienced no delays, because the low demand for those airports was considerably less than their capacity.

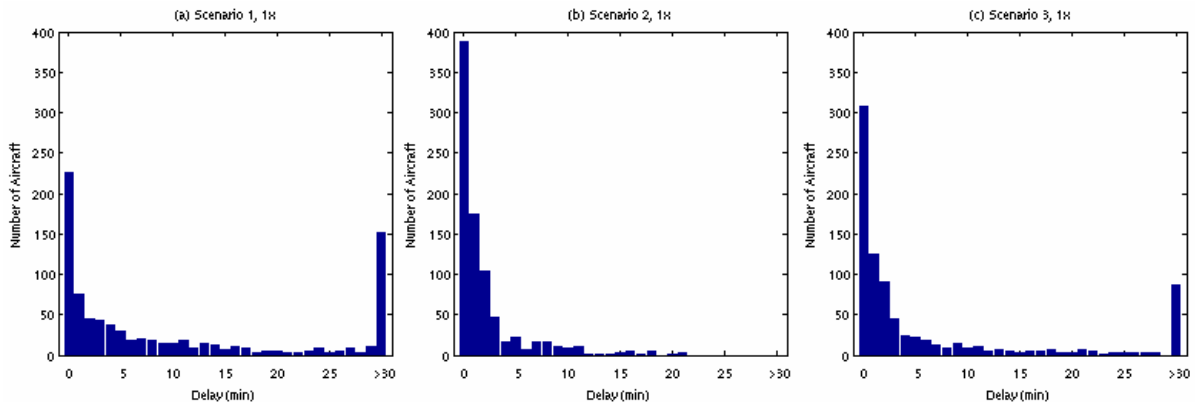


Figure 8. Distribution of aircraft total delay for each scenario.

The arrival rates at various metering points for all scenarios are depicted in Figures 9 and 10. Figure 9 shows the arrival rate per hour at each of the outer points. The plots for Scenarios 1 and 2 coincide with the demand, since no restrictions were placed on the outer points in those scenarios. Figure 9a shows that the MIT restrictions placed on COFAX in Scenario 3 moderated the demand. No such moderation is seen in the Scenario 3 results in Figures 9b and 9c, because there was not enough demand at DUNEE and PSB for the MIT restrictions placed at these points to significantly affect the flow rate.

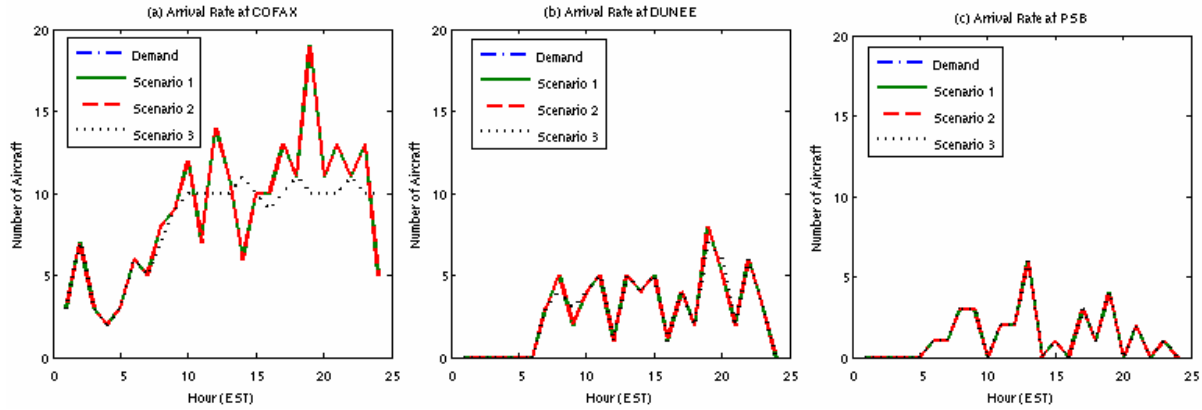


Figure 9. Arrival rate at the outer points for each scenario.

Figure 10 shows the arrival rate per hour at each of the meter fixes. The effects on the arrival rate at the meter fixes are interesting since there may be two MIT restrictions that are affecting the traffic (i.e., an outer-point restriction and a meter-fix restriction). As was the case at DUNEE and PSB, there was insufficient demand at SPUDS for the restriction there to significantly alter the hourly arrival rate, as seen in Figure 10a. On the other hand, the high demand at VCN and TERRI were not buffered by the outer points. Only in Scenario 1 were controls directly applied to these traffic flows. As a result, Figures 10b and 10d show that only the restrictions placed at the

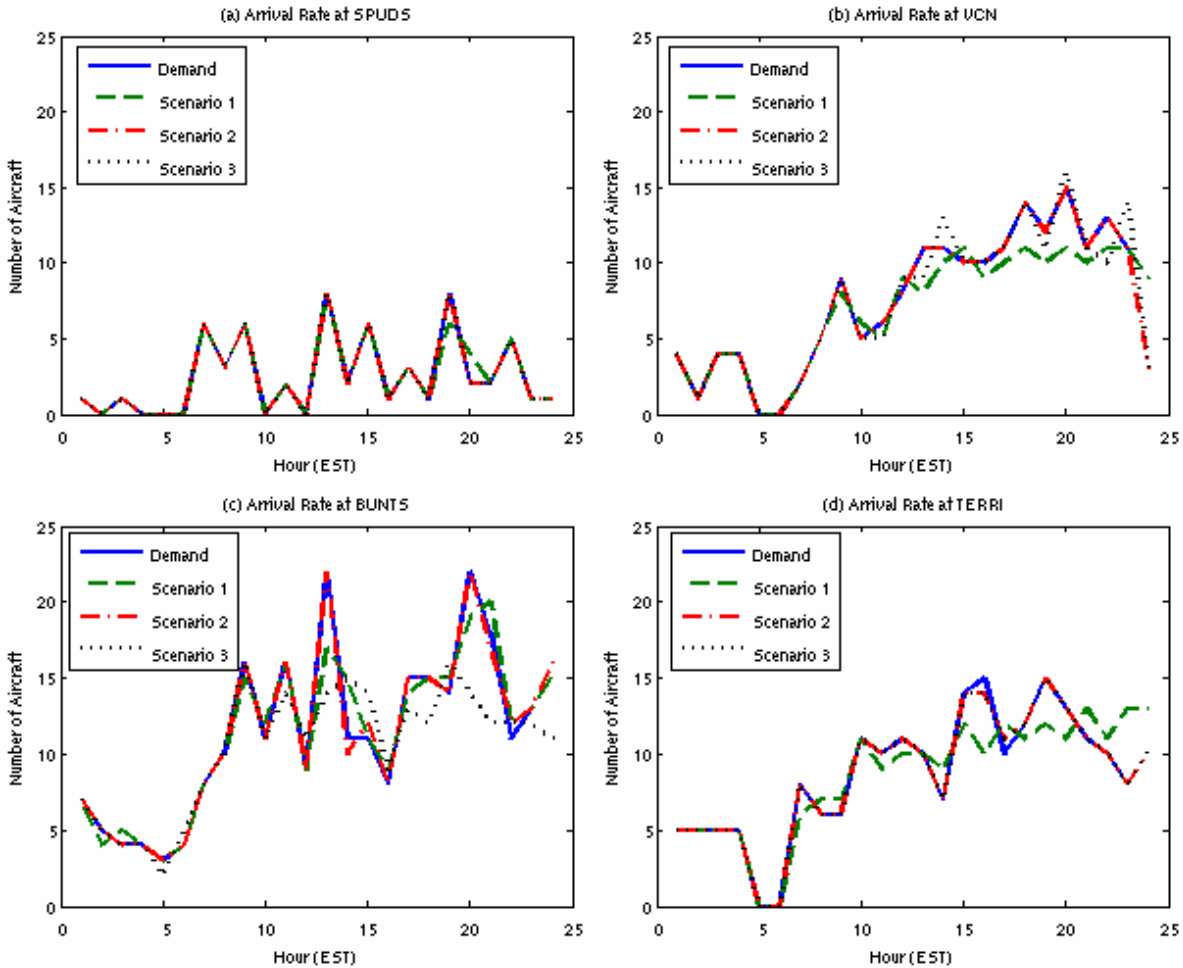


Figure 10. Arrival rate at the meter fixes for each scenario.

meter fixes in Scenario 1 were successful in moderating the demand. Lastly, Figure 10c indicates that the restrictions imposed at or upstream of BUNTS (ref. Scenario 1 and 3, respectively) were effective in reducing the arrival rate across that fix as compared to the minimum spacing that was applied in Scenario 2.

Overall, these results indicate that the rapid prototyping system produces results in terms of delay and arrival rate that are in line with expectations. They suggest that the system produces valid output for nominal air traffic demand profiles and may be considered a valid test bed for initial evaluation of alternative sequencing and scheduling algorithms. In the next subsection, the same performance metrics are examined under an increased level of demand.

B. Future-Day Traffic Demand

Since air traffic demand is expected to increase in line with historical rates,¹⁰ the simulations above were rerun with extrapolated demand sets intended to represent traffic loads twice and triple that of today. The extrapolated data sets were generated by duplicating the original traffic sample and time-shifting each aircraft by 1.5 minutes. In this way, current demand patterns were amplified, as opposed to increasing the steady-state traffic load across the 24-hr sample period.

Figures 11 and 12 display histograms of total aircraft delay for each of the simulations under two and three times the current demand, respectively. (Note that the y-axis scale is logarithmic.) A much larger proportion of aircraft experienced triple-digit delays than for the nominal case shown in Figure 8. This is expected, as the demand at each metering reference point was doubled or tripled, but the capacity was not. Arrival-rate metrics would be expected to show similar degradation under these levels of demand without significant increases in capacity. Table 4 shows a summary of the cumulative aircraft delay for each simulation with different levels of demand. As other studies have shown,¹¹ the delays grow exponentially as demand increases.

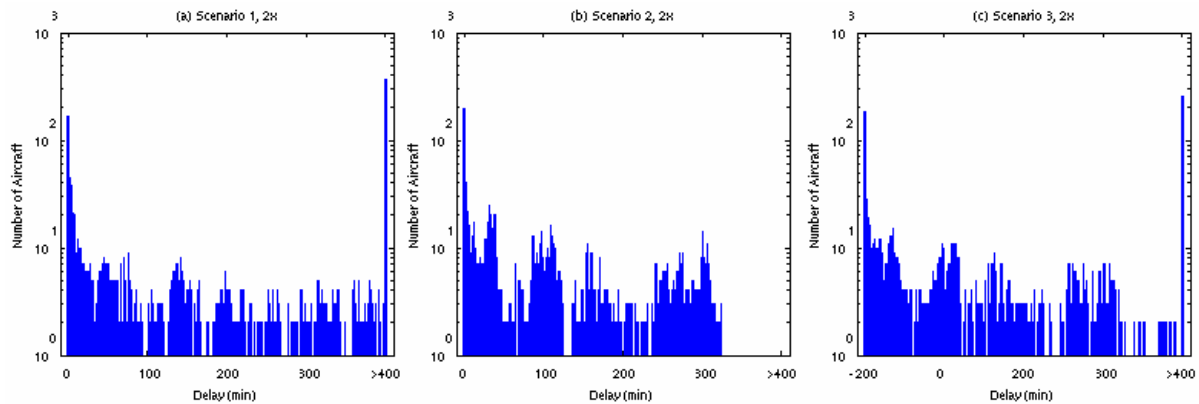


Figure 11. Distribution of aircraft total delay for each scenario under 2x demand.

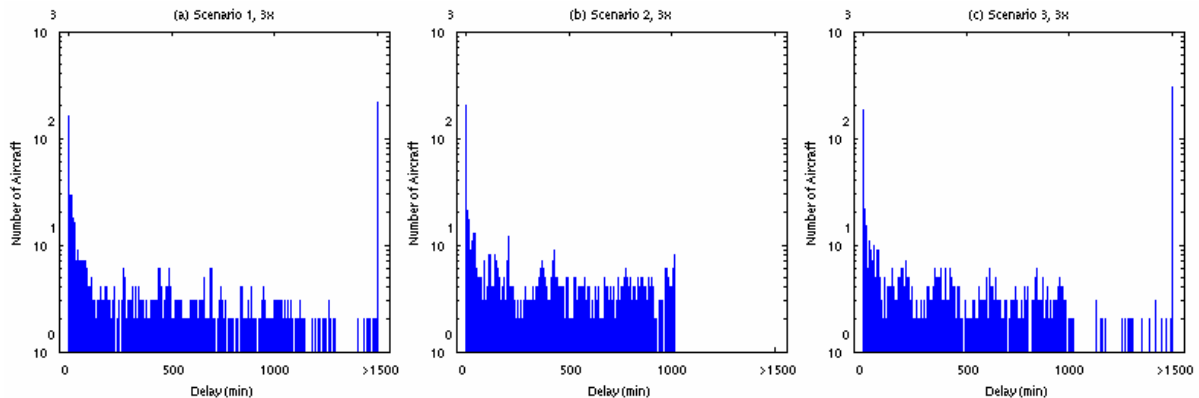


Figure 12. Distribution of aircraft total delay for each scenario under 3x demand.

Table 4. Total aircraft delay in minutes for each scenario with varying levels of demand.

DEMAND LEVEL	Scenario 1	Scenario 2	Scenario 3
1x	17454	1571	9267
2x	439037	178823	383882
3x	1478231	936812	1638767

V. Discussion

The methods and approach described in this study provide a capability for comparing results from other potential sequencing and scheduling algorithms evaluated with the same rapid prototyping and analysis methodology. The data from the study do corroborate with what is expected from empirical evidence, which can now be used as a baseline to quickly measure incremental benefits between alternative algorithms. A limitation is that the results may not be easily comparable to real-world results, since the terminal-area routing and airport details were not fully represented in FACET. Additionally, many of the uncertainties associated with the air traffic system were not incorporated in the simulation. Thus, the metrics obtained are under simplified conditions and are for preliminary analysis only; validation of leading candidate algorithms in a higher-fidelity environment would be advised.

The rapid prototyping method is best suited to foundational research, where a quick method to initially prototype and evaluate potential algorithms can expediently identify leading candidates for more thorough analysis of a higher fidelity system. Parameters used in the simulation can easily be modified for sensitivity analysis. This is particularly useful for conducting trade studies to determine to which input signals the system is most sensitive.

This study was done with minimal software changes to FACET. Some work was done to assure particular information from FACET was made available in the database. Everything else—the algorithm and analysis—was scripted entirely in MATLAB, and mYm provided the interface between these two systems. As mYm can also be used to interface MATLAB with any MySQL database, the method introduced here could be used with any airspace simulator (e.g., FACET, ACES), provided the necessary data were stored in a MySQL database.

Due to its modularity, the architecture of the rapid prototyping system lends itself to easy modification. Development work can be done on a particular system and tested without affecting the other pieces of the simulation. For example, suppose the simulation results were anticipated to be particularly sensitive to the way the airport is modeled in FACET. The changes needed to improve the fidelity of the airport model would be incorporated into FACET and the effects of the modification would be reflected in the database. This would require no changes to the algorithm implemented in MATLAB. Then, the simulations could easily be run again with the new airport modeling and be compared to prior results. In short, the prototyping methodology can be utilized in a variety of applications. The experiment described above shows just one application of the prototyping methodology.

VI. Conclusions

A method has been developed to reduce the effort associated with rapid-prototyping and analysis of new arrival sequencing and scheduling algorithms. The method makes use of three main components: 1) algorithm models scripted in MATLAB, 2) air traffic data stored in a MySQL database, and 3) a software interface between the two, implemented using mYm. The method provides an indirect, off-line interface to advanced airspace simulators well suited to initial, foundational studies of sequencing and scheduling algorithms.

The method was demonstrated by incorporating an arrival sequencing and scheduling algorithm in MATLAB and using FACET flight simulation data as input. Results from this study provide a set of base metrics that can be used to compare other arrival sequencing and scheduling algorithms that are analyzed with the same simulation method. The entire process is modular and can thus be used for analyzing a variety of other applications depending on which components are modified.

Acknowledgments

The authors thank Dr. Shon Grabbe, Dr. Kapil Sheth, Daniel Mulfinger, and Dave Darling for sharing their expertise on the FACET tool. The authors also appreciate the efforts of Calvin Tang and Sarah Lemmer for assisting with preliminary analysis of the algorithm.

References

-
- ¹ Joint Planning and Development Office, "Next Generation Air Transportation System: Integrated Plan," 2004.
 - ² Meyn, L., et. al., "Build 4 of the Airspace Concept Evaluation System," AIAA-2006-6110 AIAA Modeling and Simulation Technologies Conference and Exhibit, Keystone, Colorado, Aug. 21-24, 2006.
 - ³ Windhorst R., et. al., "The Airspace Concept Evaluation Software Architecture," AIAA-2006-6231 AIAA Guidance, Navigation, and Control Conference and Exhibit, Keystone, Colorado, Aug. 21-24, 2006.
 - ⁴ Bilimoria, K., et. al., "FACET: Future ATM Concepts Evaluation Tool," *Air Traffic Control Quarterly*, Vol. 9, No. 1, 2001, pp. 1-20.
 - ⁵ Menon, P. K., et. al., "A Rapid-Prototyping Environment for En Route Air Traffic Management Research," AIAA 2005-6046 AIAA Guidance, Navigation, and Control Conference and Exhibit, San Francisco, California, Aug. 15-18, 2005.
 - ⁶ Maret, Y., "MySQL Wrapper for MATLAB," Software, 2007. <http://sourceforge.net/projects/mym>
 - ⁷ Swenson, H. N., et. al., "Design and Operational Evaluation of the Traffic Management Advisor at the Fort Worth Air Route Traffic Control Center," 1st USA/Europe Air Traffic Management R&D Seminar, Saclay, France, June, 1997.
 - ⁸ Farley, T. C., et. al., "Multi-Center Traffic Management Advisor: Operational Test Results," AIAA 2005-7300, Proceedings of the 5th AIAA Aviation Technology, Integration, and Operations Conference, Arlington, Virginia, September 26-28, 2005.
 - ⁹ FAA Order 7110.65, "Air Traffic Control," U.S. Dept. of Transportation, Washington, DC.
http://ctas.arc.nasa.gov/project_description/index.html
 - ¹⁰ U.S. Department of Transportation, Federal Aviation Administration, "Terminal Area Forecast Summary," FAA-APO-FAA-07-1, March 2007.
 - ¹¹ National Civil Aviation Review Commission, "Avoiding Aviation Gridlock and Reducing the Accident Rate," Washington, DC, December 1997.