

NASA/CR–2012-217344



# Investigation, Modeling, and Analysis of Integrated Metroplex Arrival and Departure Coordination Concepts

*John-Paul B. Clarke, James Brooks, Evan McClain, Anwasha Roy Paladhi, and Leihong Li  
Georgia Institute of Technology, Atlanta, Georgia*

*David Schleicher, Aditya Saraf, and Sebastian Timar  
Sensis Seagull Technology Center, Campbell, California*

*Don Crisp, Jason Bertino, and Ryan Laroza  
ATAC Corporation, Sunnyvale, California*

*Carolyn Cross  
Metron Aviation, Dulles, Virginia*

## NASA STI Program . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA scientific and technical information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NASA Aeronautics and Space Database and its public interface, the NASA Technical Report Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- **TECHNICAL PUBLICATION.** Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers, but having less stringent limitations on manuscript length and extent of graphic presentations.
- **TECHNICAL MEMORANDUM.** Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.
- **CONTRACTOR REPORT.** Scientific and technical findings by NASA-sponsored contractors and grantees.
- **CONFERENCE PUBLICATION.** Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.
- **SPECIAL PUBLICATION.** Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.
- **TECHNICAL TRANSLATION.** English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include creating custom thesauri, building customized databases, and organizing and publishing research results.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at <http://www.sti.nasa.gov>
- E-mail your question via the Internet to [help@sti.nasa.gov](mailto:help@sti.nasa.gov)
- Fax your question to the NASA STI Help Desk at 443-757-5803
- Phone the NASA STI Help Desk at 443-757-5802
- Write to:  
NASA STI Help Desk  
NASA Center for AeroSpace Information  
7115 Standard Drive  
Hanover, MD 21076-1320

NASA/CR–2012-217344



# Investigation, Modeling, and Analysis of Integrated Metroplex Arrival and Departure Coordination Concepts

*John-Paul B. Clarke, James Brooks, Evan McClain, Anwesha Roy Paladhi, and Leihong Li  
Georgia Institute of Technology, Atlanta, Georgia*

*David Schleicher, Aditya Saraf, and Sebastian Timar  
Sensis Seagull Technology Center, Campbell, California*

*Don Crisp, Jason Bertino, and Ryan Laroza  
ATAC Corporation, Sunnyvale, California*

*Carolyn Cross  
Metron Aviation, Dulles, Virginia*

National Aeronautics and  
Space Administration

Langley Research Center  
Hampton, Virginia 23681-2199

Prepared for Langley Research Center  
under Contract NNL10AA14C

February 2012

## **Acknowledgments**

We would like to acknowledge Michael Sorokach, Technical Monitor, and Rosa Oseguera-Lohr, Metroplex Systems Analysis Lead, of NASA for their assistance and guidance in this project.

The use of trademarks or names of manufacturers in this report is for accurate reporting and does not constitute an official endorsement, either expressed or implied, of such products or manufacturers by the National Aeronautics and Space Administration.

Available from:

NASA Center for AeroSpace Information  
7115 Standard Drive  
Hanover, MD 21076-1320  
443-757-5802

## Contents

1	Introduction.....	11
1.1	Metroplex Definition .....	11
1.2	Metroplex Interdependencies.....	11
1.3	Research Focus .....	12
1.4	Structure of Report.....	13
2	Abbreviations/Acronyms .....	15
3	Multiplexer Algorithm .....	16
3.1	Objective.....	16
3.2	Algorithm 1: Staged Optimization for Arrival Fix and Runway Operations.....	17
3.3	Algorithm 2: Joint Optimization for Airspace Fixes and Runway Operations.....	17
3.3.1	Constraints Used to Model Metroplex Airspace.....	18
3.3.2	Airspace Fix Constraints.....	18
3.3.3	Runway Constraints .....	18
3.3.4	Solution Methodology .....	19
3.3.5	Solution Algorithm .....	19
4	TMA Algorithm .....	20
4.1	Background.....	20
4.2	Technical Approach and Algorithm Description.....	20
4.3	TMA Scheduling Capability (Dynamic Planner) Description.....	21
4.4	Stream Class Concept and Order of Consideration Algorithm.....	22
4.5	TMA Scheduling Capability Emulation Description.....	23
4.6	Validation Approach.....	24
4.7	Validation Results.....	24
5	Metroplex Geometries .....	27
5.1	Generic Geometries .....	27
5.2	New York Geometries .....	35
6	Analysis Framework and Methodology.....	37
6.1	Analysis Architecture.....	37
6.2	Assumptions.....	40
6.2.1	Required Minimum Separation Criteria.....	40
6.2.2	TRACON Traversal Time Assumptions.....	42
6.3	Metrics .....	43
7	Experimental Testbed .....	43
7.1	Simulation Platforms .....	43
7.1.1	Simple Queuing Model.....	43
7.1.2	SIMMOD and SIMMOD PRO!.....	43
7.2	Demand Scenarios .....	44
7.2.1	Generic Metroplex .....	44
7.2.2	Demand Levels .....	45
7.2.3	Airspace Adaptation.....	46
7.2.4	Traffic Demand File Output.....	47
7.2.5	Generated Traffic Demand Sets.....	48

7.2.6	Additional Demand Variables.....	49
7.3	New York Metroplex .....	50
7.3.1	Demand Levels .....	51
7.3.2	Airspace Adaptation.....	52
8	Results.....	52
8.1	Generic Airspace Results.....	53
8.2	New York Airspace Results.....	59
9	Conclusions.....	61
10	Implementation Issues and Concept of Operations.....	62
10.1	Today’s Metroplex Inefficiencies .....	62
10.2	A Potential Metroplex Solution: the “Multiplexer”.....	62
10.3	Expected Benefits .....	64
10.4	Functional Overview.....	65
10.4.1	Roles .....	65
10.4.2	Responsibilities.....	66
10.4.3	System Function.....	66
10.4.4	Operational Scenarios .....	67
10.5	Flight Applicability.....	68
10.5.1	User Classes .....	68
10.5.2	Flight Rules.....	68
10.5.3	NAS Domains .....	68
10.6	New System Architecture Requirements .....	69
10.6.1	Aircraft Equipage.....	69
10.6.2	Communication, Navigation, and Surveillance (CNS).....	69
10.6.3	Facilities.....	69
10.6.4	Weather .....	69
10.6.5	Software .....	69
10.6.6	NAS System.....	70
10.7	Human/System Interface.....	71
10.7.1	Air Traffic Controller Interface .....	71
10.7.2	Flight Deck Interface .....	72
10.7.3	Airline Operations Center (AOC) Interface.....	72
10.7.4	Traffic Flow Management (TFM) Interface .....	72
10.8	Development Challenges .....	73
10.8.1	Technical Challenges.....	73
10.8.2	Other Challenges.....	73
10.9	Future Multiplexer Capabilities .....	73
11	References.....	74
APPENDIX A – Simmod PRO! Software Tool .....		1
1	Document Purpose .....	2
2	Introduction.....	2
3	Applications .....	2
4	Features .....	2
5	Benefits .....	3
6	Hardware Requirements.....	3

7	Simmod Product Description .....	3
8	How SIMMOD Works.....	7
8.1	SIMMOD is a Discrete-Event Simulation .....	7
8.2	The Event Schedule .....	8
8.3	External Events .....	8
8.4	The Simulation Clock .....	9
8.5	SIMMOD Nodes and Links: Basic Modeling Components .....	9
8.6	Stochastic Processes.....	9
8.7	Random Linear Variables .....	10
8.8	Random Number Streams and Seeds.....	11
9	Flights .....	12
9.1	Arrivals .....	12
	Turnaround.....	12
	Overflights .....	13
9.2	Degenerate airports .....	13
9.3	Departures .....	13
9.4	Multiple, Stochastically Generated Flights.....	14
9.5	Cloning.....	14
10	Aircraft Definition .....	15
10.1	Aircraft Models.....	15
10.2	Aircraft Groups in Airspace.....	16
	Aircraft Separation in Airspace.....	16
	Aircraft Groups on the Airfield.....	17
11	Airspace Structure.....	17
11.1	Routes .....	18
	Nodes and Node Types .....	19
	Holding .....	19
	Holding Stack Characteristics.....	20
	Interface Nodes .....	20
	Links and Link Types .....	20
	Link Speed Ranges .....	20
11.2	Wind Sets .....	22
11.3	Sectors.....	22
11.4	Plans.....	23
12	Airspace Logic .....	23
12.1	Aircraft Movement Rules on the Links .....	23
	Link Types and Aircraft Speeds.....	23
	Link Time for Vectoring or Path Stretching .....	24
	Wind .....	24
	Passing and Wake Turbulence Along a Link.....	24
12.2	Aircraft Movement Control at the Nodes .....	25
	Airspace Holding .....	26
	Airspace Holding Stacks Characteristics .....	26
	Airspace Holding Strategy .....	26
12.3	Aircraft Movement Control Strategies.....	28
12.4	Level I: Node Arrival Control Strategies.....	29

	Type 1: QFIFO Node Arrival Control .....	29
	Order of Delays Generated for QFIFO .....	29
12.5	Level I: Node Arrival Control Strategies, Continued .....	31
	Type 2: SpeedFit Node Arrival Control.....	31
12.6	Level I: Node Arrival Control Strategies, Continued .....	35
	Type 3: MultiFit Node Arrival Control.....	35
12.7	Level II: Metering Strategy.....	38
	Metering Node Structure.....	39
	Meter Logic.....	39
13	Airfield Logic.....	41
13.1	Runways.....	42
	Runway Exits .....	42
	Runway Blockage .....	43
	Runway Crossing Priority.....	43
	Runway Crossing Times .....	43
13.2	Gates .....	44
	Gate Ownership .....	45
	Multiple Gates Modeled as One .....	45
	Grouping Gates into Concourses .....	45
	Gate Blocking and Gate Restrictions.....	45
	Gate Logic.....	46
13.3	Links .....	46
	Taxipaths and TaxiPlanning.....	46
	Optimized TaxiPlans.....	47
	Taxipaths and Link Characteristics.....	48
	Assigning Taxipaths Depending on Gate, Runway, and Arrival/Departure Status .....	48
	Assigning Taxipaths Depending on Gate, Towing, and Arrival/Departure Status (Towing aircraft only).....	48
13.4	Departure Queues.....	49
13.5	Aircraft Groups .....	50
	Landing-Roll Distance and Time.....	50
	Takeoff Roll Distance and Time.....	51
	Power-Back and Push-Back Aircraft.....	51
13.6	Congestion, Holdcycles, Checkpoints, and Staging Areas for Arrivals .....	51
	Staging Areas, Congestion and Taxi Checkpoints for Departures.....	52
13.7	Departure De-icing.....	52
13.8	Flight Banks .....	53
13.9	DSDPaths.....	54
14	Interface Logic .....	54
14.1	Procedures.....	54
	Landing .....	55
	Missed Approach .....	55
	Takeoff.....	55
	Related Procedures.....	56
14.2	Arrival Spreading.....	57
14.3	Staggered Runway Approaches .....	58

14.4	Dynamic Airspace Re-routing Due to Runway Congestion .....	59
14.5	Dynamic Departure Re-routing Due to Runway Congestion .....	60
14.6	Touch-and-Go Operations .....	60
15	Resetting Simulation Parameters .....	60
15.1	Flight Cloning (SETCLONE).....	61
15.2	Changing Gate Characteristics (SETGATE) .....	61
15.3	Inhibiting Injections (SETINJ) .....	61
15.4	Changing Link Characteristics (SETLINK) .....	61
15.5	Changing Meter Post Node Characteristics (SETMETER).....	61
15.6	Changing Node Characteristics (SETNODE).....	61
15.7	Changing Plans (SETPLAN) .....	62
15.8	Inhibiting Procedures (SETPROC).....	62
15.9	Changing Route Characteristics (SETROUTE).....	62
15.10	Changing Runway Characteristics (SETRWY).....	62
15.11	Changing Sector Characteristics (SETSECT) .....	62
15.12	Controlling Touch-and-Go (SETTNG).....	62
15.13	Changing the Wind Characteristics (SETWIND).....	63
15.14	Changing the Weather (SETWX).....	63
15.15	Setting Conditions for a Forced Runway Crossing (SETXNG).....	63
16	Stochastic Processes.....	63
16.1	Random Linear Variables .....	63
16.2	Random Number Streams and Seeds.....	64
16.3	Stochastic Process Descriptions.....	65
	Gate Service (Occupancy) Times for Arrivals and Departures .....	65
	Multiple Arrivals and Departures .....	65
	Landing and Takeoff Roll Distances .....	65
	Intrail (Inter-Aircraft) Separation Multiplier .....	65
	Lateness of Scheduled Flights.....	66
	Bank Holding Probability and Transfer Time.....	66
	Arrival and Departure Clone Generation .....	66
APPENDIX B – Generic Metroplex Scheduling Study .....		1
Introduction.....		2
Spanning Set Development Methodology .....		2
Most-Used Full and 2-Runway Metroplex Configurations .....		3
	Configurations for New York Metroplex 1: JFK, LGA .....	4
	Configurations for New York Metroplex 2: EWR, TEB .....	5
	Configurations for San Francisco Metroplex: SFO, OAK.....	5
	Configurations for Dallas Metroplex: DFW, DAL.....	6
	Configurations for Los Angeles Metroplex: LAX, LGB.....	6
	Configurations for Miami Metroplex: MIA, FLL.....	7
	Configurations for DC Metroplex: IAD, DCA .....	7
	Configurations for Chicago Metroplex: ORD, MDW .....	8
	Configurations for Houston Metroplex: IAH, HOU.....	8
Parameterization of Runway Layout Geometries.....		9
	Full Set of Parameter Definitions for Runway Layout Geometries.....	9

Reduced Set of Parameter Definitions for Two-Airport Geometries .....	10
Results: Spanning Set of Metroplex Runway Layouts .....	11
Metroplex Runway Layout Group 1: EWR/TEB .....	12
Metroplex Runway Layout Group 2: DFW/DAL .....	12
Metroplex Runway Layout Group 3: JFK/LGA .....	13
Metroplex Runway Layout Group 4: ORD/MDW .....	14
Metroplex Runway Layout Group 5: IAH/HOU .....	14
Metroplex Runway Layout Group 6: MIA/FLL .....	15
Metroplex Runway Layout Group 7: Half SFO/SJC .....	15
Arrival/Departure Paths for Generic Metroplexes .....	16
Generic Metroplex Runway and Airspace Geometries .....	16
Construction Method for Lateral Arrival and Departure Paths .....	16
Results: Lateral Arrival and Departure Paths for Generic Metroplexes .....	16
Altitude Profiles for Arrival and Departure Paths .....	20
Traffic Demand for Generic Metroplexes.....	21
Terminal Airspace Complexity for Generic Metroplexes .....	22
Metroplex Flow Envelope Intersection Metric Definition.....	22
Aircraft Flow Envelopes .....	22
Generic Metroplex Flow Envelopes .....	23
Results: Generic Metroplex Flow Envelope Intersections.....	24
Volume Intersections and Geometry Effects for Flow Shape 1.....	24
Volume Intersections and Geometry Effects for Flow Shape 2.....	26
Volume Intersections and Geometry Effects for Flow Shape 3.....	27
Overhead Views of Volume Intersections for all Generic Metroplexes using Flow Shape 1 and Fix Geometry 1 .....	29
Side View of Volume Intersections for Dallas Metroplex using Flow Shape 1 and Fix Geometry 1.....	31
Overhead Views of Volume Intersections for all Generic Metroplexes using Flow Shape 1 and Fix Geometry 3 .....	32
Volume Intersections for Deconflicted vs. Nondeconflicted Arrival/Departure Paths .....	34
Overhead Views of Flow Volume Intersections for Deconflicted vs. Original Arrival/Departure Routes.....	35
Generic Metroplex Environmental Impact Analysis .....	37
NASEIM Fuel Burn Calculations .....	37
NASEIM Emissions Calculations.....	37
Generic Metroplex Inputs to NASEIM.....	38
Results for the Generic Metroplex Environmental Impact Study.....	38
Environmental Impact Results for Spatially Deconflicted vs. Original Arrival/Departure Routings .....	39
Conclusions.....	41
Appendix C – Generic Metroplex Geometry Analysis.....	1
Overview.....	2
Analysis Methodology .....	3
Results.....	3

# 1 Introduction

The consensus amongst government, industry, and academic stakeholders is that there will be a significant increase in air traffic demand within the National Airspace System (NAS) by the time the Next Generation Air Transportation System (NextGen) is operational [AF09, BC06]. Much of the projected demand growth will be in the form of traffic to and from major metropolitan areas, as history has shown that they are the nucleus for both population and economic growth. Thus, even if additional airports are built to accommodate the increased traffic, the airspace above major metropolitan areas will be far more crowded than they are today, and the interactions between traffic flows will be more frequent and more consequential.

## 1.1 Metroplex Definition

A metroplex is defined by the Joint Planning and Development Office (JPDO) as a metropolitan area with high traffic demand that is served by two or more airports with arrival and departure operations that are highly interdependent [JPDO07]. The projected traffic growth will therefore increase the coupling of operations in the metroplexes that already exist, and potentially create new metroplexes. In fact, the FAA predicts that over the next 20 years, U.S. population and economic growth are expected to be concentrated in 15 metropolitan areas. These metropolitan areas are listed in Table 1.

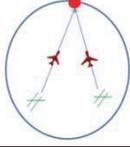
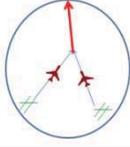
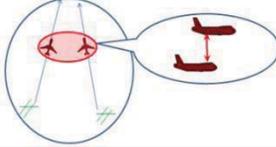
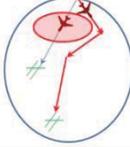
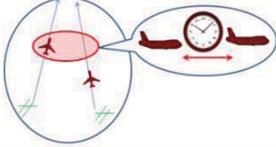
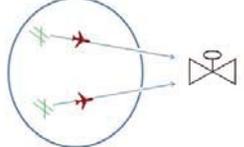
**Table 1: OEP15 Airports Anticipated as Metroplexes**

Airport	LAS	JFK	FLL	EWR	SAN	ORD	LAX	CLT	ATL	BOS	SFO	DTW	LGA	DFW
3X demand to 2015 capacity ratio	3.94	3.03	2.9	2.58	2.36	2.27	2.02	1.75	1.71	1.58	1.57	1.37	1.27	1.07
FACT-2 Report problem metro by year	2007	-	√	-	√	-	-	-	-	-	-	√	-	
	2015	√	√	-	√	-	√	√	√	√	-	√	-	
	2025	√	√	√	√	√	√	√	√	√	-	√	-	

## 1.2 Metroplex Interdependencies

In our prior work, we identified six types of interdependencies between traffic flows in a metroplex based on observations from metroplex site visits and traffic flow data analysis [RE09]. These interdependencies were found to result from the sharing of common fixes, paths or airspace volumes within the metroplex (i.e. metroplex resources) by different traffic flows, or the sharing of common downstream traffic flow restrictions. The six different types of interdependencies are listed and defined in the table below.

**Table 2: Major Metroplex Interdependencies**

#	Diagram	Definition
1		Arrivals/Departures to/from two or more proximate airports using the <i>same points in the airspace</i> – Arrival/Departure fixes
2		Arrivals/Departures to/from two or more proximate airports using <i>common path segments</i> – STARs and SIDs
3		Arrivals/Departures to/from two or more proximate airports intend to use the <i>same volume of airspace</i> but with <i>vertical separation</i>
4		Arrivals/Departures to/from two or more proximate airports intend to use the <i>same volume of airspace</i> but with <i>lateral separation</i>
5		Arrivals/Departures to/from two or more proximate airports intend to use the <i>same volume of airspace</i> but with <i>temporal separation</i>
6		<i>Downstream restrictions</i> , applied across multiple airports in the metroplex

The aforementioned interdependencies may be managed in one of two ways. Either, the traffic flows are physically separated, by lengthening the paths of some or all flights, such that different traffic flows will traverse different volumes of airspace; or the traffic flows are coordinated, by regulating the entry time and speed profile of the flights that are transiting the metroplex, such that the constituent traffic flows remain conflict-free [RE09]. The former is referred to as spatial control while the latter is referred to as temporal control. Examples of temporal control include holding and speed control.

### 1.3 Research Focus

Prior analysis of the N90 (New York) metroplex showed convincingly that, while scheduling (temporal control) and airspace redesign to separate traffic flows (spatial control) are synergistic, if one had to choose between the two then scheduling is far more effective from the viewpoint of reducing delays. The superior performance of scheduling is illustrated in Figure 1 below, where the average delay per flight for the four different combinations of airspace re-design and scheduling are shown in the form of histograms for a representative set of runways in the New

York terminal area. As is readily seen, there are significant reductions in delay whenever scheduling is utilized. For example, the average delay for the Current Airspace is significantly lower when scheduled than when it is not scheduled. Likewise, the delay for the NextGen Airspace is significantly lower when scheduled than the delay when it is not scheduled. Further, the reduction in delay when both airspace redesign and scheduling are utilized is slightly greater than the individual reduction when both are utilized separately, i.e. they are synergistic. This may be seen by the fact that the difference between the delay for the scheduled NextGen Airspace and the delay for the unscheduled Current Airspace is greater than the sum of the difference between the delay for the scheduled NextGen Airspace and the delay for the unscheduled NextGen Airspace and the difference between the delay for the unscheduled Current Airspace and the delay for the unscheduled NextGen Airspace.

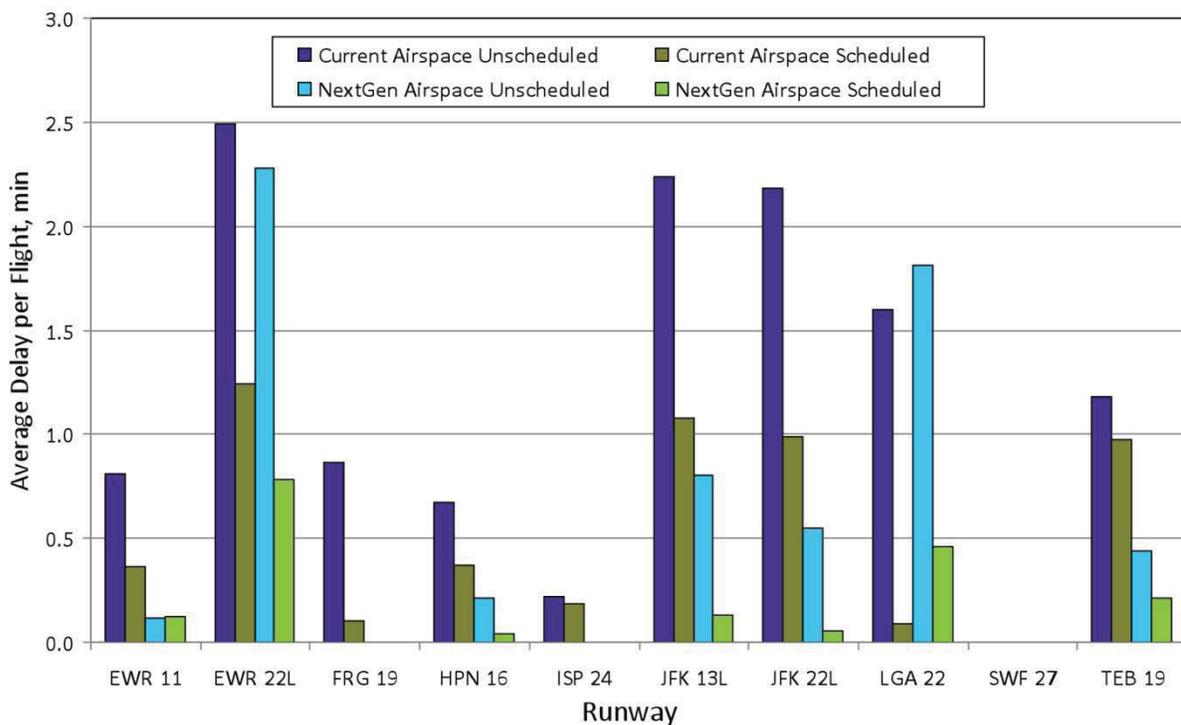


Figure 1: Benefits of Spatial versus Temporal Control in N90 (taken from [RE09])

Given these results, our team decided to focus on the development of a scheduling algorithm for metroplex operations. The name given to the tool that this algorithm would inhabit is the Multiplexer.

## 1.4 Structure of Report

The report is structured as follows. In the following section, Section 2, we list all the pertinent abbreviations and acronyms used throughout the document. In Section 3, we describe the “Multiplexer” algorithm that has been developed and present a brief validation. In Section 4, we describe the emulation of TMA that was developed to represent the current state of practice in terminal area scheduling, along with a brief validation of its fidelity. In Section 5, we present the sets of geometries that were used to evaluate the performance of the algorithms. The first set of

geometries includes fourteen (14) generic geometries that span the metroplex geometries observed in the NAS. The second set of geometries includes the two N90 (New York) geometries (current and future geometries) that were developed in our prior work. In Section 6, we present the analysis framework followed in Section 7 by the details of the evaluation test-bed. Results are presented in Section 8 followed by a summary of our findings in Section 9. In the last substantive section of the report, Section 10, we provide our vision for how the Multiplexer might be deployed by way of a concept of operations.

## 2 Abbreviations/Acronyms

ACARS	Aircraft Communications Addressing and Reporting System
ADSB	Automatic Dependent Surveillance - Broadcast
AF	Arrival Fix
ANSP	Air Navigation Service Provider
AOC	Airline Operations Center
ARTCC	Air Route Traffic Control Center
ARTS	Automated Radar Terminal System
ASDE-X	Airport Surface Detection Equipment - Model X
ATA	Actual Time of Arrival
ATCT	Air Traffic Control Tower
BADA	Eurocontrol Base of Aircraft Data
CAS	Calibrated Air Speed
CDA	Continuous Descent Arrival
CNS	Communications Navigation and Surveillance
CTAS	Center TRACON Automation System
DP	Dynamic Planner
DPs	Departure Procedures
DST	Decision Support Tool
ERAM	En Route Automation Modernization
ETA	Estimated Time of Arrival
ETMS	Enhanced Traffic Management System
FAA	Federal Aviation Administration
FAF	Final Approach Fix
FCFS	First Come First Serve
FIFO	First- In-First-Out
IFR	Instrument Flight Rules
ILS	Instrument Landing System
JPDO	Joint Planning and Development Office
NAS	National Airspace System
NEXTGEN	Next Generation Air Transportation System
Nm	Nautical mile
PBN	Performance Based Navigation
RNAV	Required Navigation
RNP	Required Navigation Performance
RTA	Required Time of Arrival
RTCA	Radio Technical Commission for Aeronautics
RUC	Rapid Update Cycle
SID	Standard Instrument Departure
STA	Scheduled Time of Arrival
STAR	Standard Terminal Arrival
TAF	Terminal Area Forecast
TARGETS	Terminal Area Route Generation and Traffic Simulation
TASAT	The Tool of Separation and Analysis of Throughput
TDMA	Time Division Multiple Access
TFMS	Traffic Flow Management System

TMA	Traffic Management Advisor
TMC	Traffic Management Coordinators
TRACON	Terminal Radar Approach Control
TS	Traffic Synthesis
VFR	Visual Flight Rules

### 3 Multiplexer Algorithm

Application of a scheduling algorithm that does not fully account for the interactions between traffic flows can result in a rapid buildup of delay that gets pushed back into the en route airspace. Thus, if a scheduling algorithm is to be used in dense operations, we must first make sure that any savings in time or fuel that are gained within the metroplex are not negated by the added cost of the delay that is pushed back into the NAS as a whole (by way of en route delays).

#### 3.1 Objective

The multiplexer algorithm is formulated as a mixed integer linear program for the scheduling of aircraft arrivals and departures. This schedule format is in terms of arrival and departure fix crossing times which allows for future changes in the objective to be minimized (fuel burn, etc.) with minimal changes to the scheduling program.

The current primary objective of the set of programs that have been developed is to minimize the change between the estimated times of arrival and departures with the computed times within the metroplex. The arrival changes are between the reported ETA to an arrival fix and the computed arrival time at the fix while the departure changes are changes to the estimated time to the departure fix and the computed time to the fix. This objective can be written as:

$$\min \sum_{i \in \text{Arrivals}} |\text{eta}_i - \text{sta}_i| + \sum_{j \in \text{Departures}} |\text{etd}_j - \text{std}_j|$$

Where  $\text{eta}_i$  or  $\text{etd}_i$  is the estimated time of the  $i^{\text{th}}$  aircraft (which is given as an input to the program) and  $\text{sta}_i$  or  $\text{std}_i$  is the scheduled time of the  $i^{\text{th}}$  aircraft (which is computed by the program). Elsewhere in this formulation this  $\text{sta}$  constraint will be referred to simply as  $t$ .

The goal for this set of optimization programs is to minimize the difference in the expected estimated time of each aircraft and the scheduled time producing an RTA such that the delays inside the metroplex could be reduced while not drastically increasing the delay absorbed en route. Since the optimization problem is set to minimize a sum of absolute values, there will be some cases where an aircraft is required to increase speed to meet a more optimal scheduled time. To prevent an unreasonable speed change, a constraint was added to each program such that no aircraft could be asked to push forward its estimated time by more than 1.2 minutes. This parameter is entirely configurable, but was chosen to represent the time change resulting from a 10 knot speed increase over a 500 Nm distance.

### 3.2 Algorithm 1: Staged Optimization for Arrival Fix and Runway Operations

This first optimization program is a simple linear programming approach that is very fast but does not account for swapping and can only push back to account for violated constraints in the metroplex. This optimization program is two staged where the problem is separated into two sub problems and solved sequentially. This algorithm was developed in a previous metroplex project, but is included here for direct comparison.

The first problem in the series was to schedule the runway separations. Each runway was treated as a separate problem. The aircraft order was determined by the ETA to the runway with the following constraints:

$$t_{i+1} - t_i \geq sep_{i,i+1}^{runway,k} - transit_j + transit_i$$

Where  $t$  is the time at the runway and the separation is extracted from a table lookup based on aircraft weight class and arrival or departure status. This method relies on the sequence of aircraft being sorted and can only push the STA back from the ETA. It cannot swap aircraft. The output for this algorithm was saved in the variable  $t^{runway}$ .

The second stage problems relied on the first stage problems to provide an intermediate schedule time. This schedule time was then used to build an earliest constraint. The constraint followed a similar format as the previous one, with the additional constraint that the scheduled time at the entry fix could not be prior to the required time generated by the runway constraints:

$$\begin{aligned} t_{i+1} - t_i &\geq sep_{i,i+1}^{entry,k} \\ t_i &\geq t_i^{runway} - 1.2 \\ t_i &\geq eta_i \end{aligned}$$

Since the scheduled times resulting from this simple algorithm can only be greater than or equal to the ETA, the objective can be simplified to:

$$\min \sum t_i$$

This formulation does not adequately account for the interaction between aircraft that share an arrival fix but go to different runways, but provides a simple and fast method for estimating delay values.

### 3.3 Algorithm 2: Joint Optimization for Airspace Fixes and Runway Operations

To more closely model the interactions between the metroplexes and provide scheduling solutions that do not push excessive delay into the en route flight regime, a more complete optimization program was developed.

### 3.3.1 Constraints Used to Model Metroplex Airspace

While our objective serves to provide the search direction for our optimization program, we must provide optimization constraints that accurately model the airspace constraints encountered by controllers to ensure that the algorithm generated STAs do not further complicate the problem. Optimization constraints are needed to prevent the trivial solution where an unchanged STA and ETA would be considered optimal. To accomplish this, both a runway constraint and a fix constraint were added as resource constraints.

### 3.3.2 Airspace Fix Constraints

The airspace fix constraints were built as follows:

Let  $A$  be the set of aircraft that use the specific airspace fix in question. Then  $A \times A$  constraints were generated from the following equation:

$$t_j - t_i - Mx_{i,j} \geq sep_j - M, \quad \forall i, j \in A \times A, \quad i \neq j$$

Where  $t_i$  is the scheduled time of arrival for the  $i^{\text{th}}$  aircraft.

The variable  $sep_j$  is the required time separation (5 Nm divided by the speed of the trailing aircraft  $j$ ). The variable  $x$  is a binary decision variable used to determine the leading and trailing aircraft pair. The value of  $x$  is one if aircraft  $i$  follows aircraft  $j$  and is 0 otherwise. The variable  $M$  is a sufficiently large variable that is used to ensure that the binary constraint is enforced. This is called the big M method and is a very common tool for modeling complex optimization programs. For this problem where we are optimizing a full day of traffic, a value of 3600 minutes was used.

This is a powerful method for generating constraints due to the ability of the program to allow for swapping the order of aircraft at each fix if swapping will lower the overall objective. The only limitation to this optimization method is that the number of constraints grows quadratically with the number of aircraft that use each individual resource. The worst case scenario for problem size considerations will occur when the majority of the traffic uses the same fix.

### 3.3.3 Runway Constraints

The runway constraints are generated in a similar manner. The primary differences are that the separation value is computed using a lookup table and that the scheduled times have to be modified by adding transit times. This constraint shown in the following equation:

$$t_j - t_i - Mx_{i,j}^{runway,k} \geq sep_{i,j}^{runway} - M - transit_j + transit_i$$

Where the transit times are factored into the equation and the separation is extracted from a table based on aircraft weight class and arrival or departure status. It should be noted here that for this algorithm to work with departures, the transit time should be negative and added to the resulting  $t$  to provide the proper departure time.

This formulation shares the same strengths and weaknesses as the airspace fix formulation. It allows for swapping when more optimal but at the cost of growing quadratically the number of constraints.

### 3.3.4 Solution Methodology

The solution methodology evolved with the project due to increasing problem sizes and runtime constraints. The first solution algorithm was to simply build the entire problem for the whole metroplex and solve. This extends the algorithm defined as Algorithm 1 by including swapping via the binary decision variables. However, the solution times were greatly increasing as the number of aircraft increased, and due to the binary variables, the solution times were large. While this problem could be solved given sufficient memory and computational resources, to increase efficiency, several math programming techniques were used.

To solve our mixed integer linear programming scheduling problem, a rolling horizon solution was used in conjunction with a Bender's decomposition scheme. This separated the scheduling into one hour pieces of the schedule to be solved sequentially. This rolling horizon solution method is commonly used and allows for a continuous solution case where every hour the next hour's schedule is optimized. If the hour is found to be too long or short, it can easily be adjusted to accommodate. The Bender's decomposition breaks the single large problem required to schedule a full hour of dense operations into separate sub-problems and master problems similar to the staged optimization procedure. The more difficult runway problems become sub-problems to the entry fix master problem. It takes the two sets of mixed integer constraints and turns them into problems, but instead of solving them sequentially, it solves them both iteratively, passing constraints up from the sub problems to the master problem until the solution converges.

### 3.3.5 Solution Algorithm

To solve the problem using the defined methodology, the algorithm sequence was as follows:

- Solve the master problems (entry fix problems) using CPLEX to generate an initial set of STAs (called  $t$  in the constraint equations).
- Take STA values and generate sub-problems (runway constraint problems) to solve for the feasibility of each runway schedule. Solve using CPLEX.
- If a sub-problem is infeasible, the dual problem will be unbounded. Use the unbounded ray to generate a constraint for the entry fix STAs following the standard Bender's scheme. Add this constraint to the master problem.
- Resolve the master problem with the updated constraints using CPLEX and rebuild the sub-problems to recheck for feasibility of each runway schedule.

- If every runway schedule is feasible, the algorithm has converged. Otherwise, iterate until converged.

## **4 TMA Algorithm**

The baseline scheduling capability assumed in the previous Metroplex NRA was a first-in-first-out scheduling algorithm. Today, there are time-based arrival scheduling utilities such as the Traffic Management Advisor (TMA) in place in most ARTCCs. In addition, controllers also use a mental model of arrival fix sequences, TRACON traversal times, and the resulting landing times to enforce efficient and conflict-free arrival fix crossing and runway landing sequences. Additionally, there are departure sequencing and scheduling aids such as the Departure Sequencing Program (DSP), which is used in the New York TRACON for managing departure sequences over shared departure fixes. As a result, a FIFO scheduler is not a fair depiction of the arrival scheduling process in current-day terminal operations. Our Metroplex NRA team plans to develop a TMA-emulation arrival scheduling algorithm to better represent current day baseline scheduling capabilities against which the team's Integrated Metroplex Scheduling Concept will be compared.

### **4.1 Background**

In the current air traffic system, the TMA is a decision support tool (DST) that assists the center Traffic Management Coordinators (TMCs) and controllers with planning and time-based scheduling of arrival traffic. TMA is a part of a suite of DSTs called the Center TRACON Automation System (CTAS). TMA is currently installed and functional at all 20 Air Route Traffic Control Centers (ARTCCs). TMA's time-based scheduling engine, called the Dynamic Planner (DP), is perhaps closest to the current state-of-the-art in multi-airport time-based scheduling controller aids. TMA's DP can handle up to five different airports within a TRACON and treats them as separate entities from a scheduling perspective. Runways at different airports are additionally treated as being independent of each other. Section 3 of this document discusses the development of a proxy model of TMA's DP scheduling capability and its application as a baseline for the assessment of a NextGen scheduling capability as applied to a set of generic metroplex geometries that cover the entire span of arrival geometry-types found in the National Airspace System (NAS), as well as its application to the scheduling simulation for a New York metroplex model. A validation study was conducted using Sensis' in-house CTAS simulation capability to assess how closely it emulates the real TMA schedules.

### **4.2 Technical Approach and Algorithm Description**

Our technical approach was two-pronged. First, we leveraged existing literature which outlined the workings of TMA [EDG93, DE95] and the details of TMA's DP component [W00]. Second, we met with the principal developer of TMA – Mr. Harry Swenson of NASA Ames – to gain additional assistance in understanding the intricacies of the DP scheduling methodology. Based on the literature search and discussions with Mr. Swenson, we were able to develop a good understanding of TMA's scheduling process, and use it to develop the emulation. We outline the DP scheduling process in the following section.

### **4.3 TMA Scheduling Capability (Dynamic Planner) Description**

The DP assists the Traffic Management Coordinators (TMCs) and controllers with planning and scheduling of arrival traffic that is within 35 to 200 nm of the destination airport. The DP computes schedules that conform to constraints that are manually input by TMCs and reflect the operational and environmental conditions of the airports and airspace. The Center TRACON Automation System's (CTAs') Trajectory Synthesizer predicts the Estimated Times of Arrival (ETAs) at an outer meter arc, the meter fix, the final approach fix (FAF), and the runway threshold. These scheduling points are collectively called Reference Points. The typical arrival geometry at an airport having a four corner-post TRACON configuration is shown in Figure 2. The DP uses these generated ETAs to compute deconflicted STAs at the meter fix, and the runway threshold.

CTAS first predicts the ETAs to the runway and to the meter fix for all incoming aircraft within the center. These ETA estimates are based on the assumption that the aircraft will follow a nominal approach trajectory with no interference from other air traffic. TMA then creates STAs for the aircraft at the meter fix, retaining the first come, first served (FCFS) order of arrival, but delaying some aircraft to maintain the mandatory separations between successive arrivals at the meter fix. The TMC may alter the meter fix sequence by entering specific sequencing constraints and the DP will reschedule to conform to the input sequence constraints. The DP through a process called Runway Allocator also attempts to assign arrival aircraft to runways so as to reduce the overall delay. STAs at the meter fix and nominal fix-to-runway travel times are then used to generate ETAs at the runway. Runway STAs are then computed by deconflicting the runway ETAs in an optimized order using an Order of Consideration algorithm to satisfy constraints for wake-vortex separation, acceptance rate, and runway occupancy. If the delay assigned to a specific aircraft is greater than the capacity of the TRACON, then the excess delay is fed back to the center. ETAs to the meter fix are updated accordingly and the scheduling process is repeated [W00].

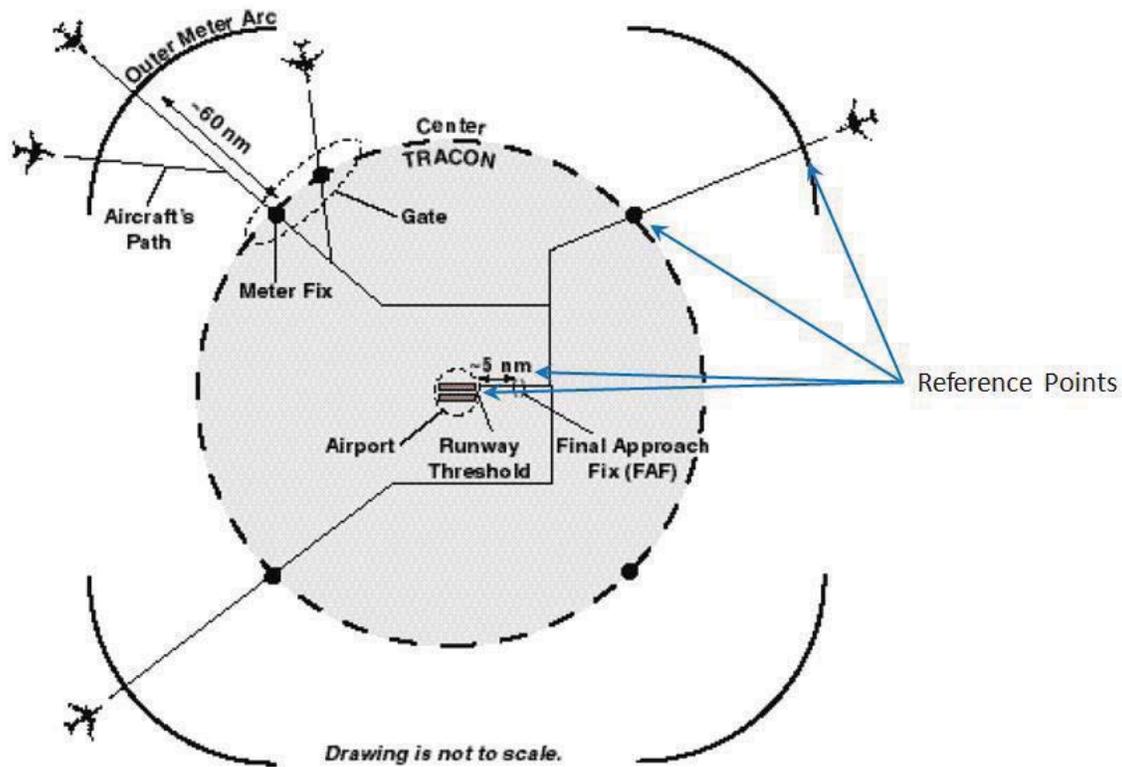
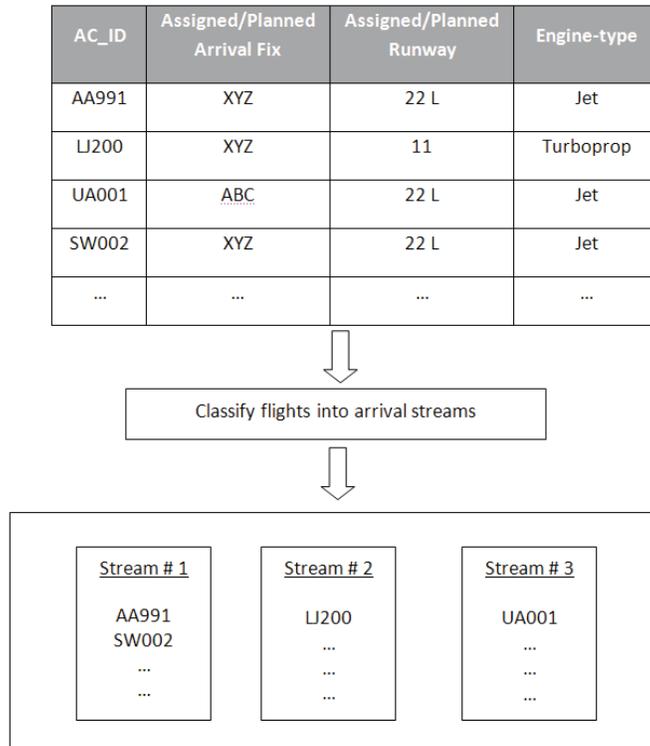


Figure 2: Typical Arrival Geometry with TMA Reference Points (taken from [W00])

#### 4.4 Stream Class Concept and Order of Consideration Algorithm

Once the runway ETAs are computed from the meter fix STAs, the DP then uses a modified sequence – conserving scheduling strategy to compute runway STAs. The aircraft enter the scheduling process based upon an Order of Consideration. To compute the Order of Consideration, aircraft with similar scheduling characteristics (engine type, destination airport and runway, and assigned arrival fix) are first grouped together into classes called stream-classes. For example, all aircraft crossing the same arrival fix and inbound to the same airport form one stream-class; all turbo-props crossing the same arrival fix and inbound to the same airport form another stream-class; all aircraft crossing the same arrival fix and inbound to another airport form another stream-class; all aircraft crossing another arrival fix form still another stream-class, and so on. Figure 2 shows a notional example of stream class classification.



**Figure 3: Classification of Flights into Arrival Stream-Classes**

The Order of Consideration algorithm ensures that aircraft within a single stream-class are sequenced to the runway in the same order as the order in which they are predicted to cross the arrival fix. However, aircraft belonging to different stream-classes may be re-sequenced with respect to each other between the arrival fix and the runway. That is, if an aircraft A belonging to stream class # 1 has its Estimated Arrival-fix Crossing Time later than that of aircraft B belonging to stream class # 2, but its Estimated Runway Landing Time is earlier than that of aircraft B, then aircraft A will be sequenced ahead of B to the runway. The order in which aircraft are selected to finalize their runway scheduled time of arrivals (STA) and Final Approach Fix (FAF) STA is determined by the following Order of Consideration algorithm.

The algorithm begins by determining the flight with the earliest meter fix STA within each stream class. Among these meter-fix-leaders-within-each-stream-class flights, the flight with the earliest runway ETA is selected as the next flight in the Order of Consideration. This flight has its runway and FAF STA computed by spacing it with respect to any previous landing on the runway and any necessary delay is absorbed in the TRACON if it is within the TRACON absorption limit. Excess delay, if any, is fed back to its meter-fix STA. This algorithm is repeated until all the flights have been scheduled to the runway. Please see [W00] for an example of Order of Consideration computation.

#### **4.5 TMA Scheduling Capability Emulation Description**

The TMA Emulator mimics the scheduling process outlined in the previous section. Since the purpose is to develop a TMA-like scheduler for multi-airport systems, the emulator expands the stream class concept as follows – all flights inbound to the same airport and using the same

arrival fix that have similar operational characteristics (i.e., the same engine type) are placed in the same stream class. The scheduling and order of consideration algorithms are also slightly adapted to apply to multi-airport systems and comprised of the following steps:

1. Start with estimated ETAs at the meter fix and deconflict in the FCFS order. This gives the initial meter fix STAs.
2. Select the flight with the earliest meter fix STA from each stream class.
3. Among these flights, "n" flights each having the earliest runway ETAs at the "n" metroplex airports are chosen as the next flights in the order of consideration.
4. If any two or more of these "n" flights are crossing the same arrival fix, then only one flight (= the leader at the arrival fix) is selected to be next in the order of consideration.
5. Runway and FAF STAs are computed for the selected flights. Any delay required to achieve the minimum required spacing for the preceding flight that is over or above the Allowed Mean Delay Threshold is fed back to the meter fix STA.
  - If delay is required to be fed back to the meter-fix, then all flights in the same stream class are vectored or held, if required, to maintain minimum separation at the arrival fix.
  - Reordering of the initial arrival fix crossing order, if required.
  - Spacing satisfied for all flights crossing the arrival fix.
6. Meter-fix and runway STAs for the selected flights are finalized.
7. The scheduled flights are removed from the processing list.
8. This process is repeated until all flights have been scheduled to the runway.

#### **4.6 Validation Approach**

Our approach for validating the TMA emulation is simulation based. A traffic demand set consisting of arrival traffic to the Denver International Airport (DEN) is processed through the Sensis in-house CTAS simulation. The TMA scheduling process working within CTAS acts on the same traffic demand set, predicting aircraft trajectories, deconflicting crossing times at the scheduling points, and providing the final TMA schedules for each flight in the input traffic demand set. The same traffic demand set is processed through the TMA scheduling emulator and the scheduling emulation final schedules are obtained. The two schedules are compared to determine if the fix crossing sequences and crossing times match.

#### **4.7 Validation Results**

The scenario used for initial validation was a simple one arrival-fix to one runway scheduling case for aircraft crossing the SAYGE arrival-fix to DEN runway 10. The purpose of this validation was to check if the sequence and timing of arrival-fix crossing produced by the TMA emulation was in agreement with what the scheduling algorithm within CTAS produced. As shown in Table 3 and Table 4, the TMA emulation produced the exact same sequence for arrival-fix crossings and the exact arrival-fix crossing times for most flights. As seen from Table 3, CTAS sometimes assigns a STA to a flight that is earlier than its ETA contrary to the algorithm

specifications and sometimes delays flights more than required to satisfy the minimum separation requirement (imposed as 7 nautical miles at the arrival-fix in the validation scenario). The TMA emulation also adheres to delaying past the ETA, and also spaces the flights at the arrival-fix by the minimum amount required (7 nm in-trail separation requirements which equates to approximately a 76 second separation crossing the arrival fix). As shown in the tables, the TMA emulation produced excellent results when compared to the CTAS scheduling algorithm.

**Table 3: Result of CTAS simulations for the validation scenario**

Flight ID	ACType	Weight Class	Engine Type	Arrival Fix	Arrival Runway	ArrFix ETA	ArrFix STA	ArrFix Spacing
COA1423	B744	H	J	SAYGE	10	18:41:55	18:41:43	
COA1427	B744	H	J	SAYGE	10	18:41:59	18:42:59	<b>0:01:16</b>
COA1432	B744	H	J	SAYGE	10	18:42:03	18:44:15	<b>0:01:16</b>
COA1435	B744	H	J	SAYGE	10	18:42:05	18:45:31	<b>0:01:16</b>
COA1439	B744	H	J	SAYGE	10	18:42:09	18:46:47	<b>0:01:16</b>
COA1441	B744	H	J	SAYGE	10	18:42:10	18:48:15	<b>0:01:28</b>
COA1440	B744	H	J	SAYGE	10	18:42:11	18:49:31	<b>0:01:16</b>
COA1448	B744	H	J	SAYGE	10	18:42:17	18:50:35	<b>0:01:04</b>
COA1422	B752	L	J	SAYGE	10	18:43:04	18:51:53	<b>0:01:18</b>
COA1429	B752	L	J	SAYGE	10	18:43:04	18:53:17	<b>0:01:24</b>
COA1428	B752	L	J	SAYGE	10	18:43:07	18:54:33	<b>0:01:16</b>
COA1431	B752	L	J	SAYGE	10	18:43:09	18:55:49	<b>0:01:16</b>
COA1438	B752	L	J	SAYGE	10	18:43:09	18:57:07	<b>0:01:18</b>
COA1424	B738	L	J	SAYGE	10	18:43:10	18:58:23	<b>0:01:16</b>
COA1421	B738	L	J	SAYGE	10	18:43:11	18:59:39	<b>0:01:16</b>
COA1426	A319	L	J	SAYGE	10	18:43:12	19:00:53	<b>0:01:14</b>
COA1442	B752	L	J	SAYGE	10	18:43:14	19:02:09	<b>0:01:16</b>
COA1425	A320	L	J	SAYGE	10	18:43:16	19:03:25	<b>0:01:16</b>
COA1433	B738	L	J	SAYGE	10	18:43:16	19:04:41	<b>0:01:16</b>
COA1434	B738	L	J	SAYGE	10	18:43:17	19:05:57	<b>0:01:16</b>
COA1444	B752	L	J	SAYGE	10	18:43:17	19:07:13	<b>0:01:16</b>
COA1430	A320	L	J	SAYGE	10	18:43:18	19:08:29	<b>0:01:16</b>
COA1437	B738	L	J	SAYGE	10	18:43:20	19:09:45	<b>0:01:16</b>
COA1449	B752	L	J	SAYGE	10	18:43:20	19:11:03	<b>0:01:18</b>
COA1436	A320	L	J	SAYGE	10	18:43:21	19:12:19	<b>0:01:16</b>
COA1443	B738	L	J	SAYGE	10	18:43:22	19:13:35	<b>0:01:16</b>
COA1447	B738	L	J	SAYGE	10	18:43:26	19:14:47	<b>0:01:12</b>
COA1445	A320	L	J	SAYGE	10	18:43:28	19:16:07	<b>0:01:20</b>
COA1446	A320	L	J	SAYGE	10	18:43:29	19:17:23	<b>0:01:16</b>
COA1451	B738	L	J	SAYGE	10	18:43:29	19:18:37	<b>0:01:14</b>
COA1450	A320	L	J	SAYGE	10	18:43:32	19:19:53	<b>0:01:16</b>

**Table 4: Results of TMA emulation scheduling simulation (compare arrival sequence, Arrival Fix STAs and Arrival Fix spacing against the data in Table 3)**

Flight ID	ACType	Weight Class	Engine Type	Arrival Fix	Arrival Runway	ArrFix ETA	ArrFix STA	ArrFix Spacing
COA1423	B744	H	J	SAYGE	10	18:41:55	18:41:55	
COA1427	B744	H	J	SAYGE	10	18:41:59	18:43:11	<b>0:01:16</b>
COA1432	B744	H	J	SAYGE	10	18:42:03	18:44:27	<b>0:01:16</b>
COA1435	B744	H	J	SAYGE	10	18:42:05	18:45:44	<b>0:01:17</b>
COA1439	B744	H	J	SAYGE	10	18:42:09	18:47:00	<b>0:01:16</b>
COA1441	B744	H	J	SAYGE	10	18:42:10	18:48:16	<b>0:01:16</b>
COA1440	B744	H	J	SAYGE	10	18:42:11	18:49:33	<b>0:01:17</b>
COA1448	B744	H	J	SAYGE	10	18:42:17	18:50:49	<b>0:01:16</b>
COA1422	B752	L	J	SAYGE	10	18:43:04	18:52:05	<b>0:01:16</b>
COA1429	B752	L	J	SAYGE	10	18:43:04	18:53:22	<b>0:01:17</b>
COA1428	B752	L	J	SAYGE	10	18:43:07	18:54:38	<b>0:01:16</b>
COA1431	B752	L	J	SAYGE	10	18:43:09	18:55:55	<b>0:01:17</b>
COA1438	B752	L	J	SAYGE	10	18:43:09	18:57:11	<b>0:01:16</b>
COA1424	B738	L	J	SAYGE	10	18:43:10	18:58:27	<b>0:01:16</b>
COA1421	B738	L	J	SAYGE	10	18:43:11	18:59:44	<b>0:01:17</b>
COA1426	A319	L	J	SAYGE	10	18:43:12	19:01:00	<b>0:01:16</b>
COA1442	B752	L	J	SAYGE	10	18:43:14	19:02:16	<b>0:01:16</b>
COA1425	A320	L	J	SAYGE	10	18:43:16	19:03:33	<b>0:01:17</b>
COA1433	B738	L	J	SAYGE	10	18:43:16	19:04:49	<b>0:01:16</b>
COA1434	B738	L	J	SAYGE	10	18:43:17	19:06:05	<b>0:01:16</b>
COA1444	B752	L	J	SAYGE	10	18:43:17	19:07:22	<b>0:01:17</b>
COA1430	A320	L	J	SAYGE	10	18:43:18	19:08:38	<b>0:01:16</b>
COA1437	B738	L	J	SAYGE	10	18:43:20	19:09:55	<b>0:01:17</b>
COA1449	B752	L	J	SAYGE	10	18:43:20	19:11:11	<b>0:01:16</b>
COA1436	A320	L	J	SAYGE	10	18:43:21	19:12:27	<b>0:01:16</b>
COA1443	B738	L	J	SAYGE	10	18:43:22	19:13:44	<b>0:01:17</b>
COA1447	B738	L	J	SAYGE	10	18:43:26	19:15:00	<b>0:01:16</b>
COA1445	A320	L	J	SAYGE	10	18:43:28	19:16:16	<b>0:01:16</b>
COA1446	A320	L	J	SAYGE	10	18:43:29	19:17:33	<b>0:01:17</b>
COA1451	B738	L	J	SAYGE	10	18:43:29	19:18:49	<b>0:01:16</b>
COA1450	A320	L	J	SAYGE	10	18:43:32	19:20:05	<b>0:01:16</b>

The team is currently continuing with more rigorous tests and the results will be presented in a paper at the AIAA Aviation Technology Integration and Operations Conference in September 2011.

## **5 Metroplex Geometries**

### **5.1 Generic Geometries**

Fourteen (14) Generic Metroplexes were developed based on two (2) arrival-departure-fix geometries from our previous work and seven (7) runway geometries that were determined in this effort to span the runway geometries of the NAS.

The set of runway geometries was developed by analyzing the runway geometries of current metroplexes, with the caveats that there are only two airports within each metroplex, with one arrival and one departure runway each. For the sake of both brevity and clarity, the details of this analysis have been reserved for Appendix A. The resulting lateral paths (i.e. the arrival/departure routes) for the complete set of 14 Generic Metroplex Airspace Geometries were constructed by combining the aforementioned runway geometries with either a single or a double corner-post-fix geometry (Geometries 1 and 3 from our previous Metroplex project). The lateral paths for the complete set of generic geometries are shown below in Figure 4 and Figure 5.

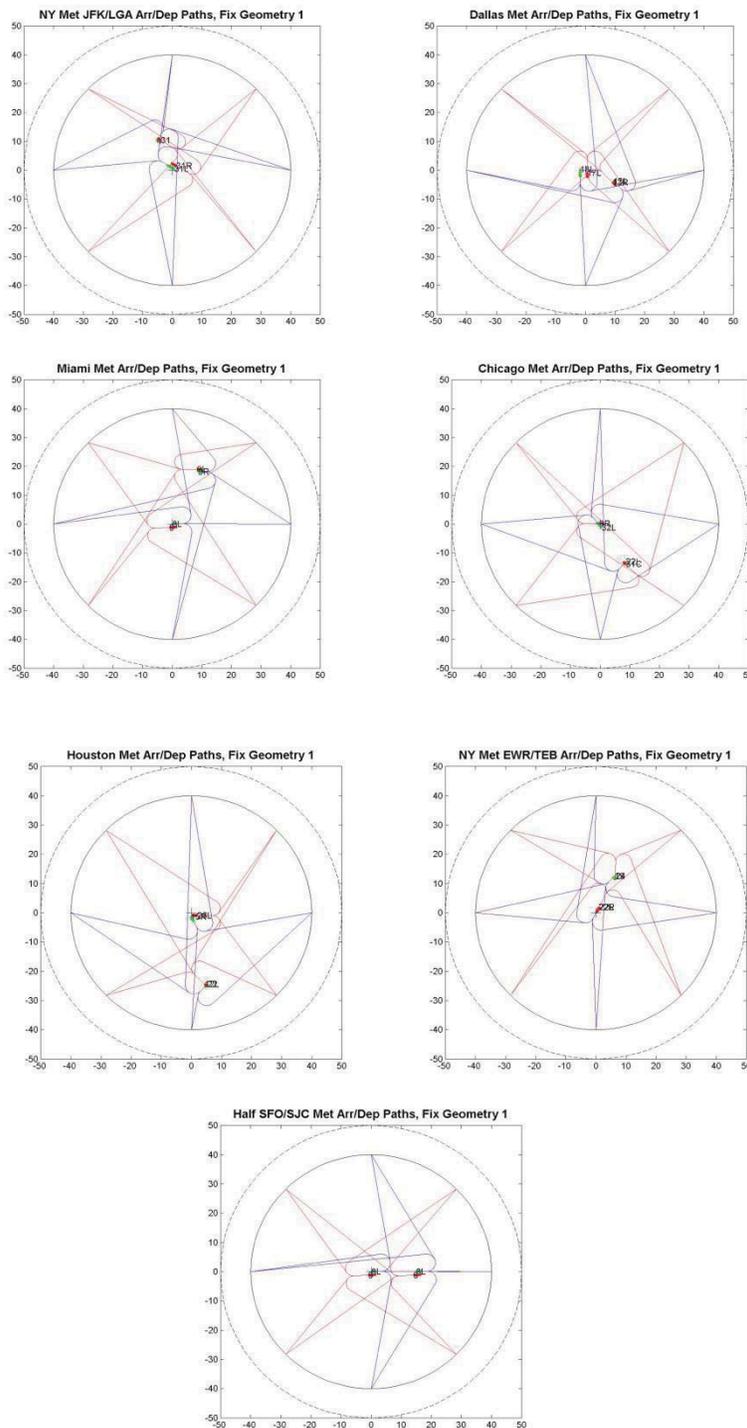
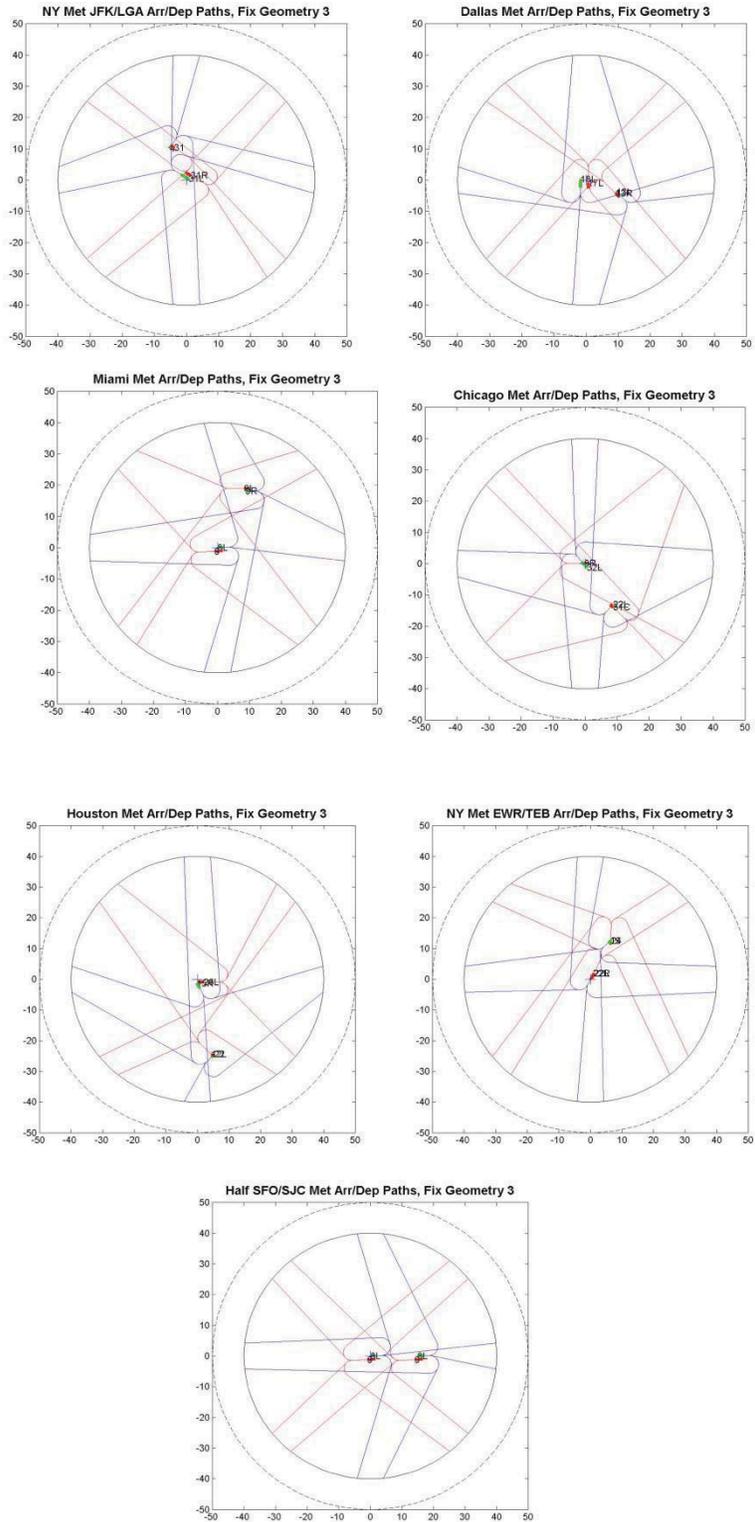


Figure 4: Lateral Paths for Generic Metroplexes with Single-Corner-Posts



**Figure 5: Lateral Paths for Generic Metroplexes with Double-Corner-Posts**

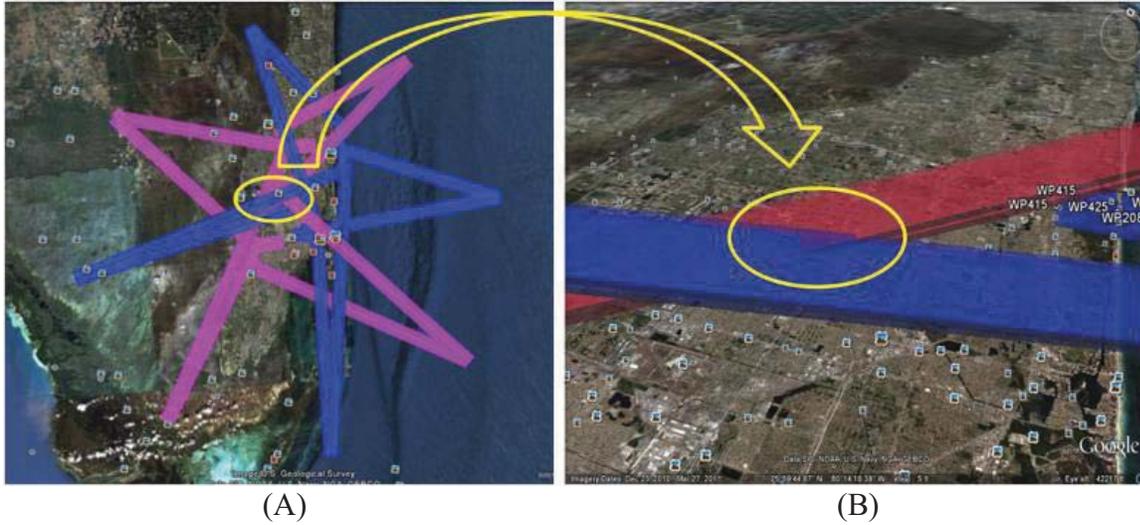
The lateral paths served as the baseline lateral routes for each metroplex geometry. The vertical trajectory from each arrival and departure fix were extended to or from the respective runways along these lateral routes. The arrival vertical profiles were developed using the Tool for Analysis of Separation and Throughput (TASAT), and the departure profiles were developed from high fidelity operational and performance data. The airspace environment applied in TASAT was for nominal wind speed and direction, with a range of respective landing weights and ambient temperature. The vertical trajectory generated by TASAT was a Continuous Descent Arrival (CDA) profile terminating at a point of intercept with the respective runway Instrument Landing System (ILS) glideslope with intercept requirements, and standard IFR approach criteria [Ref 1]. Since the use of a CDA profile has been shown to minimize fuel burn along with noise and emissions by eliminating level flight segments, the nominal arrival profiles are inherently optimized.

Since the arrival and departures were developed with a range of aircraft types/categories (small, large, B757, and Heavy) over a representative range of landing/takeoff weights, wind, and temperature, a range of vertical and lateral profiles were produced for each path in the metroplex. The highest and lowest altitude profiles, along with the maximum lateral path width for each route was taken to be the arrival/departure path maximum and minimum thresholds.

These arrival and departure procedures were then input into the FAA's Terminal Area Route Generation Evaluation and Traffic Simulation (TARGETS) tool. TARGETS further checks for the flyability of each arrival and departure path with built-in criteria with respect to altitude and speed changes, and turn radius constraints. In order to satisfy these flyability criteria, the following restrictions were imposed on the CDA profiles developed by TASAT:

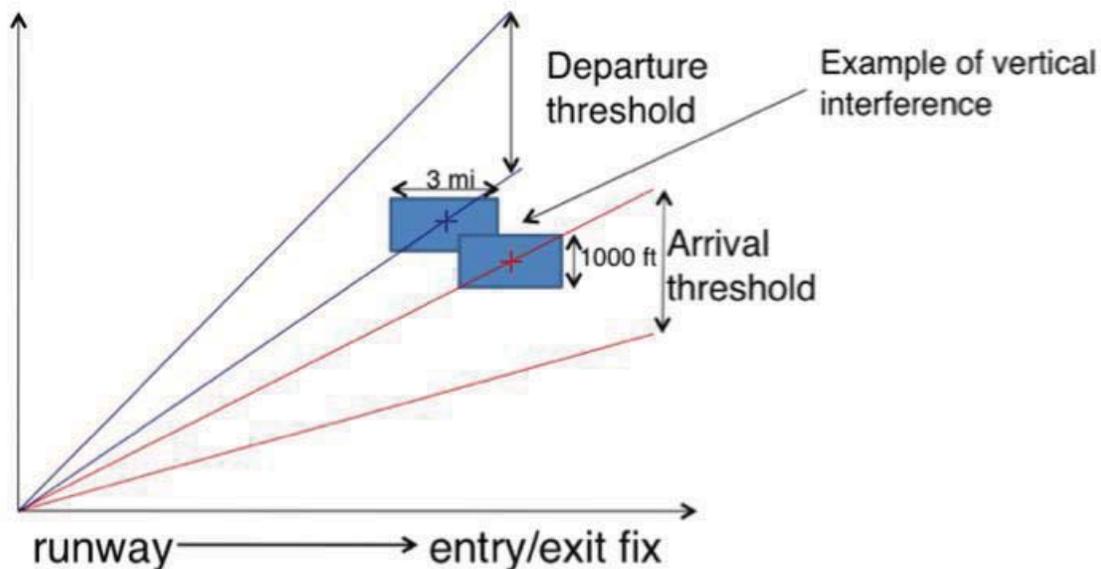
- During descent, the aircraft will maintain a flight path angle between 2.5 and 3.5 degrees.
- Until the aircraft reaches 10,000 ft, a descent speed of 300 knots will be maintained.
- A deceleration to result in a speed of 240 knots by 10,000 ft.
- A flight distance of 1 Nm is required for each 10 knot speed change.
- A flight distance of 1 Nm is required for each 318 feet of altitude change.
- When an arrival has a downwind leg, the speed restrictions are 210 knots for the turn onto the base-leg and 180 knots for the turn onto the final approach.
- The final approach fix is located 5 miles from runway threshold, at which point the aircraft is to intercept the ILS glideslope.

After development and meeting the flyability criteria in TARGETS, the baseline arrival and departure profiles were applied to all metroplex geometries. Figure A depicts the Miami metroplex Geometry 1 case study.



**Figure 6:** Aerial view of Miami Geometry 1 metroplex. On the right is an enlarged view of a typical conflict zone in this metroplex, where the upper threshold of the red arrival path intersects the lower threshold of the blue departure path.

As seen in Figure 6, where an aerial view of all the intersections in Miami Geometry 1 is present on the left in (a) and an expanded view of a single intersection if presented on the right in (b), there are several points/areas where the arrival and departure paths intersect. The next step in the metroplex design process was to deconflict these intersections in an optimized manner.

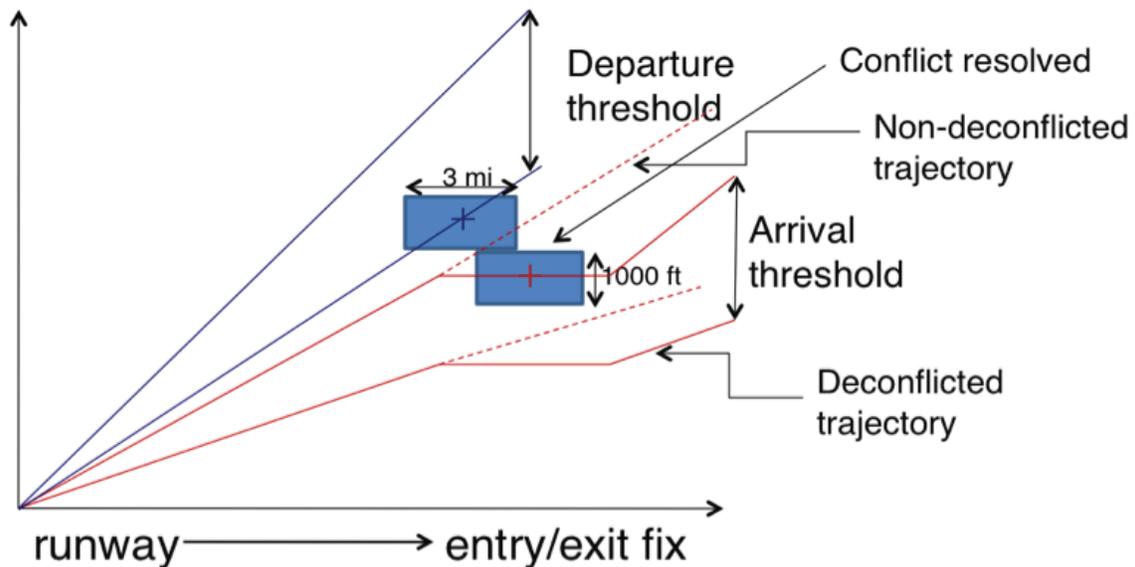


**Figure 7:** Example of an area of vertical interference.

A typical conflict region is annotated in Figure 7, where only the vertical trajectory is shown for an arrival and departure threshold. For discussion purposes, let the point of intersection between these arrival and departure paths be at the points denoted with the cross, '+', where the lower threshold of the departure trajectory intersects with the upper threshold of the arrival trajectory. Per ATC standard separation guidelines, a vertical separation of 1000 feet and a lateral

separation of 3 Nm is required between aircraft. In order to maintain this separation on all paths and completely deconflict the flow of traffic in the metroplex, cylinders with a diameter of 3 miles and height of 1000 feet are imposed on the upper and lower bounds of all trajectories. If any adjacent cylinders intersect, this region is treated as a conflict zone and deconfliction of the traffic is required. In Figure 7, the two blue intersecting squares can be viewed as cross sections of the described cylinders, centered at the cross '+' notations. As evident in this scenario, where these cylinders are 'just' in contact with each other is where the minimum separation requirements are met.

At each of these conflict zones, there are two deconfliction strategies to be considered. In Figure 8, the lower (arrival) trajectory was leveled off in order to maintain the required vertical separation. As can be seen, the descent gradient of the arrival is kept constant to facilitate determining location for the level-off, so that the edges of the two cylinders just 'touch' each other on the surface. This process guarantees the required separation is observed, while optimizing the location of the level-off. The alternative strategy would be to laterally move the arrival trajectory to avoid the conflict when feasible, given the geometry of the adjacent routes in the metroplex. Movement of the arrival trajectory, when possible, is considered more fuel efficient than movement of the departure trajectory due to the difference in the engine power settings involved.



**Figure 8:** Example of a vertical deconfliction by level-offs.

To optimize the deconflicted routes (where both methods of deconfliction were feasible), the altered arrival paths were input into TASAT to produce fuel burn results for comparison. The altered route with lower fuel burn results was chosen as the new deconflicted path. In addition, the new paths were again input into TARGETS to ensure the flyability criteria were met.

The transit times obtained from a fully deconflicted metroplex traffic flow were fed into both the TMA and Multiplexer algorithms as baseline transit times. The arrival transit times input in these algorithms are given in Table 5 below.

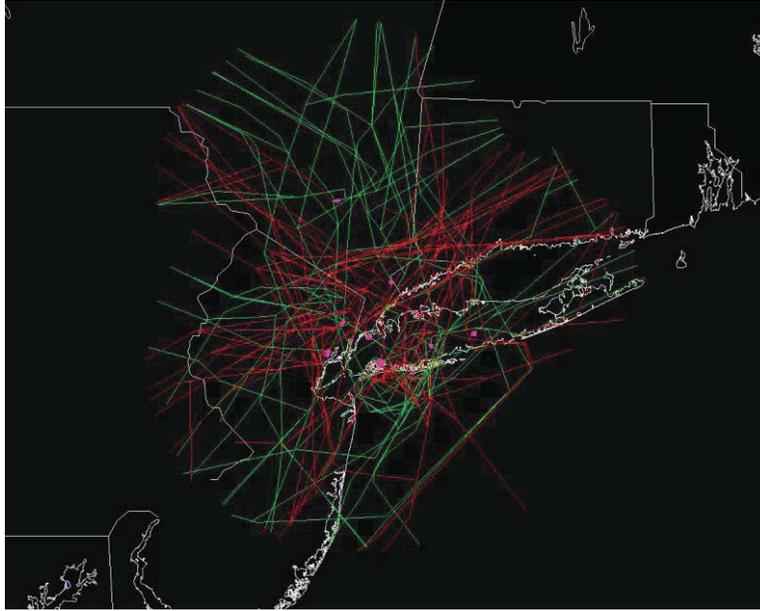
Table 5: Metroplex Transit Times

		Geometry 1			Geometry 3		
		Heavy Aircraft	Large Aircraft	Small Aircraft	Heavy Aircraft	Large Aircraft	Small Aircraft
	Route						
	MIA-FLL metroplex	MIA_NE	12.91	13.00	13.29	13.02	13.11
MIA_SE		12.91	13.00	13.28	12.65	12.74	13.02
MIA_SW		9.78	9.90	10.12	9.94	10.05	10.28
MIA_NW		9.80	9.92	10.14	9.67	9.80	10.01
FLL_NE		9.07	9.16	9.36	8.63	8.72	8.91
FLL_SE		14.48	14.57	14.89	13.94	14.04	14.35
FLL_SW		14.04	14.16	14.47	14.17	14.29	14.60
FLL_NW		9.20	9.33	9.54	11.91	12.04	12.31
SFO-SJC metroplex	SFO_NE	12.70	12.79	13.07	12.62	12.71	12.99
	SFO_SE	12.84	12.93	13.21	12.60	12.69	12.97
	SFO_SW	9.97	9.85	10.06	9.74	9.87	10.09
	SFO_NW	9.97	9.85	10.06	9.73	9.85	10.07
	SJC_NE	9.64	9.53	9.74	9.43	9.52	9.73
	SJC_SE	9.64	9.74	9.95	9.57	9.66	9.87
	SJC_SW	13.19	13.06	13.35	12.87	12.99	13.28
	SJC_NW	13.21	13.08	13.36	12.90	13.02	13.31
ORD-MDW metroplex	ORD_NE	12.44	12.53	12.81	12.21	12.30	12.57
	ORD_SE	12.51	12.61	12.88	12.92	13.00	13.29
	ORD_SW	10.02	10.15	10.37	9.87	9.99	10.22
	ORD_NW	9.85	9.97	10.19	9.71	9.83	10.05
	MDW_NE	12.79	12.90	13.18	12.52	12.62	12.90
	MDW_SE	6.57	6.70	6.85	6.46	6.60	6.74
	MDW_SW	11.99	12.09	12.36	10.92	11.02	11.27
	MDW_NW	16.32	16.41	16.77	16.55	16.64	17.00
IAH-HOU metroplex	IAH_NE	9.66	9.78	10.00	9.76	9.89	10.10
	IAH_SE	10.18	10.30	10.52	10.09	10.21	10.43
	IAH_SW	12.73	12.73	13.10	12.87	12.96	13.24
	IAH_NW	12.31	12.40	12.68	12.10	12.20	12.47
	HOU_NE	13.59	13.70	14.01	13.62	13.70	14.00
	HOU_SE	9.97	10.05	10.27	9.64	9.72	9.93
	HOU_SW	8.88	8.99	9.19	8.56	8.67	8.86
	HOU_NW	13.55	13.69	13.99	13.35	13.49	13.78
DFW-DAL metroplex	DFW_NE	9.81	9.93	10.14	9.70	9.82	10.03
	DFW_SE	12.41	12.50	12.78	12.40	12.49	12.76
	DFW_SW	12.43	12.53	12.80	12.55	12.65	12.93
	DFW_NW	10.07	10.19	10.42	9.95	10.08	10.30
	DAL_NE	9.94	10.06	10.29	9.97	10.07	10.29

	DAL_SE	11.58	11.66	11.91	11.58	11.66	11.92
	DAL_SW	11.92	12.03	12.30	11.90	12.01	12.28
	DAL_NW	11.23	11.37	11.63	11.27	11.40	11.66
TEB-EWR metroplex	TEB_NE	7.32	7.45	7.61	7.20	7.33	7.49
	TEB_SE	12.88	12.97	13.26	12.67	12.77	13.05
	TEB_SW	15.42	15.51	15.85	15.28	15.37	15.71
	TEB_NW	10.82	10.93	11.17	10.52	10.64	10.87
	EWR_NE	9.35	9.49	9.69	9.39	9.52	9.73
	EWR_SE	11.43	11.53	11.79	11.54	11.64	11.90
	EWR_SW	13.40	13.48	13.78	13.56	13.64	13.94
	EWR_NW	10.95	11.05	11.30	11.12	11.22	11.47
	JFK-LGA metroplex	JFK_NE	11.04	11.15	11.39	10.87	10.98
JFK_SE		9.35	9.49	9.70	9.37	9.51	9.72
JFK_SW		11.58	11.68	11.94	11.14	11.25	11.50
JFK_NW		13.69	13.77	14.07	13.97	14.08	14.40
LGA_NE		9.91	10.02	10.24	9.98	10.08	10.31
LGA_SE		11.12	11.26	11.51	11.03	11.17	11.42
LGA_SW		11.97	12.08	12.34	11.97	12.08	12.35
LGA_NW		13.94	13.96	14.27	11.81	11.89	12.15

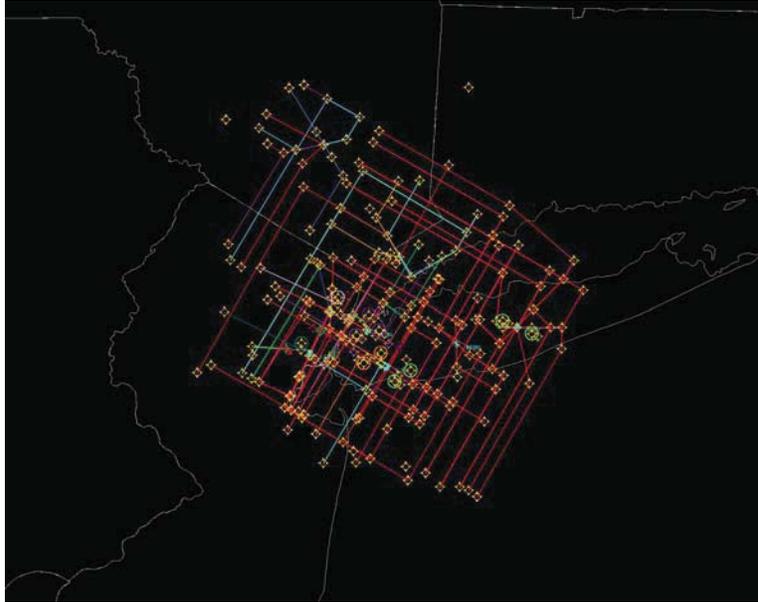
## 5.2 New York Geometries

Arrival and departure paths and vertical profiles were developed for all of the airspace within the radar coverage of N90 TRACON. The flight paths of each airport-fix pair were grouped and a route, which is representative of a nominal flight trajectory, was defined. Routes for jet and turboprop aircraft were segregated and separate routes were built for each group. Special attention was paid to route convergence and divergence points in order to capture airspace interactions. Aircraft speeds by weight class along the trajectories were also noted so that an accurate representation of the 4D trajectory could be modeled. The resulting arrival and departure route structure is shown in Figure 9 as modeled in SIMMOD for current day conditions (arrival in green and departure in red).



**Figure 9: Current N90 Geometry**

A second simulation model was developed to facilitate the study of the potential impacts of NextGen technologies and procedures, and more relevant for this study, to allow us to further illustrate the relative benefits of scheduling versus airspace redesign. The NextGen Geometry was constructed by combining the outer portion of the Current Geometry with an “inner” airspace geometry developed at Georgia Tech that is depicted in Figure 10. This inner airspace was basically connected to the outer airspace at the various entry and exit points of the inner airspace. Speed and altitude profiles were then adjusted to reflect a CDA profile for arrivals and continuous ascent and acceleration for departures. The most important characteristic of the NextGen airspace is the fact that all routes are decoupled from each other. Procedures associated with each arrival or departure fix-runway combination do not interact with each other. This significantly simplifies the airspace operations since operations at one airport do not affect the operations at another airport. One potential concern with the decoupled airspace is its conformance to existing noise constraints restrictions in N90. In the decoupled airspace design, the arrivals and departures assumed near optimal profiles, thus the noise footprint should be smaller. The purpose of the design was to test delay and throughput impact of the concept. Although some of the arrival or departure routes might fly over noise sensitive areas due to the simplified design, design improvements could be incorporated in the future to address the noise concern should an implementation be desired.



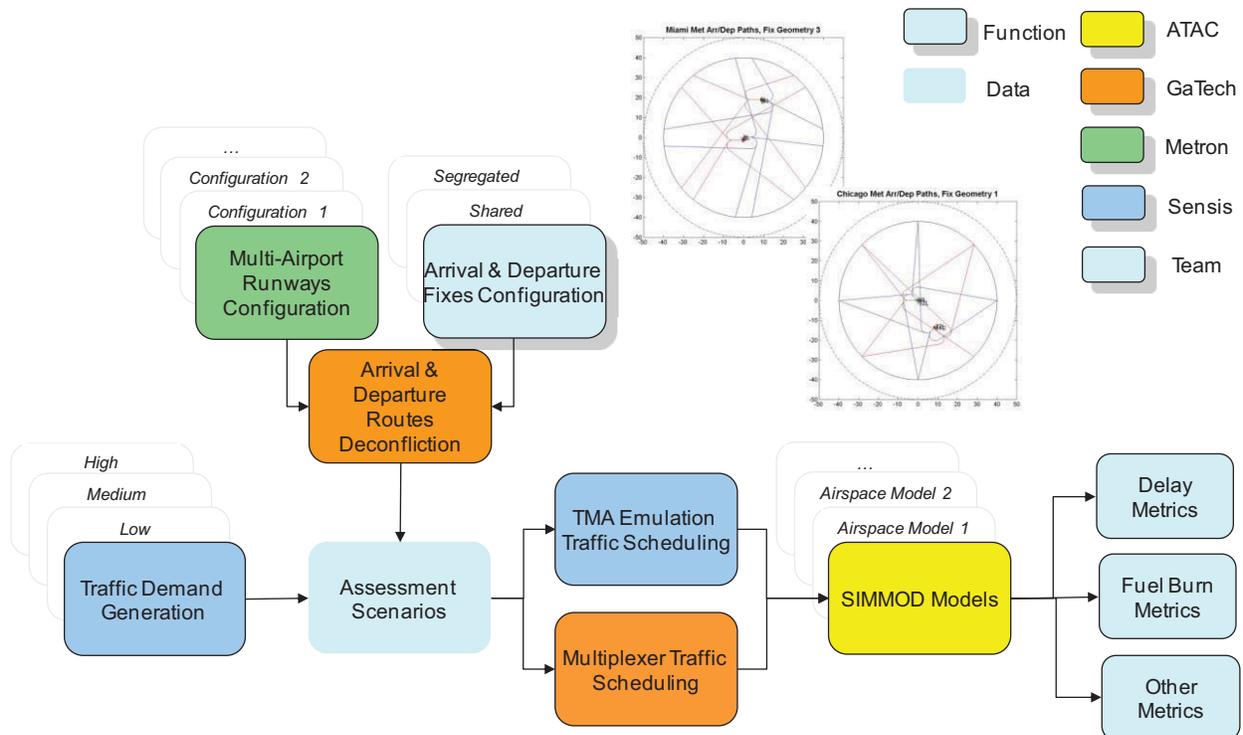
**Figure 10: Inner Portion of NextGen N90 Geometry**

## **6 Analysis Framework and Methodology**

An overview of the framework and the methodology used to evaluate the metroplex scheduling algorithms is provided below.

### **6.1 Analysis Architecture**

The analysis architecture is comprised of traffic demand, airspace model, traffic scheduling, traffic simulation, and simulation data post-processing elements. The architectures and components particular to the generic metroplex and N90 assessments are discussed below.



**Figure 11. Generic Metroplex Evaluation Architecture.**

The figure above depicts the evaluation system architecture to assess the impact of metroplex-wide flight scheduling for the range of geometries evaluated. The system is comprised of the following components: Traffic Demand Generation, Assessment Scenarios, Traffic Scheduling, SIMMOD models, and Delay, Fuel Burn, and Other Metrics. The Traffic Demand Generation component produced schedules of arrival and departure traffic to each generic metroplex airport. For this study, traffic demand scenarios were generated to represent low, medium and high metroplex traffic demand conditions. The Assessment Scenarios component creates the simulation scenarios used to evaluate the metroplex traffic scheduling algorithms. Each scenario was comprised of particular Multi-Airport Runway Configurations and Arrival and Departure Fix Configuration. This study specified 7 different runway configurations among the two generic metroplex airports, each representative of a distinct real-world configuration. This study also specified two different arrival and departure fix configurations: fixes shared between the metroplex airports and separate fixes for each metroplex airport. The combinations of multi-airport runway configurations and arrival and departure fix configurations yielded the 14 different generic assessment scenarios. For each assessment scenario, arrival and departure routes were designed to connect each metroplex airport runway with its associated arrival or departure fix, and the routes were adjusted laterally and vertically to ensure they were spatially deconflicted with one another between the fixes and the runways. For each assessment scenario, the nominal transit times were estimated for each arrival and departure route. These transit times were input to both the Traffic Scheduling and SIMMOD models. Traffic Scheduling computed STAs to the arrival and departure fixes and the runway thresholds for all generic metroplex flights. This study evaluated two scheduling algorithms: a TMA Emulation Traffic Scheduling algorithm and a Multiplexer Traffic scheduling algorithm. The TMA Emulation Traffic Scheduling algorithm modeled the scheduling algorithms and approach of the existing Traffic Management Advisor (TMA) terminal traffic scheduling system and served as the baseline

traffic scheduler. The Multiplexer Traffic Scheduling algorithm implemented a linear programming-based optimization formulation to metroplex-wide traffic scheduling.

Traffic Scheduling was conducted off-line, prior to simulation, for each traffic demand scenario, and each generic metroplex geometry assessment scenario. In turn, the resulting set of scheduled metroplex flights for each demand scenario and assessment scenario was input to the appropriate SIMMOD Model of the generic metroplex airspace. The generic metroplex arrival and departure traffic movement was simulated using the SIMMOD software, using the appropriate airspace model per the assessment scenario, with the arrival STAs to the arrival fix and departure flight STAs to the runway thresholds as the SIMMOD injection times, i.e., the entry time of the flights into the traffic simulation. The SIMMOD queuing-based traffic flow model, simulated traffic flow associated with the specified assessment scenario and produced nominal route transit times, airport capacities, and inter-flight longitudinal spacing requirements to generate flight Actual Times of Arrival (ATAs) to the fixes and runways. The STAs, ATAs and other simulation data were analyzed to compute Delay, Fuel Burn and Other Metrics to assess metroplex scheduling system performance and effectiveness.

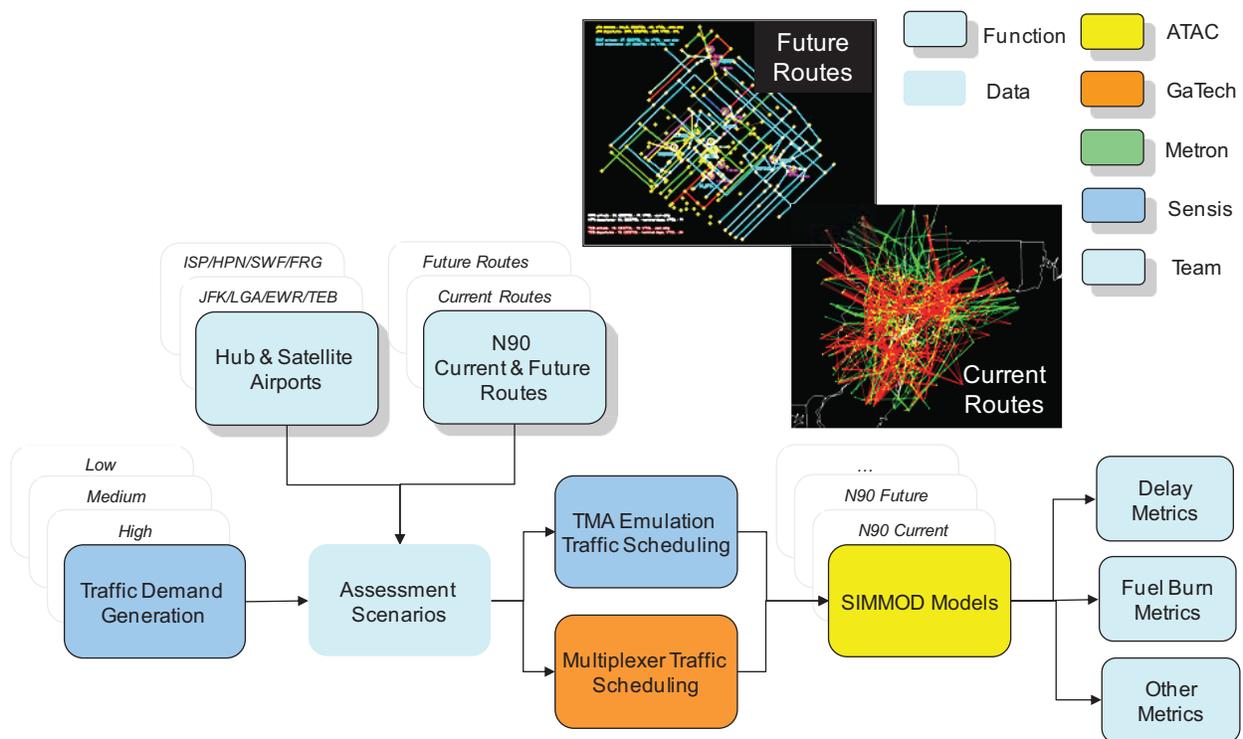


Figure 12. N90 metroplex evaluation architecture.

The evaluation architecture for the N90 metroplex simulations, depicted in the figure above, is very similar to that for the generic metroplexes. In this case, the Traffic Demand Generation specified traffic demand scenarios for the particular N90 metroplex airports to be evaluated. For this study, there were two Assessment Scenarios for the N90 metroplex: Current Routes and Future Routes. Each scenario, in the N90 metroplex model, modeled hub airports John F. Kennedy (JFK), LaGuardia (LGA), Newark (EWR) and Teterboro (TEB), and included key satellite airports Long Island MacArthur (ISP), Westchester County (HPN), Newburg Stewart

(SWF), and Republic (FRG). The Current Routes scenario modeled the existing arrival and departure route structure to those airports, and the Future Routes scenario modeled a hypothetical arrival and departure route structure for those airports, that would be enabled by the enhanced aircraft navigation performances anticipated in the future NextGen national airspace system. Traffic Scheduling was conducted for each N90 airspace model, for each demand scenario. As in the generic metroplex studies, Traffic Scheduling comprised two alternative scheduling algorithms: TMA Emulation and Multiplexer. For each demand scenario, for each airspace, for each traffic scheduling algorithm, SIMMOD Models of the appropriate airspace simulated traffic flow. The resulting simulation data were analyzed in a post-processing step to evaluate the impact of scheduling on metroplex flight delays, fuel burn and other metrics.

## 6.2 Assumptions

The Georgia Tech Metroplex research team utilized certain common assumptions across the baseline and optimized scheduling scenarios to enable a reasonable comparative assessment of the two scheduling capabilities. These assumptions included:

- (i) The required minimum separation criteria that the scheduling algorithms should adhere to (both at the runway threshold and at the TRACON boundary fix).
- (ii) The nominal traversal times from TRACON entry to runway touchdown for different aircraft depending upon their engine types.
- (iii) TRACON boundary fix usage (e.g., use of multiple altitude segregated streams passing over a common arrival-fix).

### 6.2.1 Required Minimum Separation Criteria

We assumed that the minimum required separation at the TRACON boundary fix (arrival fix) would be five nautical miles. Assumptions about the aircraft ground speeds while crossing the TRACON boundary fix are necessary for converting this distance-based separation criterion into a time separation for use in the Time-based scheduling algorithms. Our assumptions for arrival-fix crossing ground speeds are shown in Table 6. As shown in the table, when converting the distance-based separation criterion into a time separation, the minimum arrival-fix crossing speed between leading and trailing aircraft was used for the conversion. The resulting time separations for the different aircraft pairs are shown in Table 7.

**Table 6: Arrival-fix Crossing Speed Assumptions**

		Arrival-fix Crossing Speeds in Knots		
		Leader		
Trailer		H	L	S
	H	290	265	205

	L	265	265	205
	S	205	205	205

**Table 7: Arrival-fix Crossing Time Separation Assumptions**

	Required Time Separation in Seconds			
		Leader		
Trailer		H	L	S
	H	62.1	67.9	87.8
	L	67.9	67.9	87.8
	S	87.8	87.8	87.8

The runway minimum required separation criteria were assumed to be dependent on the weight classes of the leading and trailing aircraft, as shown in Table 8. The runway landing speed assumptions are shown in Table 9. These values were used to convert the distance-separations into time separations. The resulting time separation criteria are shown in Table 10.

**Table 8: Runway Minimum Required Distance Separation Criteria**

	Required Distance Separation in NM			
		Leading		
Trailing		H	L	S
	H	4	3	3
	L	5	3	3
	S	5	3	3

**Table 9: Runway Landing Speed Assumptions**

	Runway Minimum Landing Speeds in Knots			
		Leading Aircraft		
Trailing Aircraft		H	L	S
	H	140.0	136.8	102.7
	L	136.8	136.8	102.7
	S	102.7	102.7	102.7

**Table 10: Runway Landing Time Separation Criteria**

		Required Time Separation in Seconds		
		Leading		
Trailing		H	L	S
	H	102.9	78.9	105.2
	L	131.6	78.9	105.2
	S	175.3	105.2	105.2

**6.2.2 TRACON Traversal Time Assumptions**

As explained in Section 5.1, we assessed the performance of the scheduling algorithms across a range of generic metroplex airspace geometries. The main difference across these geometries was the traversal times from the TRACON boundary to the runway. The Miami-Fort Lauderdale (MIA-FLL) runway layout geometry and its computed traversal times were used as the baseline nominal TRACON traversal times. For other geometries the additional times to fly were added to the traversal times for this geometry depending upon individual airspace definitions.

Table 11 and 12 show the baseline nominal traversal times for the MIA-FLL geometry with shared TRACON boundary fixes and de-coupled TRACON boundary fixes, respectively. As also explained in Section 5.1, the TRACON boundary fixes are equally spaced across the circular TRACON boundary and are named according to their angular difference from the North direction (e.g., arrival fix 45 refers to the North East TRACON boundary fix).

**Table 11: TRACON Traversal Time Assumptions for the MIA-FLL runway layout (shared TRACON boundary fixes geometry) [all times are in minutes]**

Arrival Route	Arrival Fix	Heavy	Large	Small
MIA_NE	45	9.15	9.24	9.45
MIA_SE	135	9.97	10.06	10.28
MIA_SW	225	9.78	9.90	10.12
MIA_NW	315	9.80	9.92	10.14
FLL_NE	45	9.07	9.16	9.36
FLL_SE	135	14.30	14.41	14.73
FLL_SW	225	14.04	14.16	14.47
FLL_NW	315	9.20	9.33	9.54

**Table 12: TRACON Traversal Time Assumptions for the MIA-FLL runway layout (de-coupled TRACON boundary fixes geometry) [all times are in minutes]**

Arrival Route	Arrival Fix	Heavy	Large	Small
MIA_NE	40	10.44	10.53	10.76
MIA_SE	130	9.99	10.11	10.33
MIA_SW	230	9.94	10.05	10.28
MIA_NW	320	9.67	9.80	10.01
FLL_NE	50	8.63	8.72	8.91
FLL_SE	140	11.51	11.61	11.86
FLL_SW	220	10.71	10.83	11.07
FLL_NW	310	11.91	12.04	12.31

### 6.3 Metrics

The primary metric for all the team’s assessments was arrival delay. Delay was categorized by TRACON delay (i.e., delay to be absorbed inside the TRACON) and en route delay (i.e., delay to be absorbed before reaching the TRACON boundary). All delays were computed with respect to the initial arrival-fix (AF) and runway estimated times of arrival (ETAs).

## 7 Experimental Testbed

### 7.1 Simulation Platforms

#### 7.1.1 Simple Queuing Model

We used the same network queuing model we used in our previous metroplex study, the only difference being the greater number of geometries that will be simulated. As you may recall, this simulation environment is a set of queues that are used to model the delays that accrue when aircraft must wait for constrained resources. Because it is a queuing network, it is easily configured and thus very flexible.

#### 7.1.2 SIMMOD and SIMMOD PRO!

The majority of our simulation evaluations were conducted using the Airport and Airspace Simulation Model (SIMMOD), which has been validated by the Federal Aviation Administration (FAA), is an industry standard analysis tool used by airport planners and operators, airlines, airspace designers, and air traffic control authorities for conducting high-fidelity simulations of current and proposed airport and airspace operations.

SIMMOD *PRO!* is an addition to the SIMMOD maintaining all existing code. The addition provides the capability for the user to specify rules and rule processing logic to make decisions based on the state of the airport/airspace system and invoke the SIMMOD engine to perform in a

manner consistent with the rules and producing outputs that provide results based on the use of the rules and the decisions made during the simulation.

A more detailed description of SIMMOD and SIMMOD PRO! can be found in Appendix B of this report.

## **7.2 Demand Scenarios**

This section describes the traffic demand scenarios generated for the simulation-based evaluation of the alternative metroplex-wide traffic scheduling algorithms for each of the geometries. Traffic demand sets were generated to support evaluations for all the generic metroplex and New York metroplex geometries evaluated in this study. The demand sets were input to a queuing-based simulation of air traffic for each metroplex model.

### **7.2.1 Generic Metroplex**

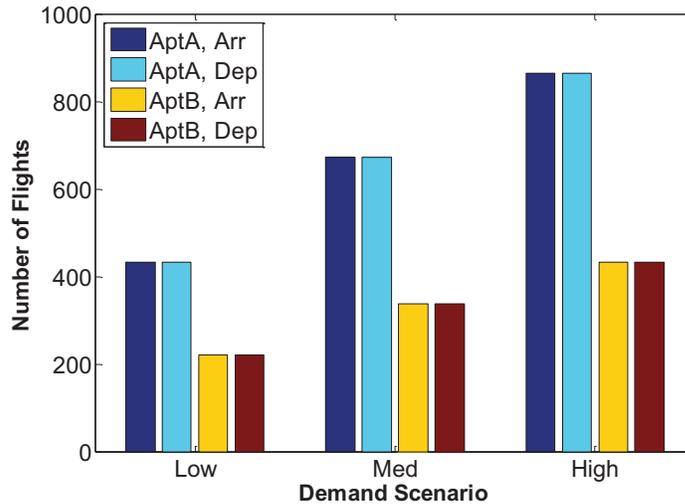
Traffic demand sets were developed to support scheduling algorithm evaluations for the 14 different generic metroplex models. Each generic metroplex model comprised two airports, Airport A and Airport B, and a set of arrival and departure fixes. The models differed in the runway configurations of the two airports, and in the configuration of the arrival and departure fixes. The models spanned 9 different runway configurations modeled from pair-wise interactions observed in current-day metroplexes. The models spanned two different arrival and departure fix configurations observed in current-day metroplexes; shared arrival and departure fixes and segregated arrival and departure fixes. The demand scenarios generated supported evaluating this range of generic metroplex geometries.

The demand sets were generated using the previously developed Metroplex Demand Analysis Tool [TI10]. The tool was developed during the previous NASA-funded metroplex research study led by Georgia Tech. The metroplex demand analysis tool adapts the traffic demand set of a real-world airport to meet the prescribed demand levels of the generic metroplex airport under study, as per the generic metroplex airport's specified capacity and demand/capacity ratio (a measure of airport capacity utilization). This ensures that subject airport scheduled arrival and departure demand, which in turn impacts flight delay accrual characteristics, are representative of real world traffic while complying with the parameters of the particular generic metroplex model under study. It also ensures the other traffic demand characteristics such as mix of aircraft types, equipage levels, and origin and destination airports also represent real-world traffic. In addition to generating a schedule of airport traffic, the tool also adapts the traffic demand to the particular subject airspace under study by assigning each flight to an arrival or departure fix in the terminal airspace, and by estimating the gate, runway and fix crossing times for each flight. For each traffic demand scenario, the metroplex demand analysis tool is used to generate a traffic demand set for each generic metroplex airport independently, and the demand sets are integrated into a single simulation input file.

The demand sets for this study were generated from a set of scheduled flights for Atlanta-Hartsfield International Airport (KATL) for September 26, 2006. The set of scheduled flights was obtained from a previous NASA project and were derived from Federal Aviation Administration (FAA) Enhanced Traffic Management System (ETMS) data.

### 7.2.2 Demand Levels

For this study, traffic demand sets were generated for Low, Medium and High generic metroplex demand scenarios. The number of arrival and departure flights for each generic metroplex airport, for each demand scenario, is depicted in the figure below. In each scenario, each generic metroplex airport has an equal number of arrivals and departures, and generic metroplex airport A has twice the quantity of flights as airport B. The demand scenarios span the range of airport traffic volumes (relative to capacity) identified through analysis of four different real-world metroplexes [ref]: Atlanta, Southern California, New York and Miami. The demand scenarios also capture the nominal relative traffic loading conditions among airport pairs in a metroplex as identified through analysis of the same four different real-world metroplexes.



**Figure 13. Numbers of scheduled arrival and departure flights for generic metroplex airports A and B for low, medium and high traffic demand scenarios.**

In each of the 9 generic metroplex models, for which details are presented in Appendix C, airports A and B had the same arrival and departure capacities. Thus, one traffic demand set was generated for each airport for a low, medium and high demand scenario, and the same the traffic demand set for each airport for each scenario was used for all 9 of the airport configurations assessed in this study.

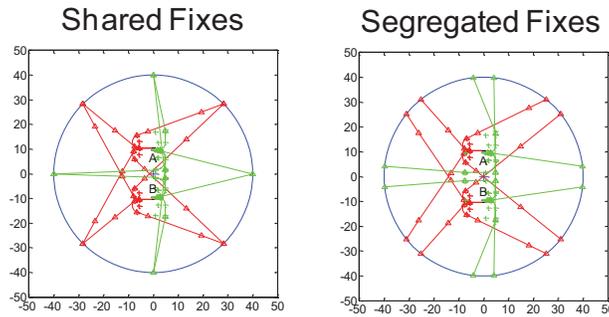
**Table 13 Generic metroplex airports A and B demand/capacity ratios for each demand scenario**

Scenario	Demand/Capacity	
	Airport A	Airport B
High	0.9	0.45
Medium	0.7	0.35
Low	0.45	0.225

In each of the 9 generic metroplex models, airports A and B had the same arrival and departure capacities. Thus, one traffic demand set was generated for each airport for a low, medium and high demand scenario, and the same the traffic demand set for each airport for each scenario was used for all 9 of the airport configurations assessed in this study.

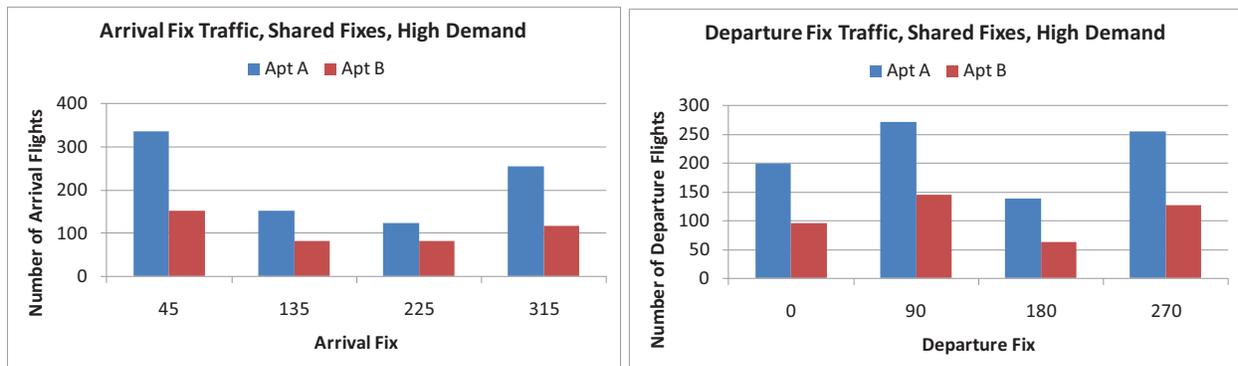
### 7.2.3 Airspace Adaptation

Traffic demand sets were generated for generic metroplex airports A and B for the low, medium and high demand scenarios. As stated above, this set of common traffic demand scenarios applied to all 9 of the metroplex configurations assessed in this study. In turn, each traffic demand set was adapted to each of the two generic metroplex airspace fixes configurations assessed in this study: shared arrival and departure fixes and segregated arrival and departure fixes. The figure below depicts the two fixes configurations assessed in this study.



**Figure 14. Two generic metroplex arrival and departure fixes configurations, shared fixes and segregated fixes, were evaluated at low, medium and high demand scenarios.**

Airspace adaptation of the traffic demand set comprised assigning each flight to an arrival or departure fix on the generic metroplex airspace boundary aligned by the bearing of the arrival flight’s origin airport and departure flight’s destination airport relative to the generic metroplex location. The generic metroplex location was implied by the selected real-world airport from which the traffic demand sets were derived, and the origin or destination airport for each flight was that which was listed in the set of scheduled traffic for the real-world airport from which the traffic demand sets were derived. Figure 15, below depicts the flight counts at the arrival and departure fixes resulting from adapting generic metroplex airports’ A and B high demand traffic scenario to the generic metroplex airspace shared fixes geometry. In turn, flight crossing points to key scheduling points and candidate simulation injection points of interest, such as the fixes and runways, are estimated based on the resulting flight geometry.



**Figure 15. Arrival and departure fixes flight counts by generic metroplex airport for shared fixes configuration for the High demand scenario**

## 7.2.4 Traffic Demand File Output

The output is a comma separated value (.csv) file listing the scheduled departures and arrivals for the generic airport A or B. The quantity of arrivals and departures reflects the specified airport capacity and demand-to-capacity ratio. Table 1 depicts an example of a generated schedule with all available output data from the demand generation process (output format is configurable). The generic source/sink airport of each flight is given as the airport's bearing using an East-North-Up convention. The generic metroplex arrival or departure fix assigned to each arrival and departure flight is given as the fix's bearing. Each flight has a gate departure time, runway takeoff time, fix crossing time, runway landing time, and gate arrival time consistent with the generic airspace geometry. Additional flight information, such as its flight number, aircraft type, engine type, and weight class are also included.

**Table 14. Output Traffic Data Set Example.**

Num	Type	Wgt	Engine	TermSpd_Kts	EnRteSpd_Kts	DepApt	DepFix	ArrFix	ArrApt	GateDepTime	RwyDepTime	DepFixTime	ArrFixTime	RwyArrTime	GateArrTime
4062	E145	L	J	259.1	539.6	322.5		315.0	Apt A	1:18:04	1:28:05	1:37:40	2:51:49	3:01:24	3:07:24
4073	B752	L	J	257.5	497.1	157.5		135.0	Apt A	1:31:36	1:41:00	1:50:46	3:13:04	3:22:50	3:29:50
4081	CRJ2	L	J	236.1	607.2	292.5		315.0	Apt A	1:33:18	1:44:05	1:54:15	3:00:31	3:10:41	3:16:44
4084	CRJ2	L	J	233.1	522.9	352.5		315.0	Apt A	1:34:40	1:46:23	1:56:44	3:17:55	3:28:16	3:34:17
4088	E170	L	J	254.9	603.7	337.5		315.0	Apt A	1:50:03	1:58:31	2:08:21	3:16:52	3:26:42	3:32:47
4102	B735	L	J	246.0	573.4	52.5		45.0	Apt A	1:54:40	2:07:49	2:17:48	3:32:26	3:42:25	3:49:33
4129	B712	L	J	243.7	510.5	337.5		315.0	Apt A	2:16:14	2:24:48	2:34:55	3:53:55	4:04:02	4:10:06
4143	CRJ1	L	J	259.4	452.6	52.5		45.0	Apt A	2:19:49	2:37:47	2:47:10	4:22:05	4:31:28	4:37:31
4146	B752	L	J	257.7	588.0	157.5		135.0	Apt A	2:24:13	2:35:19	2:44:58	3:56:11	4:05:50	4:11:59
4158	B737	L	J	251.1	629.8	67.5		45.0	Apt A	2:37:40	2:47:43	2:57:38	4:02:36	4:12:31	4:18:32
8	AT72	L	T	252.4	544.5	Apt A	270.0		232.5	1:02:58	1:15:35	1:25:27	2:43:52	2:53:44	2:58:44
9	B764	H	J	230.1	483.9	Apt A	180.0		157.5	1:07:03	1:19:19	1:30:04	2:53:43	3:04:28	3:09:32
11	MD88	L	J	241.2	462.8	Apt A	270.0		232.5	1:12:42	1:25:38	1:35:37	3:02:15	3:12:14	3:17:21
27	B712	L	J	254.6	571.9	Apt A	90.0		52.5	1:18:34	1:30:55	1:40:45	2:56:42	3:06:32	3:09:35
29	E145	L	J	255.8	533.0	Apt A	270.0		292.5	1:19:28	1:31:54	1:41:45	2:55:38	3:05:29	3:10:31
38	A319	L	J	264.1	549.0	Apt A	270.0		292.5	1:25:17	1:38:07	1:47:29	3:02:21	3:11:43	3:16:46
60	MD88	L	J	233.5	498.8	Apt A	0.0		337.5	1:40:19	1:52:34	2:03:13	3:25:50	3:36:29	3:41:35
68	CRJ2	L	J	251.7	620.4	Apt A	270.0		262.5	1:41:00	1:53:42	2:03:24	3:11:54	3:21:36	3:26:41
90	CRJ2	L	J	261.3	503.8	Apt A	0.0		337.5	1:46:27	1:58:29	2:07:59	3:32:49	3:42:19	3:46:19
121	B764	H	J	254.5	397.1	Apt A	180.0		157.5	1:53:42	2:06:04	2:15:56	3:55:45	4:05:37	4:10:38

For example, Flight 4062 is an arrival to the generic airport, departing from its source airport located at a bearing of 322.5 degrees at a scheduled gate departure time of 1:18:04 a.m. The flight has been assigned to an arrival fix located at a bearing of 315 degrees on the terminal airspace boundary, and is estimated to cross the fix at 2:51:49 a.m. The flight is estimated to land at the generic airport at 3:01:24 a.m. The flight uses an Embraer E145 jet aircraft falling into the Large weight class. Estimated average terminal area and en-route transit speeds are 259.08 and 539.63 knots, respectively.

As another example, Flight 8 is a departure from the generic airport scheduled to take off at 1:15:35 am, landing to its generic destination airport at bearing angle 232.5 degrees at an estimated time of 2:53:44 a.m. The flight has been assigned to a departure fix located at 270 degrees on the terminal airspace boundary, and is estimated to cross the fix at 1:25:27 a.m. The flight uses an ATR AT72 turboprop aircraft falling into the Large weight class. Estimated average terminal area and en-route transit speeds are 252.42 and 544.48 knots, respectively.

### 7.2.5 Generated Traffic Demand Sets

Thus, each low, medium and high traffic demand scenario for each generic metroplex airport A and B was adapted to each shared and segregated airspace fixes configuration to yield the following demand sets:

Traffic Level	Airspace Configuration	Traffic Demand Set
Low	Shared	3xKATL-0.23D2C_OutputFDS_TC1-AptB_20100727.csv
		3xKATL-0.45D2C_OutputFDS_TC1-AptA_20100727.csv
	Segregated	3xKATL-0.23D2C_OutputFDS_TC3-AptB_20100727.csv
		3xKATL-0.45D2C_OutputFDS_TC3-AptA_20100727.csv
Medium	Shared	3xKATL-0.35D2C_OutputFDS_TC1-AptB_20090925.csv
		3xKATL-0.70D2C_OutputFDS_TC1-AptA_20090925.csv
	Segregated	3xKATL-0.35D2C_OutputFDS_TC3-AptB_20090925.csv
		3xKATL-0.70D2C_OutputFDS_TC3-AptA_20090925.csv
High	Shared	3xKATL-0.45D2C_OutputFDS_TC1-AptB_20100727.csv
		3xKATL-0.90D2C_OutputFDS_TC1-AptA_20100727.csv
	Segregated	3xKATL-0.45D2C_OutputFDS_TC3-AptB_20100727.csv
		3xKATL-0.90D2C_OutputFDS_TC3-AptA_20100727.csv

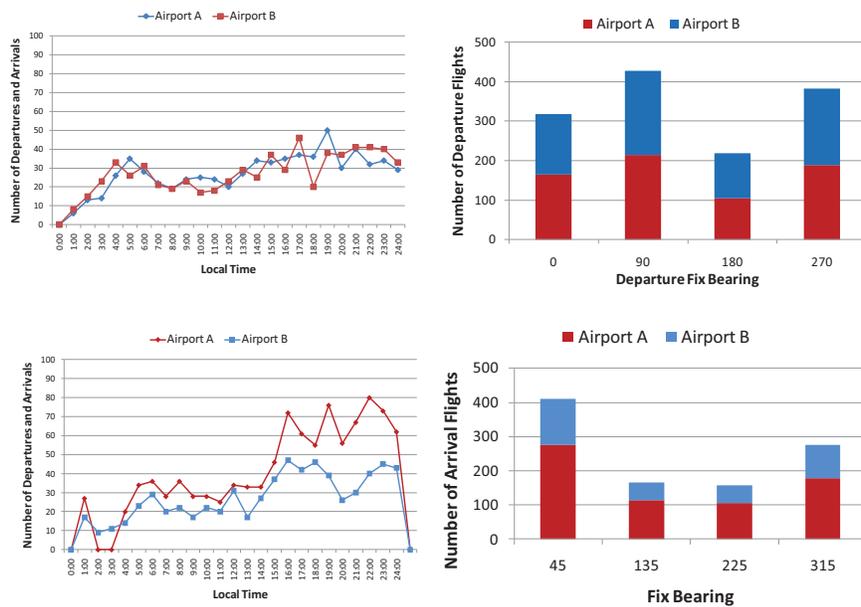
The table indicates three traffic level scenarios, Low, Medium and High. For each traffic level, each airspace configuration, Shared and Segregated, was evaluated. Two traffic demand sets were generated, One for Airport A and one for Airport B, for each distinct combination of traffic level and airspace configuration. This yielded the 12 distinct demand sets listed above.

The traffic demand set naming convention is as follows. First, “3xKATL” designates that the traffic demand set was derived from the traffic schedule for Atlanta Hartsfield International Airport (KATL) corresponding to a hypothetical future National Airspace System traffic demand condition situation in which NAS-wide traffic is three times current-day traffic levels. This future KATL traffic demand schedule was chosen to minimize traffic demand lulls typically arising during the day in order to rigorously exercise the scheduling algorithms under evaluation. Second, “-0.23D2C” specifies the generated demand set corresponds to a particular generic metroplex airport operating under 16 hour demand to capacity ratio of 0.23. This particular example corresponds to an extremely lightly loaded airport. Third, “TC1” and “TC3” identify the shared or segregated arrival and departure fixes conditions, respectively, among the metroplex airports. Fourth, “-AptA” and “-AptB” indicate the generic metroplex airport the traffic demand set is for. Lastly, “20100727” is the date on which the demand set was generated; July 27, 2010 in this case.

### 7.2.6 Additional Demand Variables

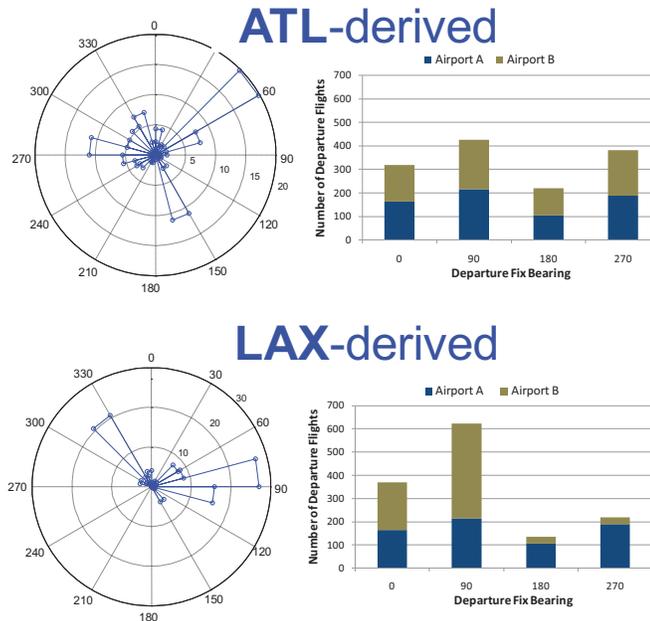
This study focused on metroplex traffic volume as the primary demand variable. However, additional demand variables including the relative traffic volumes of the generic metroplex airports and the directional distributions of traffic and may also affect metroplex-wide flight delay and fuel burn performance.

The relative traffic levels of the two generic metroplex airports determine their degree of contention for shared metroplex resources (e.g., airspace fixes). Increasing contention for shared resources places increasing demand on metroplex-wide scheduling to coordinate the multi-airport traffic flows. This parameter could be varied to assess the impact of relative traffic volume distribution on metroplex-wide scheduling effectiveness.



**Figure 16. Time-based (left) and fix-based (right) traffic distributions for equal (top) and unequal (bottom) metroplex airports A and B traffic levels.**

The directional distributions of the two generic metroplex airports determines the demand for particular shared metroplex resources, and also determines the degree of contention between the airports for use of those resources. As shown, the demand for each airport has the time of day periodicity that is typically seen at airports due to both the nature of passenger demand and airline hub-and-spoke scheduling practices. Similarly, in Figure 17, the geographical distribution of origins and destinations relative to the airports in question (ATL and LAX) is evident. The ATL-derived traffic is primarily from the Northeast, while the LAX-derived traffic is primarily from the East with significant other traffic from the Northwest. This parameter could be varied to assess the impact of relative traffic spatial distribution on metroplex-wide scheduling effectiveness.



**Figure 17. Bearing directional (left) and fix-based (right) traffic distributions for more balanced (top) and less balanced (bottom) metroplex airport traffic distributions.**

Additional demand parameters include the relative temporal traffic distributions of metroplex airports A and B as in phase or out of phase in determining the level of contention for shared metroplex resources.

### 7.3 New York Metroplex

Traffic demand sets were developed to support scheduling algorithm evaluations for the 2 different models of the New York metroplex: one model reflecting the current metroplex route structure, and another model representing a candidate future metroplex route structure enabled by aircraft area navigation and required navigation performance (RNAV/RNP) capabilities. Both of the New York metroplex models comprised primary airports JFK, LGA, EWR, TEB and satellite airports ISP, HPN, SWF, and FRG.

The demand sets were generated using the previously developed AvDemand demand generation tool [HU04] [HU07]. AvDemand was developed through NASA SBIR Phase I (2003) and Phase II (2004) projects and further enhanced via multiple NASA, FAA and JPDO contracts. The tool creates future demand sets (both flight schedules and flight plans) from current day baseline flight demand data. Alternative demand generation approaches include homogeneous or heterogeneous (e.g., FAA Terminal Area Forecast (TAF)) airport growth rates and operations- or passenger-weighted demand growth. The tool balances arrival and departure flights among the origin-destination airports captured in the input demand set. The tool includes a trajectory generator for flight time estimation based on Eurocontrol Base of Aircraft Data (BADA). Lastly, the tool features multiple methods to shape the demand characteristics and distributions

The demand sets for this study were generated from a set of scheduled flights for the aforementioned airports from September 26, 2006. The set of scheduled flights was obtained from a previous NASA project and were derived from FAA ETMS data.

### 7.3.1 Demand Levels

For this study, New York metroplex traffic demand sets were created for three demand scenarios: current-day traffic, future traffic 1.2 times current day, and future traffic 1.6 times current day. The quantities of arrival and departure flights for each metroplex airport are depicted in the tables below for each demand scenario. The demand scenarios represent the range of future airport traffic volumes estimated for different future NextGen timeframes.

**Table 15. New York metroplex airport traffic demand levels for current day scenario**

<b>Airport</b>	<b>Departures</b>	<b>Arrivals</b>	<b>Total</b>	<b>Growth</b>
KJFK	554.0	535.0	1089.0	N/A
KLGA	610.0	607.0	1217.0	N/A
KEWR	631.0	621.0	1252.0	N/A
KTEB	305.0	280.0	585.0	N/A
KFRG	32.0	33.0	65.0	N/A
KSWF	26.0	23.0	49.0	N/A
KISP	62.0	59.0	121.0	N/A
KHPN	193.0	184.0	377.0	N/A
Total			4755.0	N/A

**Table 16. New York metroplex airport traffic demand levels for future 1.2 times current day traffic scenario**

<b>Airport</b>	<b>Departures</b>	<b>Arrivals</b>	<b>Total</b>	<b>Growth</b>
KJFK	667.0	648.0	1315.0	1.2
KLGA	748.0	748.0	1496.0	1.2
KEWR	761.0	748.0	1509.0	1.2
KTEB	344.0	314.0	658.0	1.1
KFRG	34.0	33.0	67.0	1.0
KSWF	28.0	25.0	53.0	1.0
KISP	73.0	69.0	142.0	1.2
KHPN	218.0	210.0	428.0	1.1
Total			5668.0	1.2

**Table 17: New York metroplex airport traffic demand levels for future 1.6 times current day traffic scenario**

<b>Airport</b>	<b>Departures</b>	<b>Arrivals</b>	<b>Total</b>	<b>Growth</b>
KJFK	884.0	845.0	1729.0	1.6

KLGA	941.0	938.0	1879.0	1.5
------	-------	-------	--------	-----

The individual demand scenarios achieve the target metroplex-wide traffic growth amounts, with the individual airport traffic volumes at or near the target scaling. The individual airport traffic volumes vary due to the balancing of arrival and departure traffic among the origin and destination airports in the demand set, with the smaller airports exhibiting greater sensitivity to the different traffic levels arising due to this demand balancing.

Throughout the demand generation process, the temporal traffic demand profile for each airport is retained, as this is an important demand characteristic which in part determines delay accrual profiles.

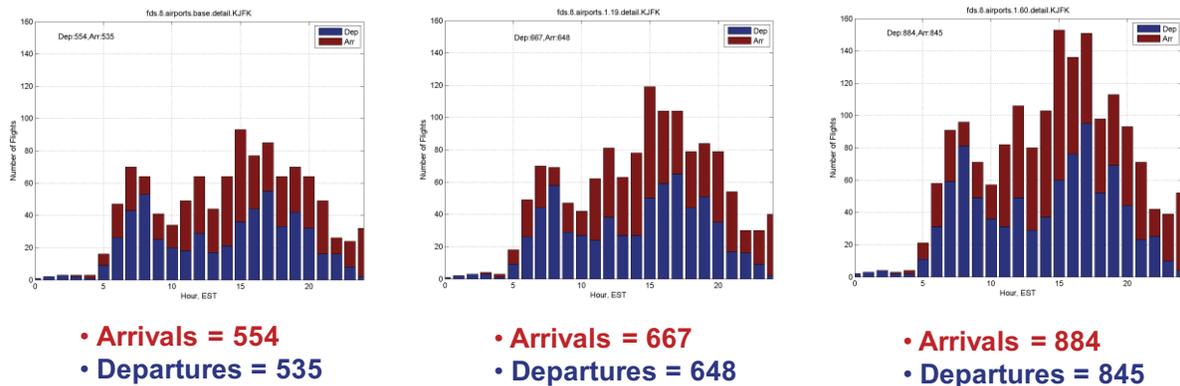


Figure 18. JFK arrival and departure demand profiles for current day (left), future 1.2x (center) and future 1.6x (right) traffic levels.

### 7.3.2 Airspace Adaptation

Following generation of the traffic demand scenarios, the traffic demand sets for each airport were updated to assign each arrival flight to an arrival fix or each departure flight to an airport runway at the physical boundaries of the New York metroplex airspace model, and to estimate arrival fix crossing times or runway entry times for the arrival and departure flights. These were estimated from nominal fix and runway assignments and transit times characterized from historical traffic data.

## 8 Results

In this section, we summarize the key results of the use of our experimental test bed to assess the potential benefits of the Multiplexer. We first discuss the Generic Airspace results and then we discuss the New York Airspace results. In general, three parameters are used to describe the performance of the Multiplexer and TMA on a per aircraft basis: the average total arrival delay; the average en route arrival delay; and the average terminal area arrival delay.

The first parameter, the total arrival delay per aircraft, is arguably the most important measure of performance as it indicates both the amount of arrival delay that each flight would have to absorb relative to its desired schedule, and how well the scheduling algorithm utilizes runway resources. The former relationship is straightforward; however the latter requires further explanation. In any system with a constrained resource, arrival delays increase as the efficiency of the utilization of that resource decreases. That is, any pause in the use of the resource that could otherwise be used

productively will, by definition, result in an arrival delay for some (in some cases all) of the subsequent traffic. Thus, when a scheduling algorithm delays the time that an aircraft gets to its runway to avoid a potential conflict at an upstream merge point or intersection, when it could have otherwise avoided that arrival delay, the scheduling algorithm induces additional arrival delay on some (and in very heavy traffic scenarios most if not all) subsequent traffic. Airlines have to absorb such arrival delays operationally (with the associated crew and fuel costs as well as passenger compensation costs) or by way of schedule changes (the actual and opportunity costs that are incurred when the flight schedule must be adjusted to maintain passenger connections and schedule integrity). Thus, even a one minute per aircraft average total arrival delay per aircraft can result in significant cost to the airline; more so because the maximum total arrival delay will be higher than the average total arrival delay. Further, average total arrival delays on the order of tens of minutes could require wholesale schedule changes.

The second and third parameters, the average en route arrival delay and the average terminal area arrival delay, provide an indication of the relative costs of the arrival delays that are incurred. To first order, it is advantageous for arrival delays to be incurred in the en route phase of flight because the fuel used per unit time in the en route phase is less than the fuel used per unit time in the terminal area. Thus, the conventional wisdom is to push as much arrival delay as possible to the en route phase. However, minimizing terminal area arrival delay should not be the “be all and end all” of a scheduling algorithm! In some cases it is better to incur a greater fraction of the arrival delay in the terminal area (than would nominally be the case with TMA) in order to increase runway utilization and thereby decrease the average total arrival delay. This point is perhaps best explained via the aforementioned example of a scheduling algorithm that delayed the time that an aircraft gets to its runway to avoid a potential conflict at an upstream merge point or intersection. In such a scenario, it could be advantageous to advance the time that a set of aircraft (including the aircraft in question) pass a merge point or intersection to ensure that they get to their runway at a time that is closer to their desired landing time, thereby reducing their own arrival delay and the arrival delay of subsequent aircraft.

## **8.1 Generic Airspace Results**

Rather than present the results for all the generic airports, which are very similar in terms of the level of delay reduction, we will only present the results for the two generic metroplexes that are based on the Miami TRACON. Results for all other generic geometries are presented in Appendix C.

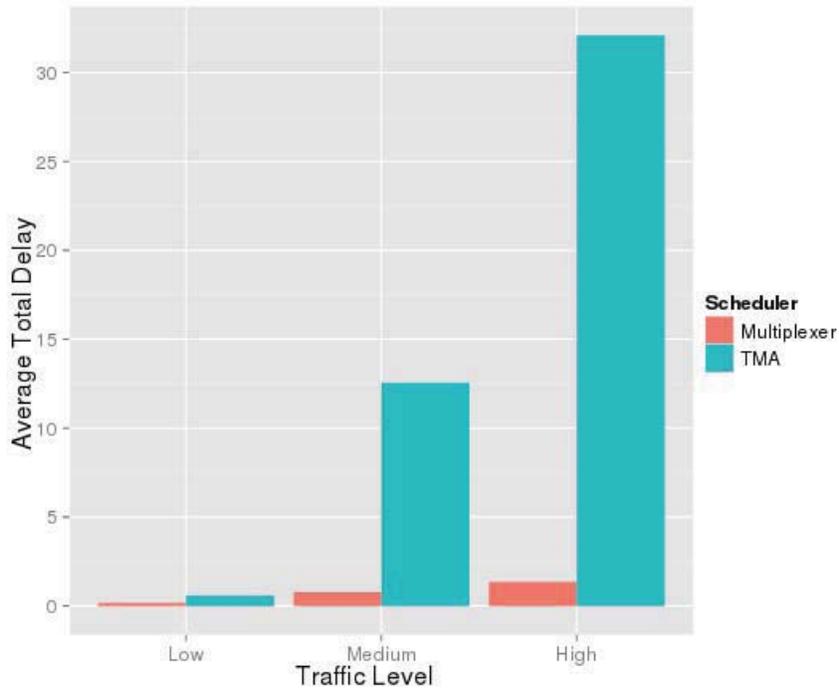
As you may recall, a key feature of all dual-corner-post geometries (the generic geometries where there are dual arrival fixes at each corner post) in our single arrival runway per airport cases is that the merge points at each arrival fix is specific to a runway. This is in contrast to the single-corner-post geometries (the generic geometries where there is a single arrival fix at each corner post) where the merge point at each arrival fix is shared by two airports’ runways and therefore functionally an intersection. Thus, it is reasonable to expect that the average total delay for a any dual-corner-post geometry will be less than the average total delay for the corresponding single-corner-post geometry, as fewer aircraft will be using a given arrival fix therefore the potential for conflicts will be less and thus the need to advance the crossing time of aircraft will be reduced. Further, it is also reasonable to expect that the Multiplexer will be better

able to determine the true minimum average total delay as it explicitly considers the effect of conflict resolution at these merges on runway utilization, while TMA does not.

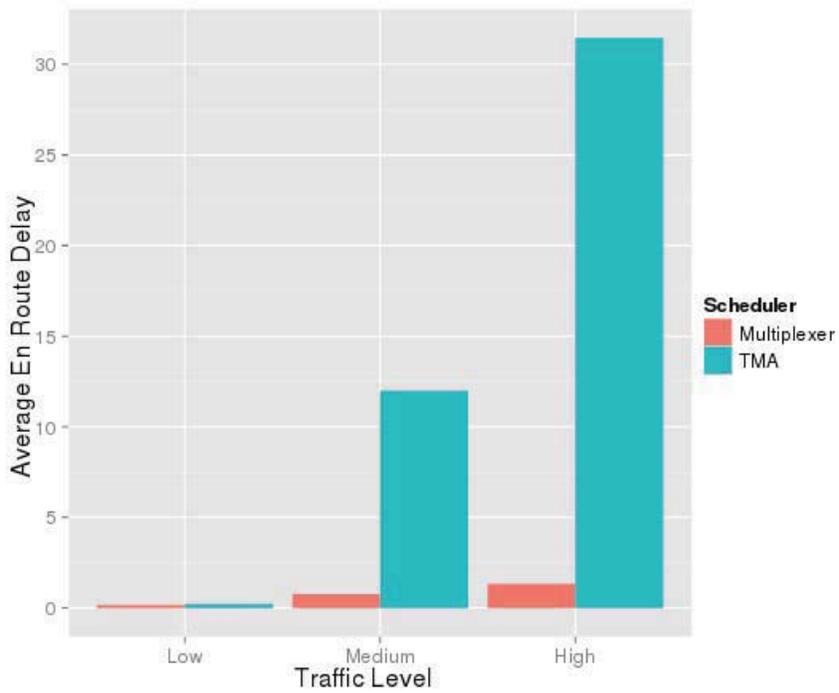
Both of the underlying hypotheses associated with the two aforementioned expectations are validated by the data in Figure 19 through Figure 24. As may be seen from head-to-head comparisons of the average total arrival delay (for either the Multiplexer or TMA) at all three traffic levels simulated, the average total arrival delay for the dual-corner-post geometry is always lower than the average total arrival delay for the single-corner-post geometry. At the low traffic level, the average total arrival delay for the dual-corner-post geometry is approximately 80% of the average total arrival delay for the single-corner-post geometry when TMA is being used, and approximately 84% when the Multiplexer is being used. At the medium traffic level, the average total arrival delay for the dual-corner-post geometry is approximately 57% of the average total arrival delay for the single-corner-post geometry when TMA is being used, and approximately 76% when the Multiplexer is being used. At the highest traffic level, the average total arrival delay for the dual-corner-post geometry is approximately 59% of the average total arrival delay for the single-corner-post geometry when TMA is being used, and approximately 82% when the Multiplexer is being used.

As may be seen from head-to-head comparisons of the average total arrival delay for a given geometry and traffic level, the Multiplexer performs much better than TMA. At the lowest traffic level, the average total arrival delay with the Multiplexer is approximately 31% of the average total arrival delay with TMA for the single-corner-post geometry, and approximately 33% for the dual-corner-post geometry. At the medium traffic level, the average total arrival delay with the Multiplexer is approximately 7% of the average total arrival delay with TMA for the single-corner-post geometry, and approximately 9% for the dual-corner-post geometry. At the highest traffic level, the average total arrival delay with the Multiplexer is approximately 4% of the average total arrival delay with TMA for the single-corner-post geometry, and approximately 6% for the dual-corner-post geometry.

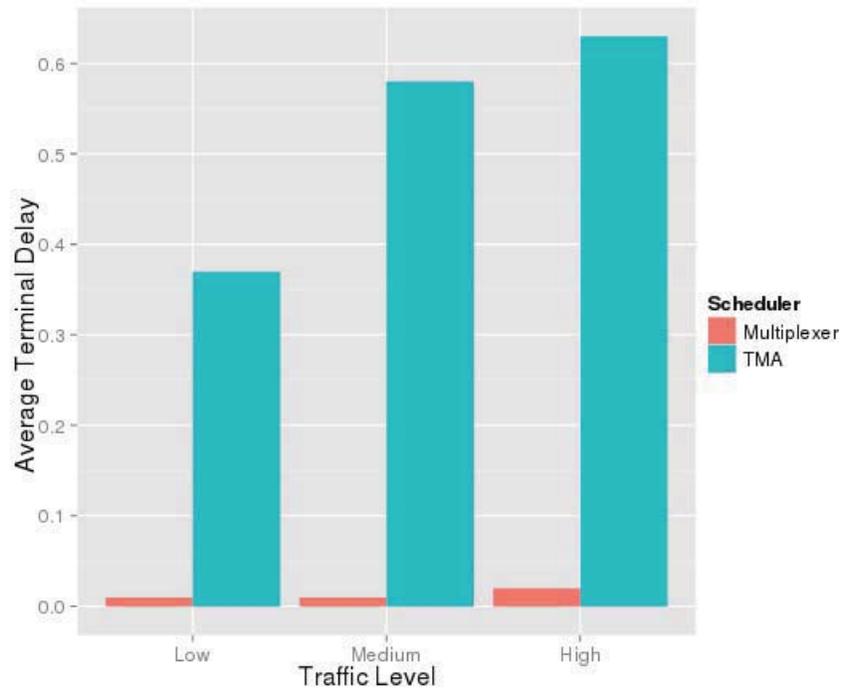
Overall, it may be seen that the arrival delay rises rapidly with traffic level when TMA is being used but remains relatively low when the Multiplexer is being used. This indicates that, by breaking away from the precedence order that is followed by TMA and explicitly considering runway operation in the determination of the fix schedule, the Multiplexer is able to generate a better runway schedule. The net result is that all the aircraft are able to land sooner (as evidenced by the lower total arrival delay).



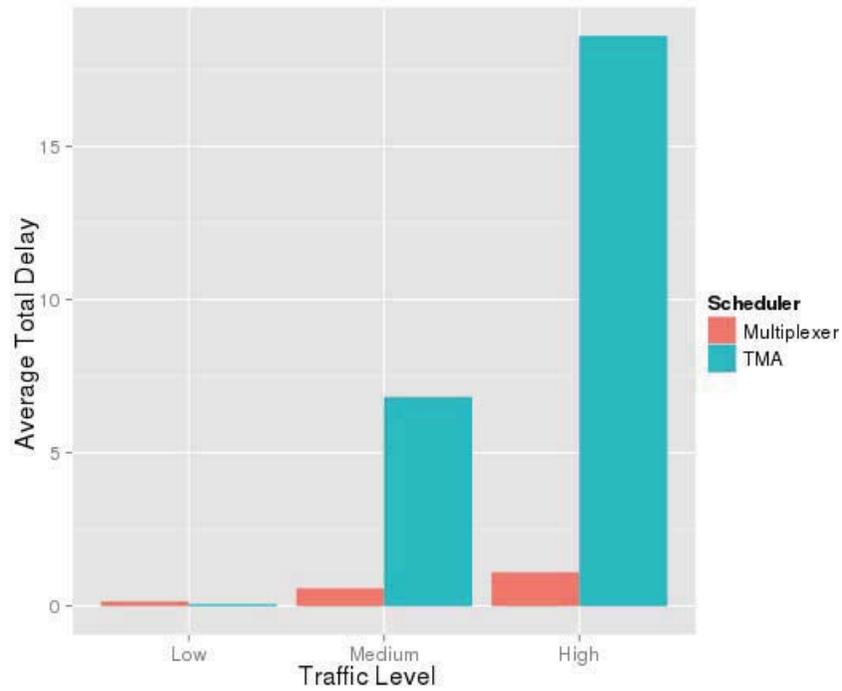
**Figure 19: Average Total Arrival Delays (per aircraft) at Low, Medium, and High Traffic for the Miami-based Metroplex with a Single Arrival Fix at Each Corner-Post**



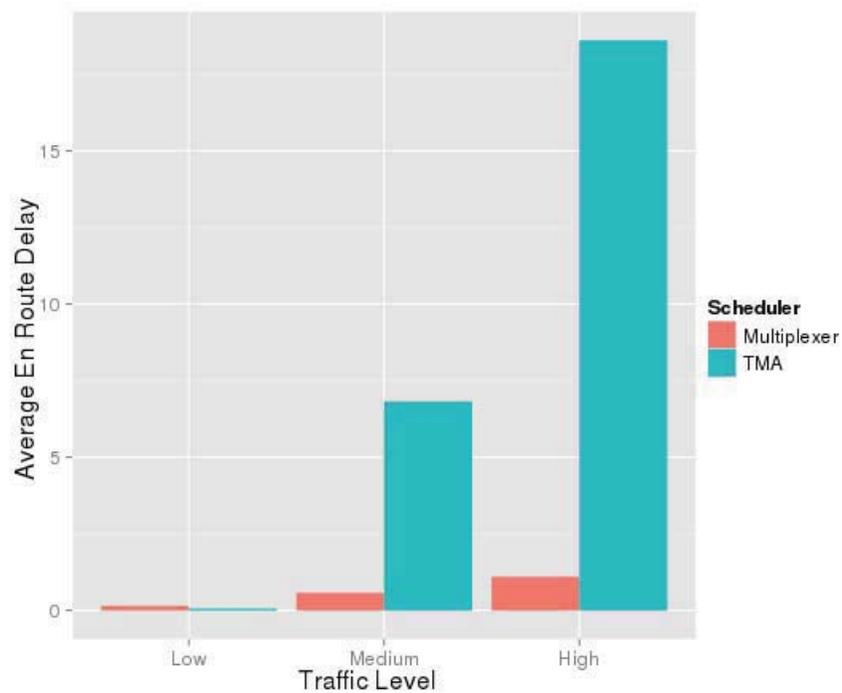
**Figure 20: Average En Route Arrival Delays (per aircraft) at Low, Medium, and High Traffic for the Miami-based Metroplex with a Single Arrival Fix at Each Corner-Post**



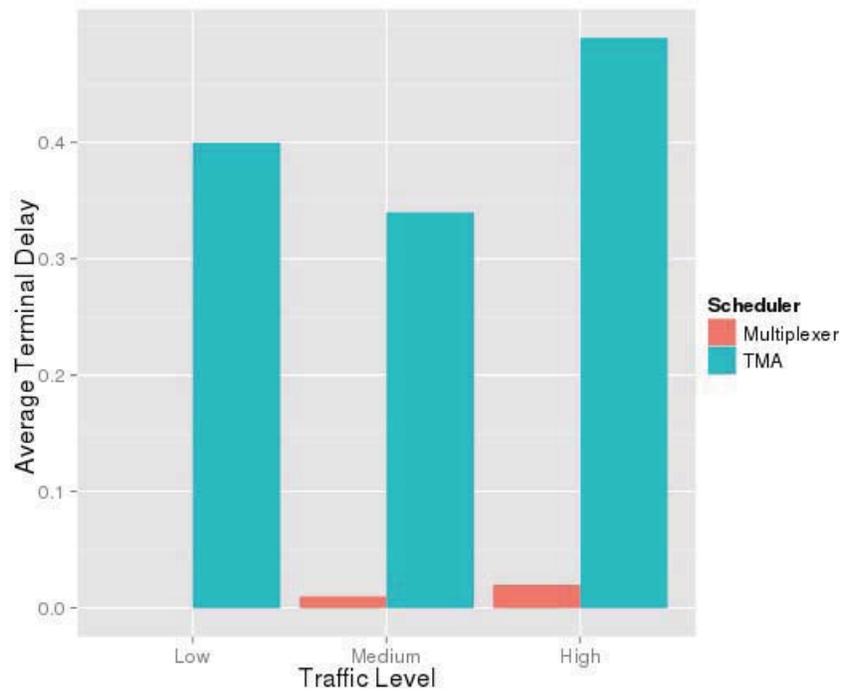
**Figure 21: Average Terminal Area Arrival Delays (per aircraft) at Low, Medium, and High Traffic for the Miami-based Metroplex with a Single Arrival Fix at Each Corner-Post**



**Figure 22: Average Total Arrival Delays (per aircraft) at Low, Medium, and High Traffic for the Miami-based Metroplex with Two Arrival Fixes at Each Corner-Post**



**Figure 23: Average En Route Arrival Delays (per aircraft) at Low, Medium, and High Traffic for the Miami-based Metroplex with Two Arrival Fixes at Each Corner-Post**



**Figure 24: Average Terminal Area Arrival Delays (per aircraft) at Low, Medium, and High Traffic for the Miami-based Metroplex with Two Arrival Fixes at Each Corner-Post**

One source of the overall reduction in total delay due to the Multiplexer algorithm is the allowance for a slight speedup in some flights. The effect of the slight speedup can be utilized to reduce the overall delay of the system in the case where simply slowing down or holding an

aircraft would require subsequent aircraft to accumulate delay. This can reduce the exponential delay build up that is caused by the domino effect of each aircraft's delay requiring the next to also be delayed. A simple example found in the MIA 1 Low demand case can be seen in Table 18 and Table 19 with the values for the ETAs and STAs respectively (all times are in minutes). These two tables demonstrate the how the Multiplexer slightly speeds up two aircraft while slightly slowing down two others.

**Table 18: Example ETAs**

Flight	Class	Fix	Rwy	ETA Fix	Transit	Delay	ETA Rwy	Sep
1135	L	40	A	824.58	10.52	0	835.10	3.22
1139	L	40	A	826.77	10.52	0	837.28	2.18
1133	L	40	A	827.08	10.52	0	837.60	0.32
1144	L	230	A	830.30	10.05	0	840.35	2.75
1141	L	40	A	830.57	10.52	0	841.08	0.73
1131	L	40	A	830.83	10.52	0	841.35	0.27

**Table 19: Example STAs**

Flight	Class	Fix	Rwy	STA Fix	Transit	Delay	STA Rwy	Sep
1135	L	40	A	824.58	10.52	0.00	835.10	3.22
1139	L	40	A	826.62	10.52	-0.15	837.14	2.04
1133	L	40	A	827.94	10.52	0.85	838.45	1.32
1144	L	230	A	829.72	10.05	-0.58	839.77	1.32
1141	L	40	A	830.57	10.52	0.00	841.08	1.32
1131	L	40	A	831.88	10.52	1.05	842.40	1.32

In this example, if only slowing down or holding were allowed to achieve the required 1.315 minutes of separation at the runway, the results can be seen in Table 20. A simple comparison between the sums of total delay is 1.172 for the Multiplexer and 3.213 minutes for the push back

only case. If we consider the sum of the absolute value in delay, the Multiplexer still shows improvement: 2.63 vs. the same 3.213 minutes of total delay for these six aircraft.

**Table 20: Example With Push Back Delay Only**

Flight	Class	Fix	Rwy	ETA Fix	Transit	Delay	STA Rwy	Sep
1135	L	40	A	824.58	10.52	0.00	835.10	3.22
1139	L	40	A	826.77	10.52	0.00	837.28	2.18
1133	L	40	A	827.08	10.52	1.00	838.60	1.32
1144	L	230	A	830.30	10.05	0.00	840.35	1.75
1141	L	40	A	830.57	10.52	0.58	841.66	1.32
1131	L	40	A	830.83	10.52	1.63	842.98	1.32

## 8.2 New York Airspace Results

SIMMOD *PRO!* was used to determine the arrival delay for the New York geometries. The injection times at the terminal area entry were computed off-line using our TMA emulation and the Multiplexer. Thus, only terminal area arrival delay times were determined via simulation. Each en route arrival delay time was computed by taking the difference between the corresponding fix injection time and the corresponding unscheduled or desired fix-crossing time.

The average total, en route and terminal area arrival delays for the Current Airspace and the NextGen Airspace are listed in Table 20, for the current traffic level. As a head-to-head comparison of the average total arrival delay for the Current Airspace with the Multiplexer and the average total arrival delay for the Current Airspace with TMA indicate, the Multiplexer would provide significant reductions in the average total arrival delay. For the Current Airspace, the delay with the Multiplexer would be approximately 61% of the delay with TMA, thus providing a significant reduction of 39% delay over the baseline TMA case. On average for a Current Airspace baseline, this reduction yields a net reduction of 1.4 minutes per flight. This may not sound like much, but it represents only the average reduction and not the individual flight differences, which could be significantly higher than that. In fact, with an average total delay in the TMA case of 3.62 minutes per flight, there is only that much (3.62 minutes per flight) in terms of delays that can be eliminated. Our estimation of the absolute value of minutes of delays should be tempered with the simplistic analytical assumptions that ignore any downstream delay, missed connections, diversions, etc.

The 1.4 minutes per flight cost reduction may not sound like much, but if this were available to US airline flights across the board, at an average of \$60.99/minute direct operating cost and an additional \$37.56/hour in average value of passenger time (based on 2009 statistics, see

<http://www.airlines.org/Economics/DataAnalysis/Pages/CostofDelays.aspx>), the airlines that fly 10.0 million US domestic and international flights (see: [http://www.bts.gov/press\\_releases/2010/bts015\\_10/html/bts015\\_10.html](http://www.bts.gov/press_releases/2010/bts015_10/html/bts015_10.html)) would see a total of over \$862 million dollars in annual savings.

As would be expected, the Multiplexer provides additional delay reductions in the case of shifting from a Current Airspace baseline to a NextGen Airspace future, but these delay reductions are decreased due to the complementary airspace improvements. Under a current day traffic level and movement to an implemented NextGen airspace case, the delay reductions from the Multiplexer reduce from 39% to 10%.

**Table 21. New York Metroplex Arrival Delays for Current and NextGen Airspace at Current Traffic Level (minutes)**

<b>Airspace Structure</b>	<b>Scheduler</b>	<b>Average Total Delay [min]</b>	<b>Average En Route Delay [min]</b>	<b>Average Terminal Area Delay [min]</b>
Current 1.0	TMA	3.62	1.39	2.23
	Multiplexer	2.22	0.51	1.71
NextGen 1.0	TMA	1.97	0.55	1.42
	Multiplexer	1.77	0.54	1.23

That said, the Multiplexer does provides significant (albeit mostly statistical) benefits, and most importantly these benefits are maintained as traffic levels increase. This is exemplified by the change in average total arrival delay for the NextGen Airspace as shown in Table 22. At the current traffic level, the average total arrival delay with the Multiplexer is approximately 90% of the total average total arrival delay with TMA. At 1.2 times the current traffic level, the average total arrival delay with the Multiplexer rises slightly too approximately 93% of the total average total arrival delay with TMA. At 1.6 times the average total arrival delay with the Multiplexer falls slightly too approximately 88% of the total average total arrival delay with TMA.

**Table 22. New York Metroplex Arrival Delays for NextGen Airspace at Three Different Traffic Levels (minutes)**

<b>Airspace Structure</b>	<b>Scheduler</b>	<b>Average Total Delay [min]</b>	<b>Average En Route Delay [min]</b>	<b>Average Terminal Area Delay [min]</b>
NextGen 1.0	TMA	1.97	0.55	1.42
	Multiplexer	1.77	0.54	1.23
NextGen 1.2	TMA	21.41	12.25	9.16
	Multiplexer	19.90	10.37	9.52

NextGen 1.6	TMA	93.05	80.59	12.46
	Multiplexer	81.88	66.78	15.10

Perhaps the most interesting observation about the data in Table 22 is the fact that (at 1.2 and 1.6 times the current traffic level) the average terminal area arrival delay is higher with the Multiplexer than with TMA. This is clear evidence of the point that was made earlier in the section that in some cases it is better to incur a greater fraction of the arrival delay in the terminal area (than would nominally be the case with TMA) in order to increase runway utilization and thereby decrease the average total arrival delay. This is an indication of how important it is to consider the impact of crossings and merge points on downstream resources. Even though the Multiplexer does not explicitly consider all the crossing and merge points in the New York Metroplex, the fact that it does consider the common entry fixes and the runways jointly means that the resulting schedule has a better chance of success even after uncertainties and intersections come into play. It would certainly be interesting to explore the development of a version of the Multiplexer that considers all the merge points and intersections in the New York airspace.

The results in the tables are also interesting from another perspective. TMA is designed to limit the delay in the terminal area for any given aircraft to 5 minutes. Thus, it seems odd that the average terminal area delay would be greater than 5 minutes. However, TMA in its current instantiation would not consider all the crossing and merge points within the metroplex, thus it would not be able to account for the impact that these might have on the traffic flows (i.e., a controller might have to slow or vector an aircraft to allow another aircraft to cross or merge before it).

## 9 Conclusions

As indicated by the results presented in the previous section, metroplex-tailored scheduling in the form of the Multiplexer will provide benefits across the entire range of metroplexes in the NAS (i.e., from simplistic abstractions to complex metroplexes such as the New York Metroplex).

That being said, the important things to note in this regard are:

1. There are significant benefits to explicitly considering all the constraints on timing that are made manifest at crossing and merge points, as each crossing or merge point becomes an opportunity to magnify uncertainties (i.e., a small variation in fix crossing time can result in a significant change in future crossing times and trajectory);
2. By properly advancing traffic through upstream crossings (the entry fixes serving multiple runways) so that there is no starvation of downstream resources (the runways), and by explicitly considering runway operation in the determination of the fix schedule, it is possible to generate a better runway schedule and thereby enable all aircraft to land with a minimum level of delay.
3. The Multiplexer provides significant benefits in terms of delay reduction even though TMA has, because it is a scheduler, accrued some of the total benefits that are due to scheduling;

4. Even after the benefits of TMA are subtracted, the benefits of scheduling and the benefits of airspace redesign are of similar magnitude. Because of the expected significantly easier introduction of scheduling changes versus airspace design changes, we believe that adjusting flight crossing times through scheduling remains the more cost-effective of the two control strategies.

Given these findings, we developed a concept of operations for the Multiplexer in the hope that it would provide impetus for its adoption in one form or the other, perhaps by way of enhancements to TMA.

## **10 Implementation Issues and Concept of Operations**

### **10.1 Today's Metroplex Inefficiencies**

At major US metropolitan area airports with nearby airports, there are inevitably air traffic flow interdependencies for flows into and out of the proximate airports. These interdependencies along with factors such as poor situational awareness and traffic predictability lead to significant congestion for the major metropolitan area airports as well as inefficiencies at the proximate airports and the surrounding airspace. This metroplex congestion and inefficiencies are exacerbated by other major factors such as traffic volume, convective weather, reduced-visibility conditions, conservative air traffic spacing, unbalanced air traffic flows, and mixing of different aircraft types and performance levels.

These metroplex inefficiencies are commonly seen at major US metroplexes of varying levels of air traffic and airspace complexity. As identified in the RTCA Taskforce 5 report [RT09], key US metroplexes include those of least complexity (such as Atlanta, Charlotte, Dallas-Ft. Worth, Houston, Las Vegas, Minneapolis, and Phoenix), greater complexity (e.g., Boston, Denver, Detroit, Memphis, Philadelphia, San Francisco, and Washington, DC), and the most complex (Chicago, New York/New Jersey, and Southern California). Also, as predicted in NASA research efforts such as McClain, et al., [MC09] the number and complexity of US metroplexes are forecasted to grow with the expected growth of air traffic.

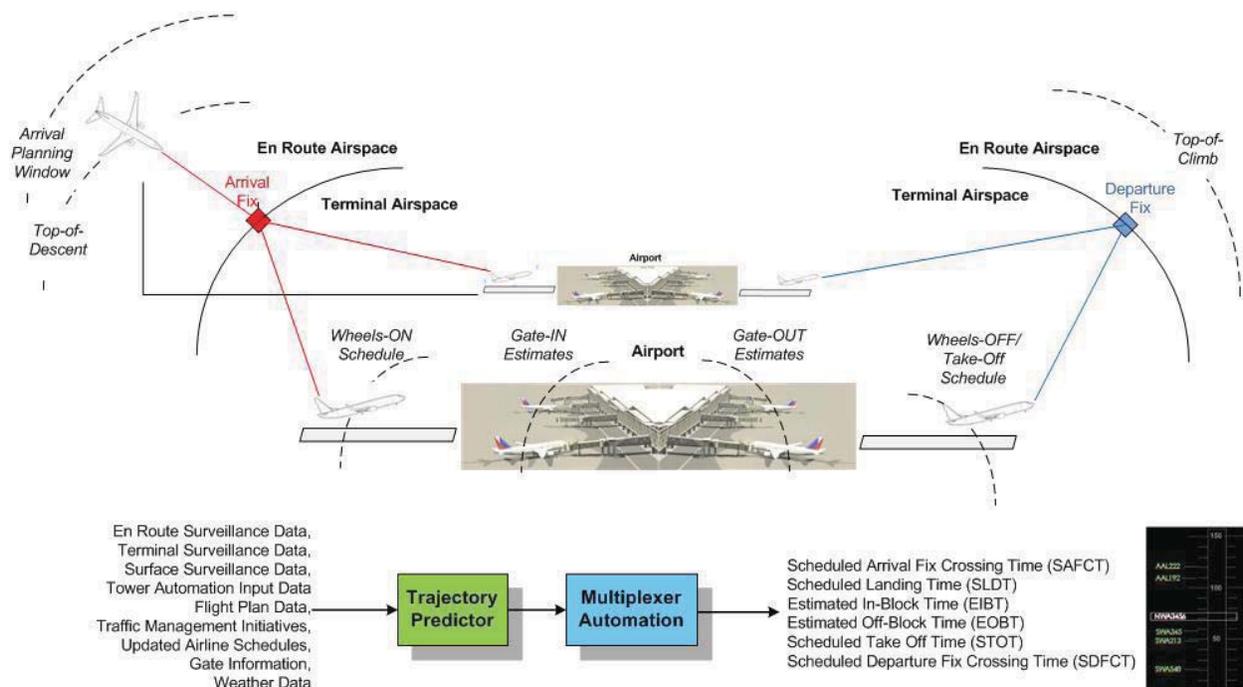
The FAA is attacking the metroplex problem with a near-term focus on optimizing Area Navigation (RNAV) operations and a mid-term focus on integrating procedures that deconflict airports, establishing and maximizing use of 3 nm terminal separation rules, and leveraging more advanced Performance-based Navigation (PBN) solutions where needed. However, more can be done. The RTCA TF5 report recommended the additional development of “ATC, flow, and surface management tools” but was not specific in what technical solutions would be appropriate. One potential mid-term technical solution would be the development and implementation of the Georgia Tech Metroplex NRA team’s “Multiplexer” concept.

### **10.2 A Potential Metroplex Solution: the “Multiplexer”**

In a metroplex environment, multiple proximate airports vie for the concurrent usage of shared resources like common points in the airspace (e.g., arrival fixes, departure fixes, other merge points), common routes in the airspace (e.g., Standard Arrival Routes (STARs), Departure

Procedures (DPs)), or common volumes of airspace (e.g., arrival corridors). Air Navigation Service Provider (ANSP) responses to such cross-airport interactions encompass the entire spectrum from pure temporal separation where the ANSP works to regulate the times at which aircraft enter the TRACON airspace or times at which aircraft cross certain points in the airspace, to pure spatial separation where the ANSP provides guidance to traffic flows to multiple interacting airports by separating them vertically or laterally.

The Georgia Tech Metroplex NRA team’s “Multiplexer” concept is focused on the pure temporal separation of traffic. A decision support tool that will enable the ANSP to temporally separate interacting traffic flows to and from multiple metroplex airports is needed to enable more efficient traffic flows in the NextGen metroplex environment. With such a tool available to the ANSP, flights from individual airports will be able to fly their arrival/departure routes with the ANSP providing temporal controls to enable more efficient use of the available metroplex airspace and current metroplex airspace design while maintaining safe separations.



**Figure 25: The Multiplexer Automation provides scheduled air traffic times at key nodes throughout the metroplex to support more efficient metroplex air traffic planning.**

The Multiplexer concept builds off of the original ideas of the concept from the previous Georgia Tech Metroplex NRA team research defined in [RC10]. The Multiplexer concept applies a Time Division Multiple Access (TDMA) approach to solving the metroplex aircraft network problem. Wikipedia defines TDMA as: “a channel access method for shared medium (usually radio) networks. It allows several users to share the same frequency channel by dividing the signal into different time slots. The users transmit in rapid succession, one after the other, each using their own time slot. This allows multiple stations to share the same transmission medium (e.g., radio frequency channel) while using only a part of its channel capacity.” [W11]

The Georgia Tech Metroplex NRA team proposes a similar temporal separation tool for allocation of shared airspace resources like meter fixes, STARs, DPs, and corridors of airspace. Each user (a metroplex airport in this case) would be allowed to share the resource by allocating a time-slot to it. Each resource will have a dynamically computed schedule of usage and this schedule shall be computed by optimizing traffic coming from all metroplex airports. For example, in the case of New York metroplex, the busiest departure fix – ELIOT – is commonly shared between LaGuardia airport (LGA), Newark Liberty International Airport (EWR), and Teterboro Airport (TEB) departures. The proposed Multiplex temporal scheduler computes an optimized departure fix crossing schedule and optimized wheels-OFF schedules for all flights expected to depart from all three airports within a given look-ahead time window.

### **10.3 Expected Benefits**

Effective use of the Multiplexer tool will result in a wide set of overall benefits to a range of metroplex stakeholders including:

- Reducing flight delays and actual block times
- Improving flight and flow predictability
- Reducing schedule block times
- Increasing metroplex airport capacity and throughput
- Reducing fuel burn, emissions, and noise, and
- Reducing controller workload

The Multiplexer tool's advisories, enhanced situational awareness, and predictions will assist:

- Ramp controllers with determining improved departure and arrival sequences and target gate-out and gate-in times.
- The Ground controllers at individual metroplex airports in determining more efficient and TMI-compliant departure sequence by providing target take-off time constraints
- The Local controllers at individual metroplex airports by delivering aircraft with efficient separation (i.e., separation as close as possible to the minimum required spacing) on the final approach
- The TRACON arrival and departure controllers by providing target meter-fix crossing times for arriving and departing flights
- The TRACON arrival and departure controllers in a more efficient handling of merging and crossing traffic within the metroplex by providing target intermediate-fix crossing times, target landing/take-off times, etc., and by de-conflicting traffic at the merge/cross points
- ARTCC sector controllers with more accurate arrival fix crossing times and reduced workload due to a smoother departure flow integration into ARTCC airspace

- Traffic management coordinators in identifying metroplex flow “hot spots” and supporting strategic decision making to match dynamic metroplex demands to metroplex airport and airspace capacities
- Aircraft Operator flight dispatchers with more accurate predictions on when aircraft will be crossing key NAS thresholds enabling more accurate gate arrival and push-back times. By using Aircraft Communications Addressing and Reporting System (ACARS), the flight dispatchers can relay this information to pilots and, subsequently, passengers as well.

Air Navigation Service Provider potential benefits can be summarized as follows. The Multiplexer tool can reduce controller workload by providing the controllers with deconflicted target fix-crossing times and target landing or take-off times. Also, improved flow management will enable more efficient utilization of metroplex resources (e.g., boundary fixes, runways, terminal routes) during peak traffic periods, resulting in increased throughput.

Aircraft Operator potential benefits can be summarized as follows. Improved flow management in the metroplex can reduce delays for aircraft that arrive into major hub airports during heavy traffic periods as well as reduce the standard deviation of aircraft transit times thereby increasing the predictability of aircraft operations and improving fuel efficiency. This can improve on-time departures and arrivals and schedule reliability that can enable the aircraft operators to reduce scheduled block times, thereby decreasing operating costs and increasing revenues as well.

## **10.4 Functional Overview**

### **10.4.1 Roles**

In the case of the ANSP and Ramp Control personnel, the Multiplexer tool expects certain inputs and provides useful outputs. Metroplex surface and terminal operational procedure adaptation data input and updating is required for effective Multiplexer prediction and operation. Also, dynamic input of planned airport configuration changes is expected by the appropriate ATCT Supervisors in the metroplex. The Multiplexer tool is expected to be used in a number of different ways by the different ANSP personnel. The Multiplexer automation is used by the TRACON controllers to meter traffic crossing the boundary fixes to balance the arrival/departure demand across multiple boundary fixes, multiple TRACON sectors, and multiple metroplex airport runways. The tool is also used by the TRACON controllers to handle merging and crossing traffic by utilizing the tool-provided target fix-crossing times. The tool is also used by airport Ground Controllers as guidance for building the sequence of departures so that the departure traffic load is balanced across all available airport runways, TRACON departure sectors and departure fixes. The tool will also simplify the job of airport Local Controllers by delivering a sufficiently spaced and order-optimized sequence of aircraft on final approach. ARTCC Controllers also can use the Multiplexer to ensure en route arrival traffic flows take into account expected metroplex air traffic dependencies. Finally, Ramp Controllers can also use the tool for guidance in building more efficient pushback sequences. In general, the Multiplexer is expected to not only provide automatic advisories, but also mechanisms to incorporate dynamic controller-desired flight constraints (e.g., enabling personnel to input additional slots, assign aircraft specific slots, enforce desired aircraft sequences, and prioritize emergency flights).

In the case of the Flight Operator personnel, the Multiplexer tool also expects certain inputs and provides useful outputs. The Multiplexer tool expects that the Flight Operators will be sharing flight specific updated departure and arrival gate information for the purpose of providing aircraft intent as early as possible. The concept also enables Flight Operator personnel to input flight specific preferred runway use and have them incorporated into the Multiplexer scheduling. Flight Operator Dispatchers and ATC Coordinators can use their interface into the Multiplexer automation to obtain improved predictability on actual take-off time and gate arrival time. In addition, the Dispatchers can relay these predicted times to Pilots with ACARS messages or radio communication.

#### **10.4.2 Responsibilities**

There is no change in the legal responsibilities of the operational personnel under the Multiplexer concept; the Multiplexer tool acts as an advisory decision support tool only. The ultimate responsibility of separating traffic remains with the controllers and is expected to reduce their workload and increase overall efficiency.

#### **10.4.3 System Function**

The Multiplexer tool acts as a decision support tool for ARTCC, TRACON, ATCT, and Ramp controllers. The Multiplexer provides a Metroplex ARTCC-TRACON-Airport time-based metering system function with a surface traffic prediction module and integrated arrival/departure Multiplexer scheduling. The nominal operation of the Multiplexer is now described.

The position, ground speed, and intent of each aircraft are obtained by an ERAM, STARS/ARTS, or ASDE-X data feed, depending on where the aircraft is located. Traffic Management Initiative intent is provided by the TFMS feed. Estimates of winds aloft are provided by the National Weather Service Rapid Update Cycle (RUC) weather model to determine aircraft speeds (Mach, CAS, and TAS). Site-adapted Airport, TRACON, and ARTCC routes are used to predict the aircraft position/trajectory from each gate to runway threshold to top-of-climb and vice versa starting with top-of-descent to the arrival gate. Metroplex Trajectory Synthesis (TS) algorithms combine the previously mentioned synthesized position and aircraft intent data with: aircraft type, gate, flight plan information, airport configuration, and any AOC user preferences to generate company-preferred trajectories that are used to build an estimated time of arrival (ETA) schedule for the expected runways and fixes.

The real-time Multiplexer scheduler then builds an optimal recommended aircraft sequence and scheduled time of arrival (STA) at fixes and runways that comply with all restrictions in effect including the incorporation of airport configuration information, aircraft separation rules, and TRACON acceptance rates. The Multiplexer scheduler is refreshed every twelve seconds, corresponding to the rate of a secondary radar data feed from the ERAM system, and uses a look ahead time window of 1 hour. Freeze horizons, where the traffic sequences are frozen, are built into the metering processes prior to the critical metering points. The freeze horizons are based on the uncertainties inherent in the aircraft trajectories, and, as such, vary with the different phase of flight.

The Multiplexer tool computes the scheduled time-access times for each shared metroplex resource; arrival/departure fix, merge/crossing-point, runway, or shared airspace corridor, by using a mathematical optimization based scheduling algorithm. The optimized deconflicted crossing /landing/take-off times are routed back to the ARTCC, TRACON, ATCT, and Ramp controllers and accessible by Traffic Flow Management specialists and Flight Operator personnel. The Multiplexer tool also displays predicted and actual crossing times. The controllers use these times as guidance for metering the traffic within their sphere of control.

#### 10.4.4 Operational Scenarios

The use of the Multiplexer tool to support operations can be described in a few operational scenarios. In the figures below, we develop one set of Multiplexer departure procedures (Figure 26) and one set of arrival procedures (Figure 27) which are focused on two flights; Mesaba Airlines Flight 123 (MES123) flying from Washington National Airport (DCA) to John F Kennedy International Airport (JFK), and Atlantic Southeast Airlines Flight 345 (ASQ345) flying from Washington Dulles International Airport (IAD) to LaGuardia International Airport (LGA). For the sake of brevity and clarity, not all the procedural steps are shown.

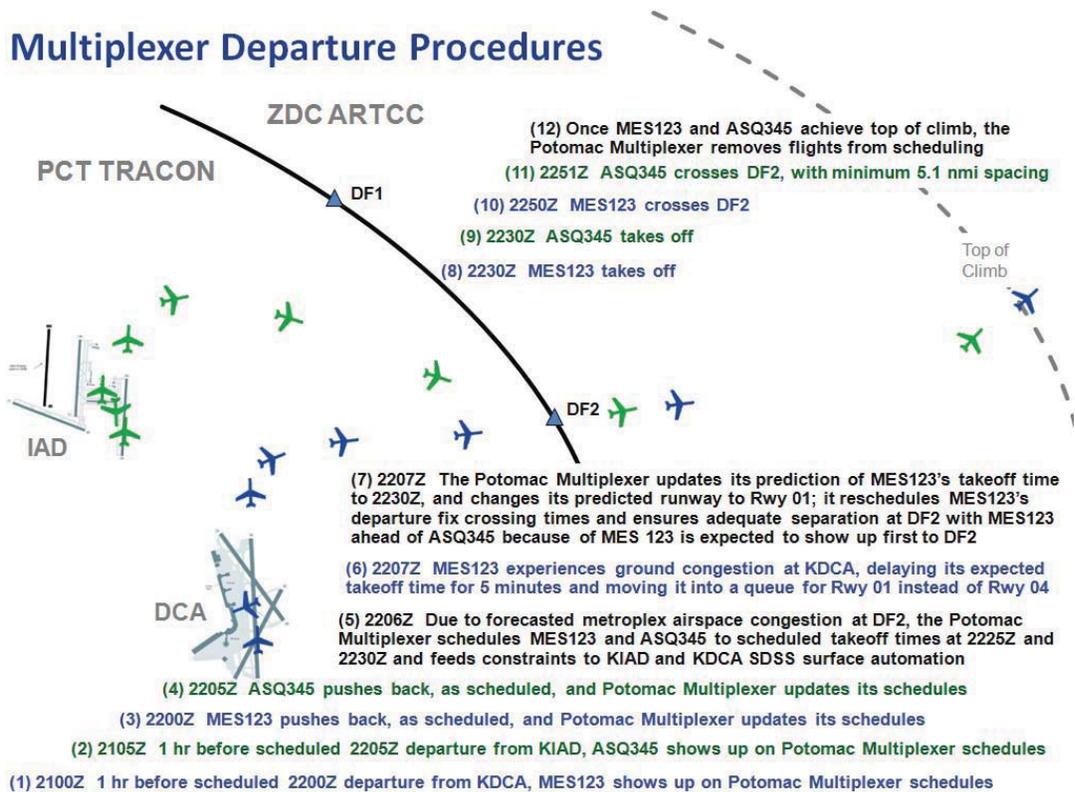


Figure 26: The Multiplexer Automation supports enhanced departure procedures for more efficient operations through the metroplex (MES123-focused actions in blue; ASQ345-focused actions in green; ANSP-focused actions in black).

## Multiplexer Arrival Procedures

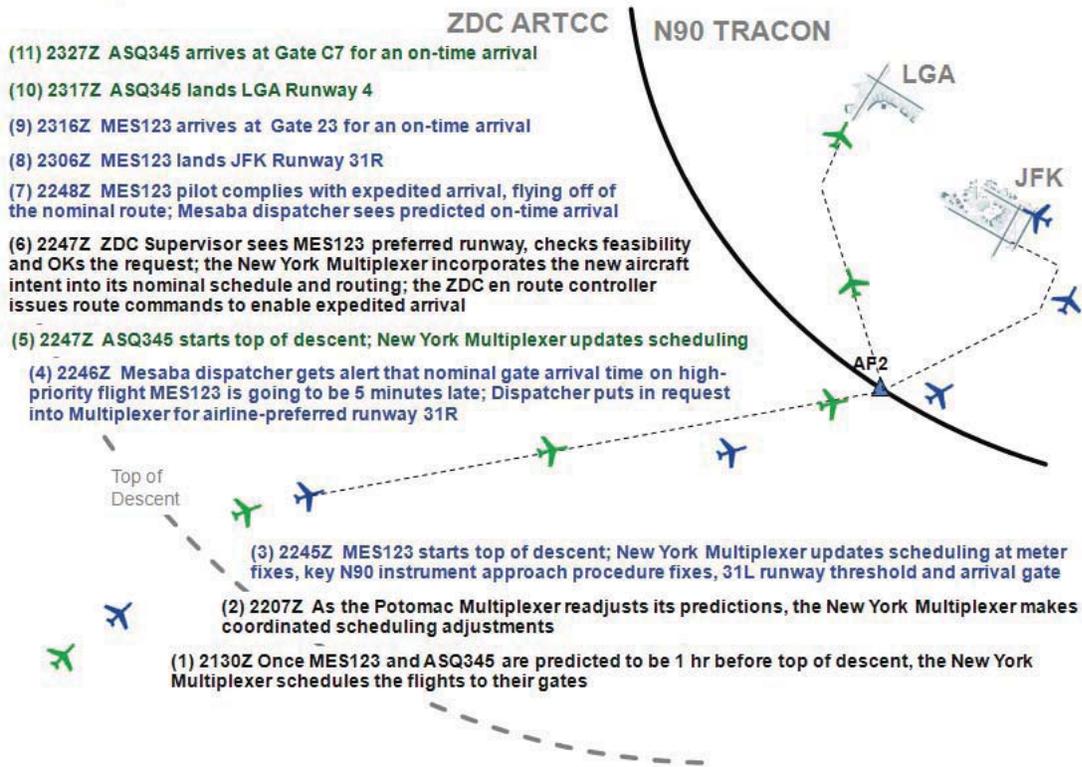


Figure 27: The Multiplexer Automation supports enhanced arrival procedures for more efficient operations and incorporation of user preferences through the metroplex (MES123-focused actions in blue; ASQ345-focused actions in green; ANSP-focused actions in black).

## 10.5 Flight Applicability

### 10.5.1 User Classes

The Multiplexer concept provides services for all user class operations including Air Carrier, Air Taxi, General Aviation, and Military flights.

### 10.5.2 Flight Rules

To benefit from the scheduling capabilities of the Multiplexer tool, IFR flight plans will be required for all aircraft requesting access to the metroplex resources. All VFR overflight traffic is expected to self-separate and is not incorporated into Multiplexer scheduling. VFR and other (e.g., military) arriving/departing traffic who are not on IFR flight plans need to be incorporated into Multiplexer scheduling at the runways through air traffic controller inputs.

### 10.5.3 NAS Domains

The Multiplexer tool is currently designed for arrival and departure aircraft in a metroplex environment (i.e., the extended terminal and airport surface domains). Therefore, flights in current ARTCC and TRACON airspaces or on airport ramps or other movement areas will be affected.

## **10.6 New System Architecture Requirements**

### **10.6.1 Aircraft Equipage**

The Multiplexer concept requires no explicit aircraft equipage.

### **10.6.2 Communication, Navigation, and Surveillance (CNS)**

The accuracy of the aircraft position information input to the Multiplexer tool will have a direct effect on the effectiveness of the calculated schedules, so it is easily understood that the more accurate the surveillance data is, the better the NAS performance will be when using the Multiplexer schedules. Therefore, the increased usage of high-accuracy surveillance sources like GPS-driven ADS-B surveillance is preferred (but not required).

### **10.6.3 Facilities**

The Multiplexer concept will impact all of the key air traffic facilities in a given metroplex including: ARTCCs, TRACONs, ATCTs, Ramp Towers, and Flight Operator Operational Control Centers.

### **10.6.4 Weather**

No special requirements, but understood that accurate trajectory prediction is dependent on accurate current and future wind information. To ensure all-weather use of the Multiplexer, integration of metroplex weather impacts on aircraft routing is desired.

### **10.6.5 Software**

The Multiplexer automation is expected to be implementable in a software architecture that can mimic the current NASA CTAS software architecture with some modifications (see Figure 28). On the input software processing, we envision a new process, here called “XDR” that would process the airport ASDE-X data that would be required for the surface movement predictions. These additional data would require routing through the CTAS Input Source Manager (ISM) and the Communication Manager (CM) and used by enhanced Route Analyzer (RA) and Trajectory Synthesizer (TS) algorithms. In addition, the Multiplexer scheduling algorithms would be integrated into a new Dynamic Planner (DP) process, here known as the “Metroplex Dynamic Planner” (MDP) that would provide the metroplex-wide arrival and departure scheduling algorithms.

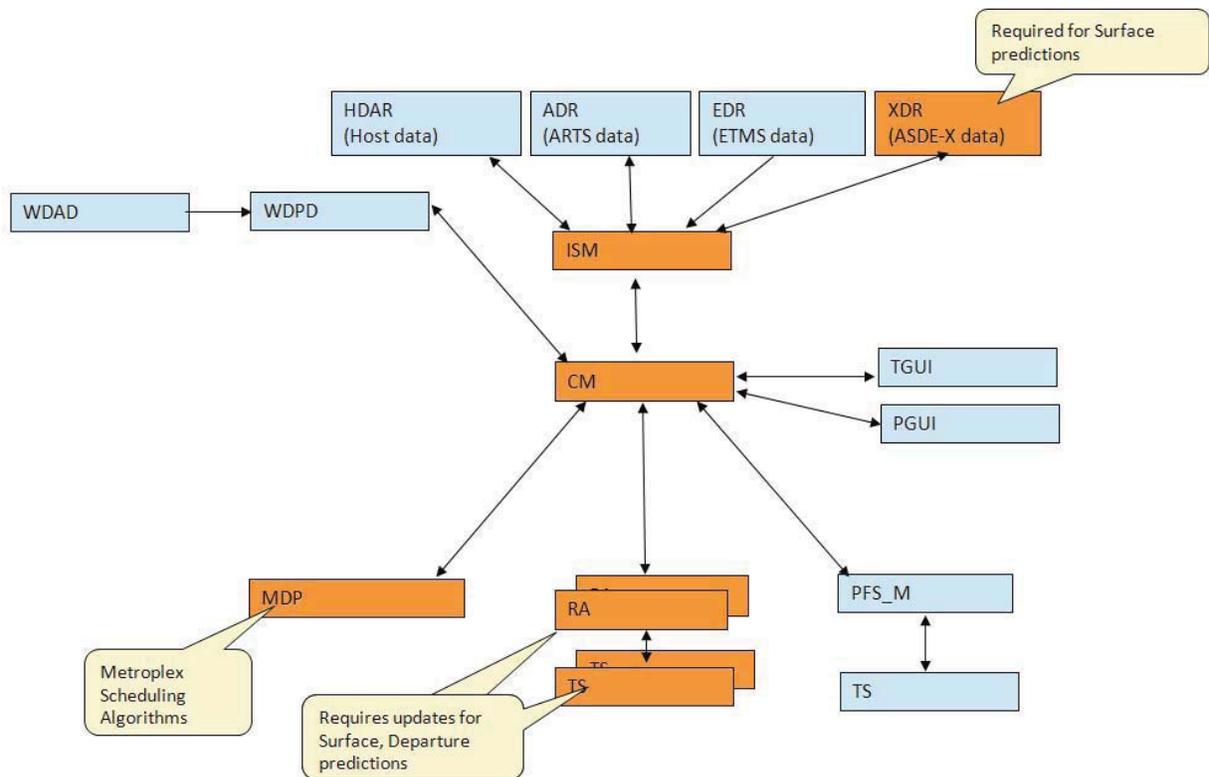


Figure 28: The expected nominal Multiplexer software architecture would require some enhancements to the typical NASA CTAS software processes (in orange).

### 10.6.6 NAS System

The Multiplexer automation is expected to interface with a number of existing and planned NAS systems (see Figure 29). On the input side, the Multiplexer would interface with a host of traffic and weather surveillance systems and datalink systems including ASDE-X (Airport Surveillance Detection Equipment, Model X), ITWS (Integrated Terminal Weather System), CIWS (Corridor Integrated Weather System), RUC (Rapid Update Cycle), ACARS (Aircraft Communications Addressing and Reporting System), and ARTCC, TRACON, and Ramp surveillance systems. The En Route Automation Modernization (ERAM) and Traffic Flow Management System (TFMS) automation would be integrated as well along with, when available, the Tower Flight Data Management (TFDM) ATCT automation systems. Ideally the interface would be through a System Wide Information Management (SWIM) Server. The Multiplexer Automation itself would support multiple Graphical User Interfaces (GUI) across the multiple operational personnel involved in air traffic management from the ramp tower to the en route controllers including an airline’s ATC Coordinator. Any Multiplexer information useful for the pilot such as predicted takeoff times can be relayed by the airline ATC coordinator. For efficiency and cost purposes, it is expected that the Multiplexer automation would interface directly with the FAA’s Traffic Management Advisor (TMA) for the en route controller and, whenever deployed by the FAA, the TFDM Arrival/Departure Management Tool (A/DMT) GUIs expected to be part of the future TFDM system [R09].

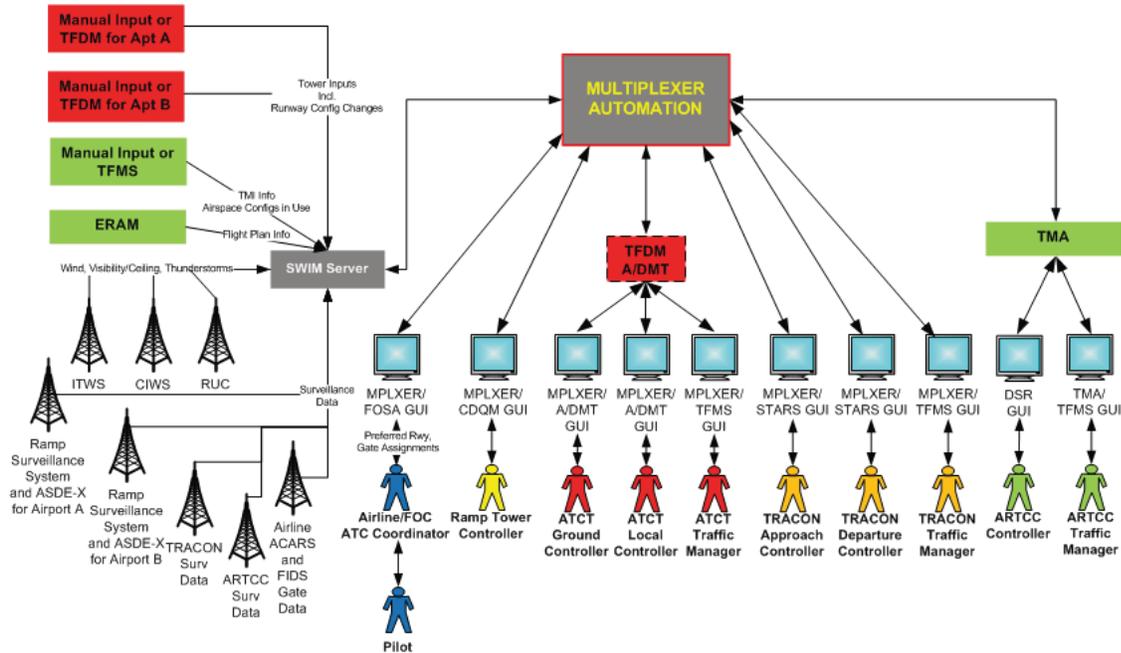


Figure 29: Nominal Multiplexer Automation System Interfaces

## 10.7 Human/System Interface

### 10.7.1 Air Traffic Controller Interface

Graphical User Interfaces (GUIs) for displaying the Multiplexer tool-generated target crossing times and assumed runway and fix assignments will be required. As shown in Figure 30, TMA-like time based metering displays in terms of either metering lists for the R-side controllers or timelines for the Traffic Management Units would be displayed on the appropriate TRACON displays (i.e., Standard Terminal Automation Replacement System (STARS)) or ARTCC displays (i.e., Display System Replacement (DSR) and TMA GUI). The assumed Multiplexer runway and fix assignments can be integrated into the far right, lowest line of the appropriate STARS or ERAM datablocks.

- TMA-like time-based metering displays (e.g., timelines, metering lists) provided at all key scheduling points from top-of-descent to top-of-climb

- Runway assignment and fix assignments in datatags

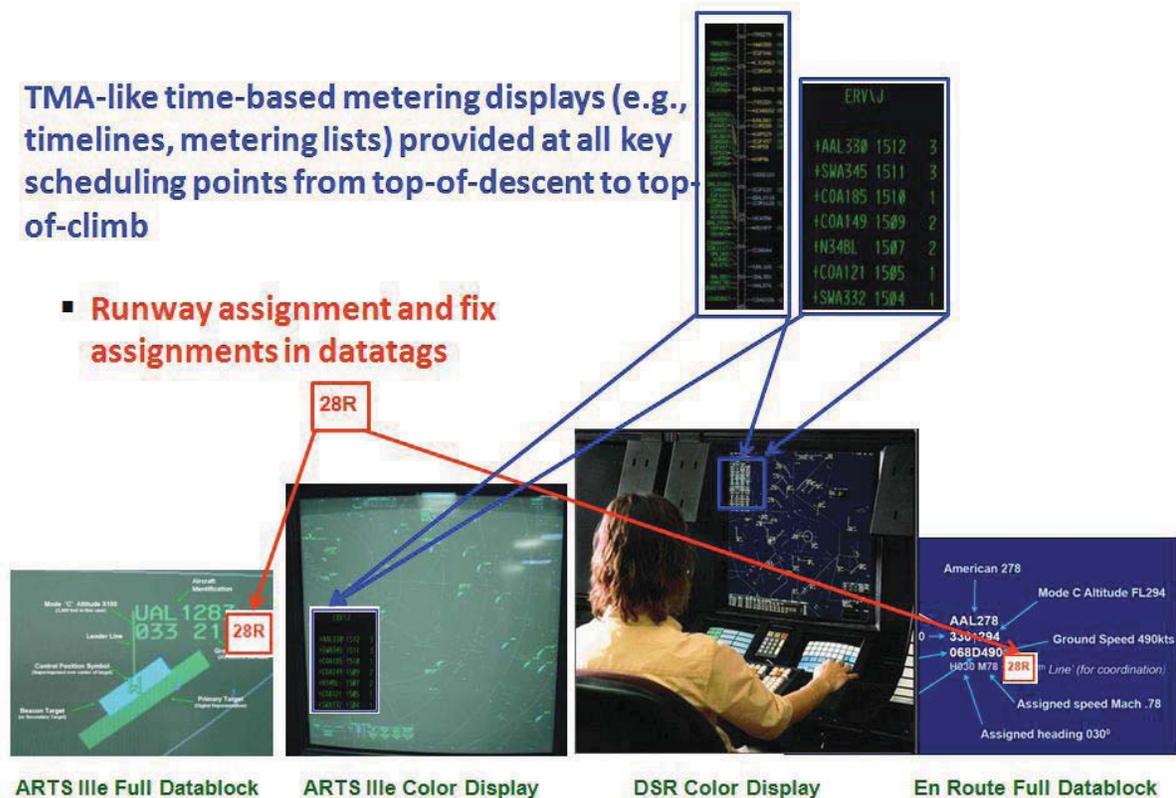


Figure 30: Nominal Air Traffic Controller Human/System Interfaces

### 10.7.2 Flight Deck Interface

There is no planned direct interface between the flight deck and the Multiplexer tool, but the AOC flight dispatcher can relay specific flight information via ACARS to the flight deck.

### 10.7.3 Airline Operations Center (AOC) Interface

AOCs will need to input dynamic AOC-driven information such as gate assignments. The Multiplex tool needs to communicate its planned take-off, landing, gate-in, and gate-out times with the AOC so that the AOC can plan optimal terminal-gate usage accordingly.

### 10.7.4 Traffic Flow Management (TFM) Interface

Airport surface /metroplex airspace congestion needs to trigger upstream TFM restrictions when necessary. Therefore, there is a need for an interface between the Multiplexer tool and the Traffic Flow Management System (TFMS). Also, certain dynamic information such as airport configuration, plans, and airspace configurations (e.g., SIDS, STARS, and RNAV) routes in use will require input by metroplex TFM specialists. Also, as shown in Figure 7.6, TMA-like time based metering displays in terms of timelines for the Traffic Managers would be displayed on the appropriate integrated TFMS or TMA GUIs.

## 10.8 Development Challenges

We expect the successful development and deployment of the Multiplexer automation to have a range of technical and other challenges. Some of the more prominent challenges are mentioned below.

### 10.8.1 Technical Challenges

Some of the key technical challenges that the development and deployment of the Multiplexer automation will encounter include:

- Development of fast and accurate aircraft flight prediction, scheduling, and sequencing algorithms for flights across an entire metroplex over the desired 1 hour look-a-head time,
- Development and integration of the Multiplexer scheduler output User Interface into existing or new decision support platforms (e.g., ERAM, ARTS, TFDM),
- Development of an integrated sensor and communication network backbone for the Multiplexer, and
- Integration of metroplex weather impacts on aircraft routing predictions to facilitate all-weather use of the Multiplexer.

### 10.8.2 Other Challenges

Some of the key other challenges that the development and deployment of the Multiplexer automation will encounter include:

- Establishment of aircraft operator to ANSP data exchanges (e.g., gate information),
- Integration of other potential overlapping planning systems (e.g., TMA, SDSS),
- Provision of a reliable source of: airport configuration status, plans, gate pushback predictions, and runway assignments, and
- ANSP acceptance of dynamic user trajectory preferences.

## 10.9 Future Multiplexer Capabilities

The Multiplexer concept described herein could be extended in a number of useful ways. Some of these potential future concept enhancements will now be discussed.

- The Multiplexer concept could be enhanced beyond pure metering to incorporate runway and fix change advisories. These advisories, if operationally acceptable, will provide future metroplex benefits.
- Promising new convective weather planning tools are currently being researched which could be integrated [DR08] [SB09]. The Multiplexer concept can incorporate convective

weather and generate metering advisories to ensure all-weather peak metroplex performance.

- The Multiplexer scheduling algorithms could be leveraged to support increasing levels of ANSP automation support including the enabling of “what if” metroplex configuration (both airspace and airport) impact assessments that could be used to analyze more efficient metroplex configurations. This idea could be further expanded to provide particular metroplex configuration switch recommendations or provide dynamic metroplex-focused Traffic Management Initiatives.
- Another category of enhancements would be through the incorporation of advanced procedures. The Multiplexer scheduling can be adapted to incorporate new “best-equipped/best-served” policies that are being discussed by key aviation stakeholders [M09]. The incorporation of these policies on metroplex flights will incentivize the adoption of aircraft avionics that will support improved aircraft trajectory predictability which in turn can lead to improved metroplex efficiencies.
- The Multiplexer scheduling can be adapted to more precise 10-second increments. This will require new air traffic control procedures and adaptation on the air traffic control automation platforms, but this can result in improved metroplex performance.
- To ensure the full incorporation and integration of flights from all airports in a metroplex into the Multiplexer scheduling, the development of lower-cost ATCT metering displays and inputs for non-major US airports would be helpful.
- Advanced terminal procedures that use dynamic reconfiguration of routes and flows to and from “dynamic anchor points” [F07] could be incorporated into the Multiplexer concept and this can benefit metroplexes that are particularly impacted by dynamic weather and those that typically use non-static ATC routing to deal with this [SR09].

Finally, the Multiplexer concept is integratable with other NextGen concepts being developed concurrently. Some of these concepts being developed include: System-Oriented Runway Management (SORM) [LB11], Airborne Precision Spacing (APS)/Interval Management (IM) [B06], Terminal Area Precision Scheduling and Spacing (TAPSS) system [ST11], and Controller Managed Spacing (CMS) [KC11].

## 11 References

- [AF09] FAA, “FAA Aerospace Forecast – Fiscal Years 2009-2025,” FAA, Washington DC, March 2009.

- [B06] Barmore, B., “Airborne Precision Spacing: a Trajectory-based Approach to Improve Terminal Area Operations,” 25<sup>th</sup> Digital Avionics Systems Conference, October 15, 2006.
- [BA09] Bailey, C., Smith, M., Tamburro, R., “Affecting Operational Change Through Analysis –Thanksgiving 2008,” AIAA 2009-6962, 9th AIAA Aviation Technology, Integration, and Operations Conference (ATIO), Hilton Head, South Carolina, September 2009.
- [BC06] Borener, S., Carr, G., Ballard, D., and Hasan, S., “Can NGATS Meet the Demands of the Future?” *Journal of Air Traffic Control*, Vol. 48 No. 1, pp. 34-38.
- [CR09] Clarke, J-P, Ren, L., Schleicher, D., et al., “Characterization of and Concepts for Metroplex Operations: Final Report,” NASA Project NNX07AP63A, September 30, 2009.
- [DR08] De Laura, R., M. Robinson, R. Todd, and K. MacKenzie, “Evaluation of Weather Impact Models in Departure Management Decision Support: Operational Performance of the Route Availability Planning Tool (RAPT Prototype),” 13<sup>th</sup> Conference on Aviation, Range, and Aerospace Meteorology, New Orleans, LA, 20-24 January, 2008.
- [DE95] Denery, D. G. and Erzberger, H., "The Center-TRACON Automation System: Simulation and Field Testing," Proceedings of the Advanced Workshop on ATM (ATM 95) sponsored by the National Research Council of Italy, Capri, Italy, 2-6 Oct. 1995; NASA TM-110366, Aug. 1995.
- [EDG93] Erzberger, H., Davis, T. J., and Green, S. M., “Design of center-TRACON automation system,” Proceedings of the AGARD Guidance and Control Panel 56th Symposium on Machine Intelligence in Air Traffic Management, Berlin, Germany, 1993.
- [F07] Fong, Robert, et al., “NASA NGATS ATM-Airspace Project: System-Level Design, Analysis, and Simulation Tools Technical Integration Report,” NASA Ames Research Center, Moffett Field, CA, September 30, 2007.
- [FAA07c] FAA, “New York/New Jersey/Philadelphia Metropolitan Area Airspace Redesign Final Environmental Impact Statement,” FAA, Washington DC, July 2007. Available at: [http://www.faa.gov/airports\\_airtraffic/air\\_traffic/nas\\_redesign/regional\\_guidance/eastern\\_reg/nynjphl\\_redesign/feis/](http://www.faa.gov/airports_airtraffic/air_traffic/nas_redesign/regional_guidance/eastern_reg/nynjphl_redesign/feis/) (accessed: January 20, 2009).
- [FAA07b] “Operational Evolution Partnership Version 1.0 2007 – 2025, Executive Overview,” FAA, Washington DC, June 2007. Available at: [http://www.faa.gov/about/office\\_org/headquarters\\_offices/ato/publications/oepr/vepr/1/overview/overview.pdf](http://www.faa.gov/about/office_org/headquarters_offices/ato/publications/oepr/vepr/1/overview/overview.pdf) (accessed: April 30, 2008).

- [FAA08b] “NextGen Implementation Plan – OEP Metropolitan Areas,” FAA, Washington DC, July 7, 2008. Available at: [http://www.faa.gov/about/office\\_org/headquarters\\_offices/ato/publications/nextgenplan/0608/solution\\_sets/metro/](http://www.faa.gov/about/office_org/headquarters_offices/ato/publications/nextgenplan/0608/solution_sets/metro/) (accessed: September 6, 2008).
- [JPDO-07] JPDO, “Concept of operations for the next generation air transportation system,” Version 2.0, JPDO, Washington DC, June 2007.
- [KC11] Kupfer, M., Callantine, T., Martin, L., Mercer, J., and E. Palmer, “Controller Support Tools for Schedule-Based Terminal-Area Operations,” 9<sup>th</sup> USA/Europe Air Traffic Management Research and Development Seminar, Berlin, Germany, June 2011.
- [LB11] Lohr, G. W., Brown, S., Stough, H. P., Atkins, S., Eisenhower, S., and D. Long, “System Oriented Runway Management: A Research Update,” 9<sup>th</sup> USA/Europe Air Traffic Management Research and Development Seminar, Berlin, Germany, June 2011.
- [M09] May, Jim, “ATA Testimony by CEO Jim May before the House T&I Aviation Subcommittee on ATC Modernization,” March 18, 2009, accessed at [http://www.airlines.org/PublicPolicy/Testimony/Pages/testimony\\_3-17-09HouseTI.aspx](http://www.airlines.org/PublicPolicy/Testimony/Pages/testimony_3-17-09HouseTI.aspx), June 2011.
- [MC09] McClain, Evan, Clarke, John-Paul, Huang, Alex, and David Schleicher, “Traffic Volume Intersection Metric for Metroplex Clustering Analysis,” AIAA-2009-6069, AIAA Guidance, Navigation, and Control Conference, Chicago, IL, 10-13 Aug 2009.
- [R09] Ryan, S., “Tower Flight Data Manager (TFDM),” presentation to Marshall Mowery, FAA ATO-T System Engineering, November 5, 2009.
- [RE09-1] Ren, L., et. al., “NASA Project NNX07AP63A Characterization of and Concepts for Metroplex Operations: Final Report,” NASA Contract No. NNX07AP63A, unpublished, July 2009.
- [RE09-2] Ren, L., et. al., “Characterization of and Concepts for Metroplex Operations: Atlanta Large TRACON (A80) Site Survey Report, NASA Contract No. NNX07AP63A, unpublished, July 2009.
- [RT09] RTCA NextGen Mid-Term Implementation Task Force, “NextGen Mid-Term Implementation Task Force Report,” RTCA Task Force 5, Washington DC, September 9, 2009.

- [SB09] Song, L., Bateman, H., Masalonis, A., and C. Wanke, "Integrated Collaborative Departure Traffic Management," AIAA-2009-6932, 9<sup>th</sup> AIAA Aviation Technology, Integration, and Operations (ATIO) Conference, Hilton Head, South Carolina, 21-23 September, 2009.
- [SC09-1] Schleicher, D., et. al., "Characterization of and Concepts for Metroplex Operations: SCT Site Report," NASA Contract No. NNX07AP63A, unpublished, March 2009.
- [SC09-2] Schleicher, D. R., et. al., "Characterization of and Concepts for Metroplex Operations: Miami Site Survey Report," NASA Contract No. NNX07AP63A, unpublished, May 2009.
- [SC11] Schleicher, D., Clarke, J.-P., Saraf, A., and S. Timar, "A Concept of Operations for a NextGen Metroplex Scheduling Concept," accepted paper, 11<sup>th</sup> AIAA Aviation Technology, Integration and Operations Conference, Virginia Beach, Virginia, 21-22 September 2011.
- [SR09] Schleicher, David R., Ren, Liling, Gutterud, Rich, Timar, Sebastian, Crisp, Donald L., Lewis, Taryn Butler, Clarke John-Paul, and Aditya Saraf, "NASA Project NNX07AP63A Characterization of and Concepts for Metroplex Operations: *Miami Site Survey Report*," May 5, 2009
- [ST11] Swenson, H. N., Thippavong, J., Sadovsky, A., Chen, L., Sullivan, C., and L. Martin, "Design and Evaluation of the Terminal Area Precision Scheduling and Spacity System Trajectory and Queue Management," 9<sup>th</sup> USA/Europe Air Traffic Management Research and Development Seminar, Berlin, Germany, June 2011.
- [TI09] Timar, S., et. al., "Characterization of and Concepts for Metroplex Operations: New York Site Report," NASA Contract No. NNX07AP63A, unpublished, March 2009.
- [TI10] Timar, S., et. al., "Metroplex Demand Analysis and Applications," AIAA 2010-7611, AIAA Modeling and Simulation Technologies Conference, 2 - 5 August 2010, Toronto, Ontario Canada.
- [W00] Wong, G. L., "The dynamic planner: The sequencer, scheduler, and runway allocator for air traffic control automation" NASA/TM-2000-209586, April 2000.
- [W11] [http://en.wikipedia.org/wiki/Time\\_division\\_multiple\\_access](http://en.wikipedia.org/wiki/Time_division_multiple_access), accessed June 2011.

## APPENDIX A – Simmod PRO! Software Tool



# 1 Document Purpose

As described in the statement of work for this project, a report shall be provided that will “describe how the Simmod *PRO!* software tool works to facilitate understanding of how the results of subtasks 5.2.2, 5.3.1, and 5.3.2 and any other simulations were derived. This requirement is essential for NASA to understand how the simulations/tools operate to properly interpret results obtained from this study.”

There is no proprietary information contaminated in this document.

## 2 Introduction

Simmod *PRO!* is an advanced simulation tool used for conducting analyses not addressable by other models available in the market. It provides the flexibility and power of true rules-based modeling capability through the implementation of a generalized simulation scripting language. This greatly expands the capabilities to simulate the dynamics, variability, site-specific features and situation-specific factors in air traffic operations. Simmod *PRO!* was developed in 1997 and maintains its state-of-the-art capabilities through periodic updates and enhancements and the continuous application of the software by airports, airlines, consultancies, air navigation service providers, and universities and research institutes.

## 3 Applications

Simmod *PRO!*'s modeling power is derived from user-defined rules that query the state of the simulation and provide dynamic decision-making. Each step of a flight can be controlled based on user-defined rules that respond to the ever changing conditions in the air or on the ground to allow modeling of:

- Dynamic airfield and airspace rerouting
- Complex taxiing operations
- Departure queue sequencing
- Disruptive events (e.g., weather, system failures, runway closures, very large aircraft)
- Advanced operating concepts
- Complex interactions among neighboring airports

## 4 Features

The rules-based input provides for “if-then” capability that specifies the actions to be taken by the simulation based on the state of the system. Examples of the dynamic states of the system that can be queried to control the simulation include:

- Resource availability, such as de-icing trucks and ground support equipment
- Departure queue length and the duration of those aircraft in the queues
- The number and types of aircraft on the ground or in the air, on a link or at a node
- Gate occupancy, staging area utilization, de-icing pad capacity
- A flight's aircraft type, airline, origin, destination or location on the ground or in the air

## 5 Benefits

Simmod *PRO!* provides the capability to conduct complex simulation analyses not achievable by other modeling tools currently on the market. Examples of the benefits of applying Simmod *PRO!*:

- Integration of airfield and airspace resources directly into the simulation
- Flexibility to test and optimize new procedures related to de-icing, gate procedures, cargo operations, taxi paths, departure sequencing, and staging of aircraft
- Enhanced rules based logic allows for advanced user input
- Resource scheduling and tracking within the simulation model for non-aircraft elements or components such as de-ice vehicles, buses, tow vehicles, etc.

## 6 Hardware Requirements

Simmod *PRO!* runs on standard desktop and laptop PCs with:

- Windows XP, Vista, or Windows 7
- Minimum of 1GB memory
- 100 GB of hard disk space

## 7 Simmod Product Description

The Airport and Airspace Simulation Model (SIMMOD) is an industry standard analysis tool used by airport planners and operators, airlines, airspace designers, and air traffic control authorities for conducting high-fidelity simulations of current and proposed airport and airspace operations. SIMMOD has been in continual use and development since the early 1980s. ATAC was one of the original developers and we had been funded for numerous years by the US Federal Aviation Administration (FAA) and NASA to provide software development services and customer support. When US government budgets became tight in the late 1990s, this funding was stopped; however, ATAC continued to provide development efforts and customer support to the SIMMOD user community. It is currently used by airport authorities, government agencies, and consultants throughout the world. ATAC releases software enhancements each year that are provided at no additional cost to our supported customers.

The SIMMOD simulation logic is embodied in an executable program referred to as the “engine”, which is driven by user input parameters. It is designed to “play out” airport and airspace operations within the computer and calculate the real-world consequences of potential operating conditions. It has the capability and flexibility to address a wide range of “what if” questions related to airport and airspace capacity, delay, and efficiency, including questions associated with:

- Existing or proposed airport facilities (e.g., gates, taxiways, runways, pads)
- Airport operating alternatives (e.g., taxi patterns, runway use, departure queuing)
- Existing or proposed airspace structures (e.g., routes, procedures, sectors)
- Air traffic management/control technologies, procedures, and policies
- Aircraft separation standards parameters (e.g., weather, aircraft type, flight state)
- Airline operations (e.g., flight schedule, banking, gate use and service times)
- Current and future traffic demand (e.g., volume, aircraft mix, new aircraft types)

Based upon a user-input scenario, SIMMOD tracks the movement of individual aircraft through an airport/airspace system, detects potential violations of separations and operating procedures,

and simulates air traffic control actions required to resolve potential conflicts. The model properly captures the interactions within and between airspace and airport operations, including interactions among multiple neighboring airports.

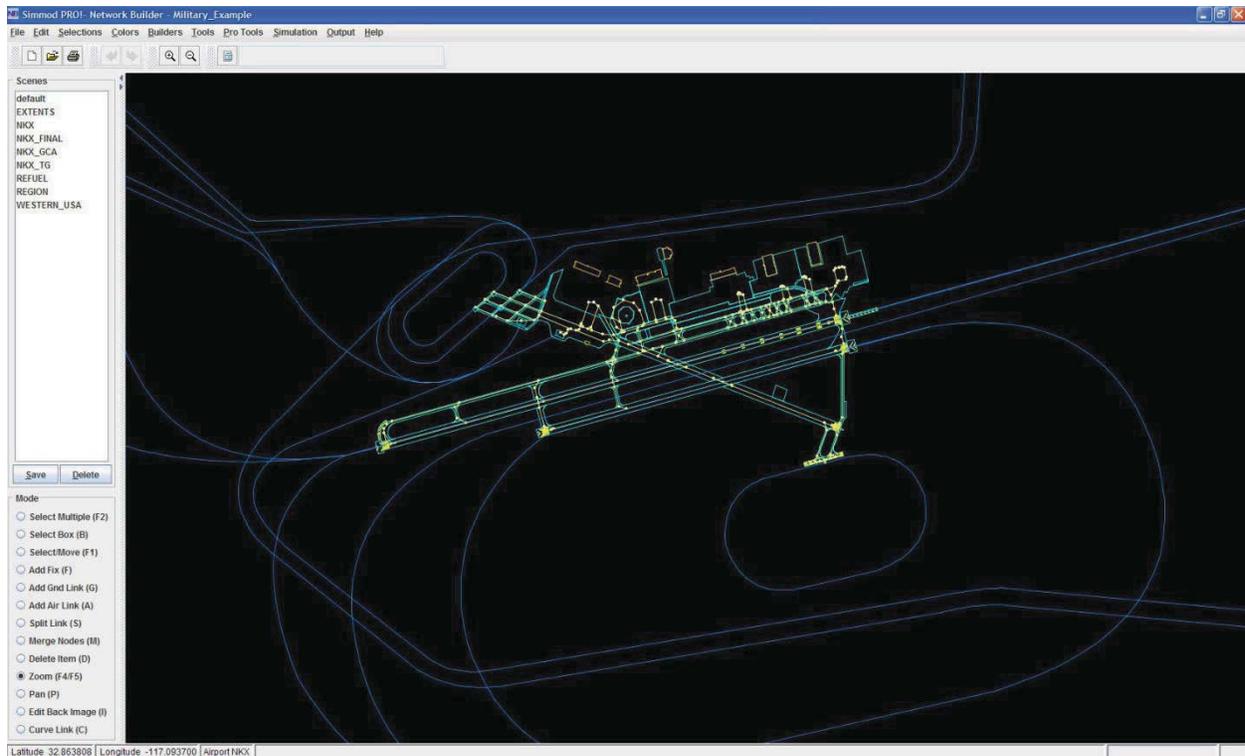
As SIMMOD simulates airport and airspace operations, it computes and records detailed information on the activities and events associated with the operation of each aircraft on the airport and within the airspace. These results are provided as outputs that are available to the user for evaluating alternatives; including aircraft travel time, delay and operating costs, and system capacity, throughput and traffic loading.

SIMMOD reads the model inputs in the form of text files, runs the simulation, and generates output in the form of text files. As a legacy of the time when the US government funded engine development, ATAC continues to make the latest engine available as a free download from our website. However, it is very difficult to use — the input text files required by SIMMOD follow a very complex format and the engine has no embedded output processing or visualization tool. As an extensive practitioner of airspace and airport simulation analysis, ATAC has created a user-friendly version of the software called Simmod *PRO!*. Simmod *PRO!* includes SIMMOD as the core simulation engine, but it adds a graphical-user interface (GUI) for data input, a traffic animator to visualize the results, and reporting tools to analyze the results.

Simmod *PRO!* can significantly reduce SIMMOD application development time by providing a user with the following features:

- An easy-to-use graphical user interface to design, build, and execute the simulation
- A Network Builder that enables a user to graphically construct the node-link structure of the airport/airspace system being studied
- Relational database tables that store and manage simulation data
- Error and consistency checks of input data
- Easy operation with other desktop applications (Excel, Access, etc.)
- A 2D traffic animator that facilitates input preparation and aids in analysis and presentation of results
- A reporting module that allows detailed results to be quickly analyzed

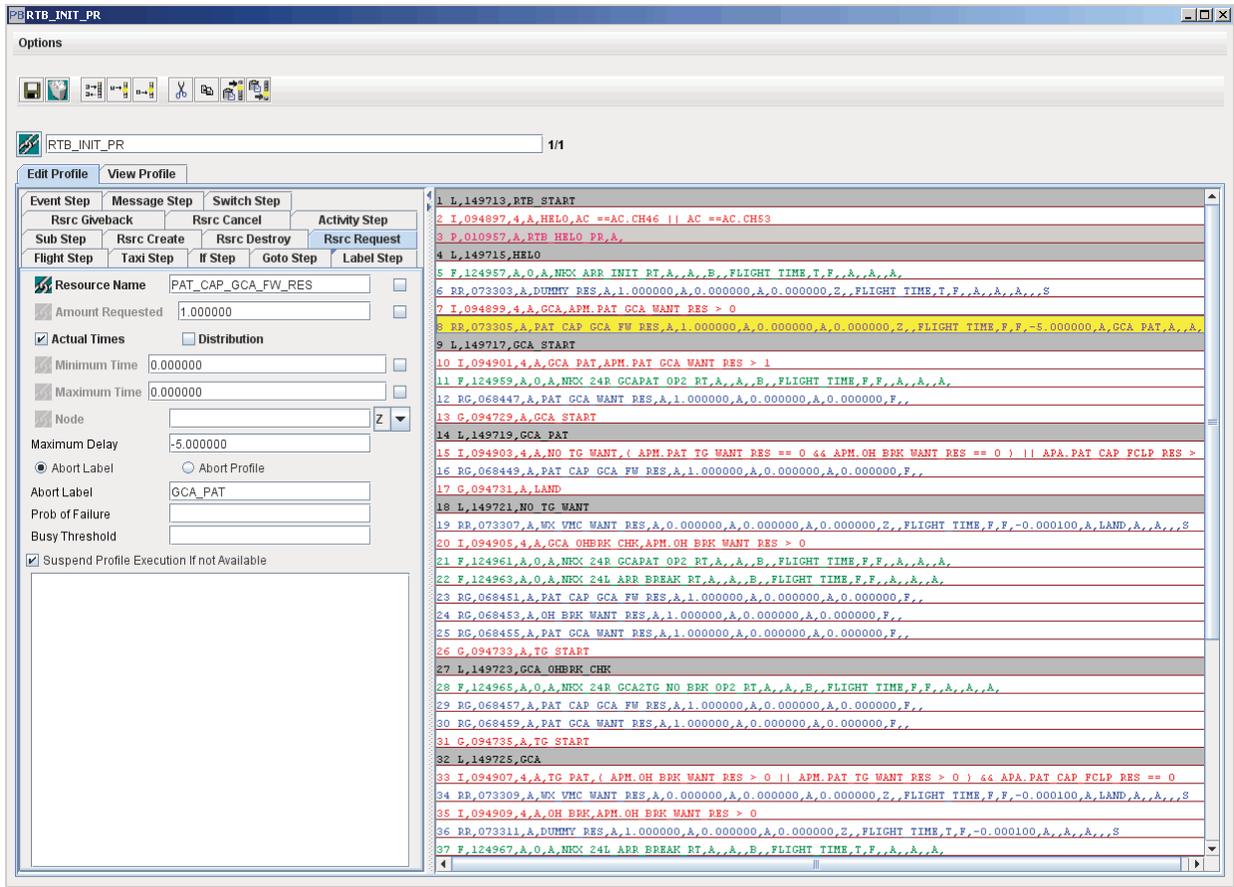
The Network Builder shown in Figure 7-1 below provides the capability to model multiple airports, each having multiple runways, taxiways, gates, deicing areas, staging areas, departure queues and concourses, as well as extremely detailed airspace routes and sectors. Importing CAD drawings of the airfield and terminal into the simulation allows quick development of a very accurate and detailed ground model. A user can build a high-fidelity airspace model with the aid of the Simmod *PRO!* Network Builder can display airspace boundaries, aerial photos, navigation aids, and fix locations, and standard GIS data. In addition, the Network Builder incorporates many tools that allow the analyst to easily implement the complex capabilities of the simulation engine.



**Figure 7-1 Simmod PRO! Network Builder**

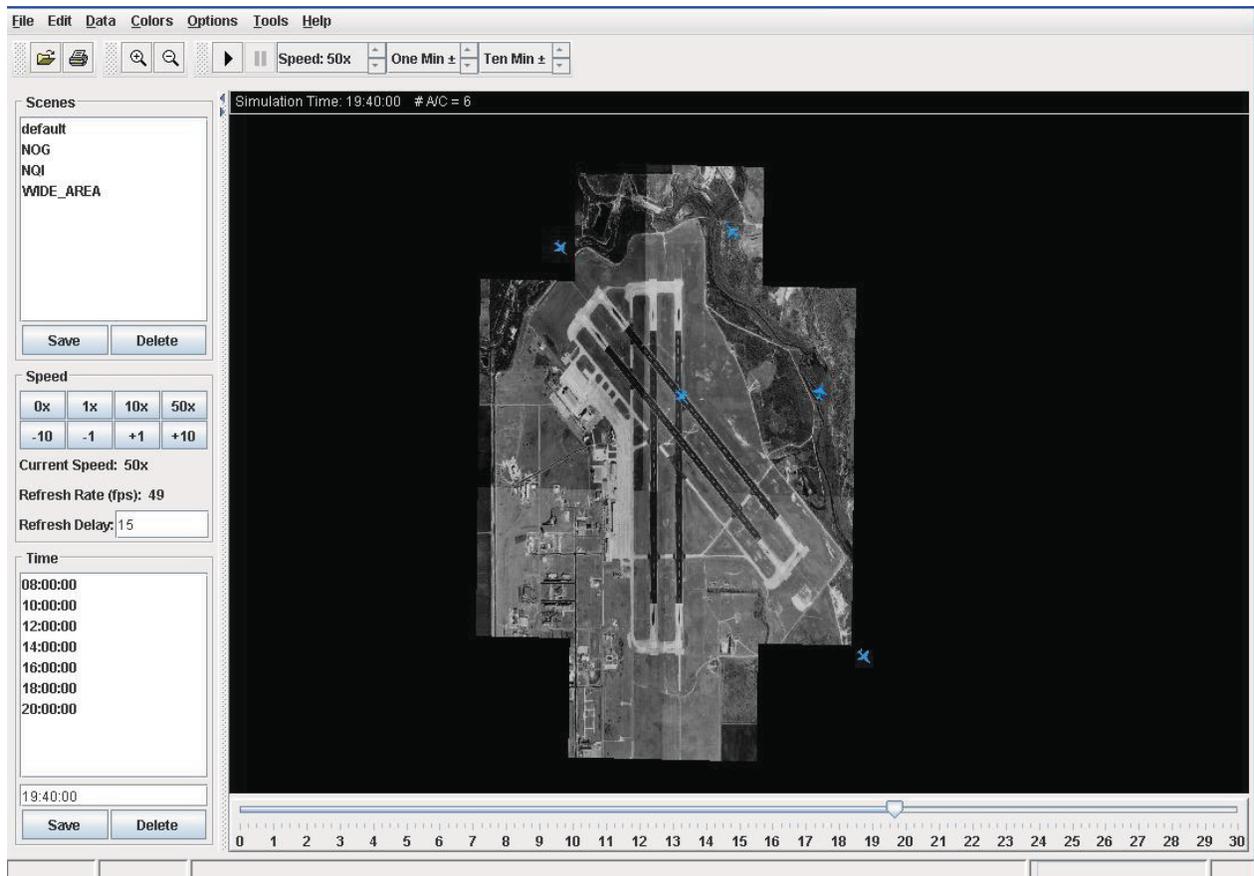
Simmod *PRO!* provides the capability for the user to specify rules and decision logic that responds to the state of the airport/airspace system and invoke the SIMMOD engine to adapt to the changing environment. Output from the model includes results on the use of these rules and the resulting decisions made during the simulation, providing a corresponding set of output statistics.

Rules take the form of a scripting language that allows the user to create “if...then...” logic. These rules can be as simple or complex as desired by the user. The feature provides the capability to implement logic to accurately model site-specific ground and air operational details, the dynamics of operations that depend on the state of the system, and probabilistic behavior of system elements. These rules are created as needed through the Profile Builder, as shown in the Figure below.



**Figure 7-2 Simmod PRO! Profile Builder**

The Simmod *PRO!* Animator presents a detailed view of simulated aircraft operations, both on the ground and in the air. The user can view detailed flight and aircraft information during the animation replay simply by clicking on the aircraft's icon. The Animator can import and display graphical information as background to the animation, such as noise contours, terrain, and other GIS data. Figure 7-3 below presents a screen capture of the animator.



**Figure 7-3 Simmod PRO! Animator**

The Reporting Module includes over 800 preset reports as well as the capability to create customized reports. It enables the analyst to easily determine output statistics down to the aircraft level or at any aggregate level. In addition, output is compatible with other desktop applications, allowing for ease of additional data analysis.

## 8 How SIMMOD Works

### 8.1 SIMMOD is a Discrete-Event Simulation

SIMMOD is a discrete-event simulation model: it represents a system evolving over time by means of a mathematical model, the state of which changes at discrete points in time. These points are those at which an event occurs, where an event is an instantaneous occurrence that changes the state variables.

Consider a simple runway departure queue. An airplane joining this queue is an event.

SIMMOD calculates the effect of this new event on the existing system and modifies or adds state variables resulting from this event before considering the next event.

Suppose you wish to estimate the average delay of planes in this basic runway queue. The state variables for the simplest possible model would include the following:

- The status of the runway (occupied or empty)
- The number of planes in the departure queue for this runway (if any)
- The time at which each plane enters the departure queue

The status of the runway is needed to determine whether an aircraft can advance to the runway immediately when it reaches the front of the queue.

Assume that when a plane departs from the runway, the runway is available for the next departure. When a plane departs, SIMMOD checks the departure queue. If there are no aircraft waiting in the queue, the runway remains empty; otherwise, the first aircraft waiting in the queue occupies the runway and departs.

The time at which each aircraft enters the queue is used to compute its delay in the queue, the difference between the time it enters and the time it leaves the queue for departure.

There are two different events in this example: (1) entering the queue and (2) leaving the runway. Entering the queue is an event because it causes the runway status (a state variable) to change from empty to occupied or it increases the number of aircraft in the queue (another state variable) by one. Correspondingly, leaving the runway is an event because it causes the runway status to change from occupied to empty or decreases the number in the queue by one.

## 8.2 The Event Schedule

The event schedule and the simulation clock work together to process SIMMOD events in the proper sequence. SIMMOD is sometimes called an “event stepped” simulation, because it steps forward from one event to the next. With each step, the model bypasses an interval in which no events occur. It processes each event strictly according to the order of its appearance in the event schedule.

If two events are scheduled to occur at the same time, SIMMOD prioritizes them and processes them in sequence, without moving the simulation clock forward. The priority can be tracked by examining the Simulation Log, which lists every event in chronological order. Figure 8-1 shows a simple event schedule.

Table 8-1: A Simple Event List

Flight ID	Event Time	Event Type
AA 234	11:40	Arrive at node 120
TW 11	11:42	Finish loading at gate S12
CO 888	11:44	Arrive at node 320
GA 25	11:44	Check for end of hold at node 33

When SIMMOD finishes processing an event (which includes updating the state variables changed by the event) it checks the event schedule for the next event. Each event can cause other events to be added to the schedule. Most of the events that occur in a typical simulation are generated through this chain reaction process. External events, defined and scheduled by SIMMOD user input data, can cause many series of subsequent, interacting events.

## 8.3 External Events

External events are defined and scheduled directly by data input. One external event defined in the input data can generate a series of internal events. There are three types of external events in SIMMOD:

- **Initiating flights.** You schedule the creation (or injection) of flights in SIMMOD. Thereafter, the aircraft proceed automatically through the simulation according to

their flight information, route assignment, established procedures, and so on. See Section 9, Flights, for more complete information.

- **Resetting parameters.** You can schedule changes in the airport and airspace network and the air traffic control logic during the simulation. These changes are discussed in Section 15, Resetting Simulation Parameters.
- **Specifying traces (SIMMOD output).** You specify the kinds of information to be collected for analysis and the times at which to collect it.

## 8.4 The Simulation Clock

SIMMOD keeps track of the current value of simulated time as it proceeds and advances simulated time from one value to another. Processing the next event updates the simulation clock time.

At the start of a simulation run, SIMMOD initializes the simulation clock to 00:00 and schedules the times of external events. Then it advances the simulation clock to the time of the first event, updates the state of the system to account for changes made by the event, and adds to the schedule internal events generated by the event. Then it advances the simulation clock to the time of the next event and repeats the process. SIMMOD continues advancing the simulation clock from one event to another and updating the system until it reaches the end of the event list or a user-specified event that ends the simulation.

Note that the simulation clock cannot reverse to process an event. Thus, external events must appear on the schedule in the correct time sequence, or SIMMOD will incur a “fatal error” and the simulation will stop.

## 8.5 SIMMOD Nodes and Links: Basic Modeling Components

SIMMOD represents an airport or airspace system as a series of nodes connected by links. A node is a point in a coordinate system where SIMMOD evaluates an aircraft’s position with respect to other aircraft in the system. A link defines the path between two nodes. Aircraft move from one node to another only along a defined link. \

SIMMOD maintains ground (airfield) nodes and airspace nodes as separate groups. Ground nodes describe airfield locations such as gates, departure queues, or runway and taxiway intersections. Airspace nodes describe airspace locations such as navigational fixes, holding queues, merge points for routes, or interfaces with an airport.

SIMMOD also maintains ground and airspace links separately. Ground links can represent taxiways and runways. Airspace links can represent routes. Larger structures (e.g., runways and routes) are usually composed of several links.

Nodes and links in the airspace and on the airfield have slightly different characteristics, which are discussed at length in Section 12, Airspace Logic, and Section 13, Airfield Logic.

Note: For more information on discrete-event simulation and simulation modeling in SIMSCRIPT II.5®, see *Building Simulation Models with SIMSCRIPT II.5* by Edward C. Russell (CACI Inc.-Fed., 1983); *SIMSCRIPT II.5 Reference Handbook, 3rd Edition* (CACI Inc.-Fed., 1989); or *Simulation Modeling and Analysis* by Averill M. Law and W. David Kelton (McGraw-Hill, 1982).

## 8.6 Stochastic Processes

SIMMOD is a stochastic model. It uses random variables to produce unique output representing day-to-day variations in air traffic phenomena. Because SIMMOD is designed to produce

realistic results from any iteration of a defined application data set, it is usually necessary to run several iterations with a single data set in order to establish statistically significant tendencies. For a run of several consecutive iterations, the Report program produces aggregate values and, where appropriate, averages and standard deviations. For example, where frequencies indicating the usage of a given facility are reported for a single-iteration run, average frequencies are reported for a multiple-iteration run.

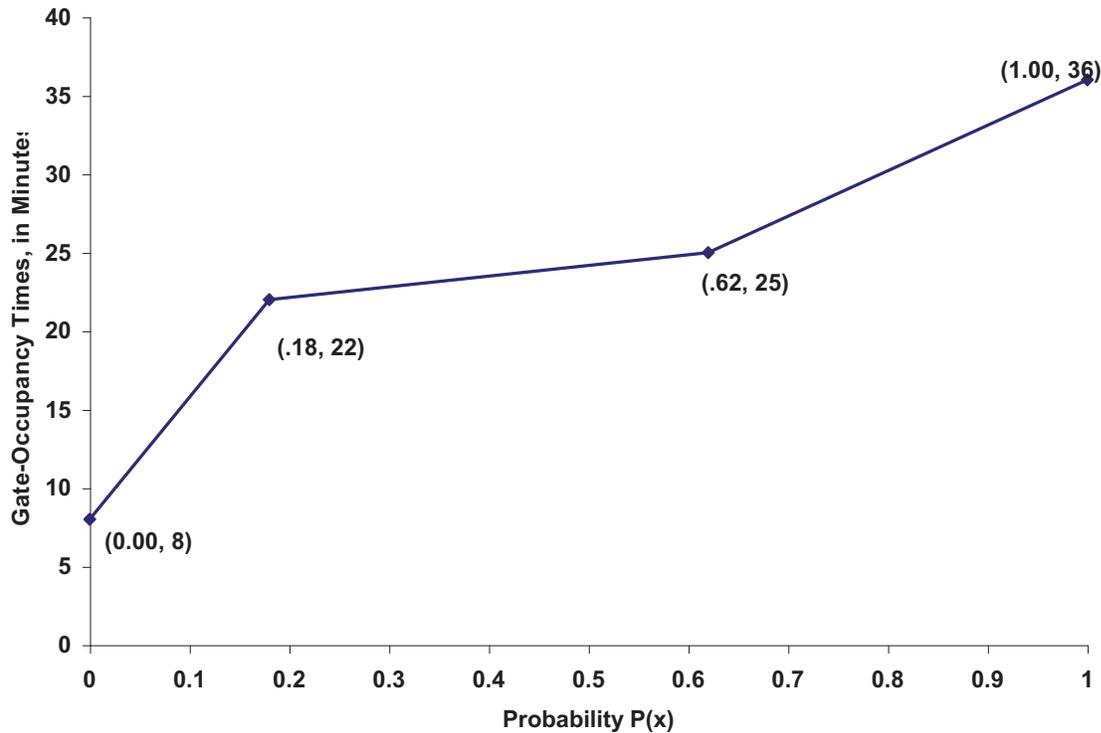
## **8.7 Random Linear Variables**

SIMMOD uses random linear variables to simulate certain airport and airspace phenomena. You can provide variation in your model by defining distribution values for the following variables:

- Gate-occupancy times (for loading or unloading passengers)
- Injection time of multiple arrivals and departures
- The cloning of arrivals and departures
- Takeoff and landing roll distances
- Intrail separation multiplier (to vary the defined separation requirements)
- Lateness of flights
- The probability of holding flights to accommodate late arrivals in hubbing operations
- The transfer time (time for unloading and loading passengers) between flights in hubbing operations
- Push-back/power-back times
- Runway crossing start-up times
- Slot window times

User-defined cumulative probability distributions determine the amount of variation. SIMMOD generates a random real number between 0 and 1, and uses this number to select values from the distribution.

Cumulative probability distributions are defined by pairs of numbers. The first number in the pair represents the probability of a value less than or equal to  $x$  occurring; the second defines the corresponding value  $x$ . Two such pairs, one indicating 0% probability for the lower value in the distribution range and the other 100% probability for the upper value, defines a basic linear function. Additional pairs specify a more detailed distribution.



**Figure 8-1: Cumulative Distribution of Gate Occupancy Times for Commuter Aircraft**

Figure 8-1 shows a cumulative distribution that defines the range and probabilities for gate-occupancy times for commuter aircraft. This distribution has four number pairs: (0.0, 8); (0.18, 22); (0.62, 25); and (1.0, 36). The lowest value that can be returned is 8 minutes; the highest is 36 minutes. There is an 18% probability that the gate time will be 22 minutes or less, and a 62% probability that it will be 25 minutes or less. Note that a percentage is entered as a number between 0.0 and 1.0; 100% is entered as 1.0. In this example, 62% was entered as 0.62.

## 8.8 Random Number Streams and Seeds

Because SIMMOD uses many random numbers in every run of an application data set, it creates a sequence, or stream, of random numbers for each iteration. These streams are created by a random number generator built into SIMSCRIPT II.5, the language in which SIMMOD is programmed.

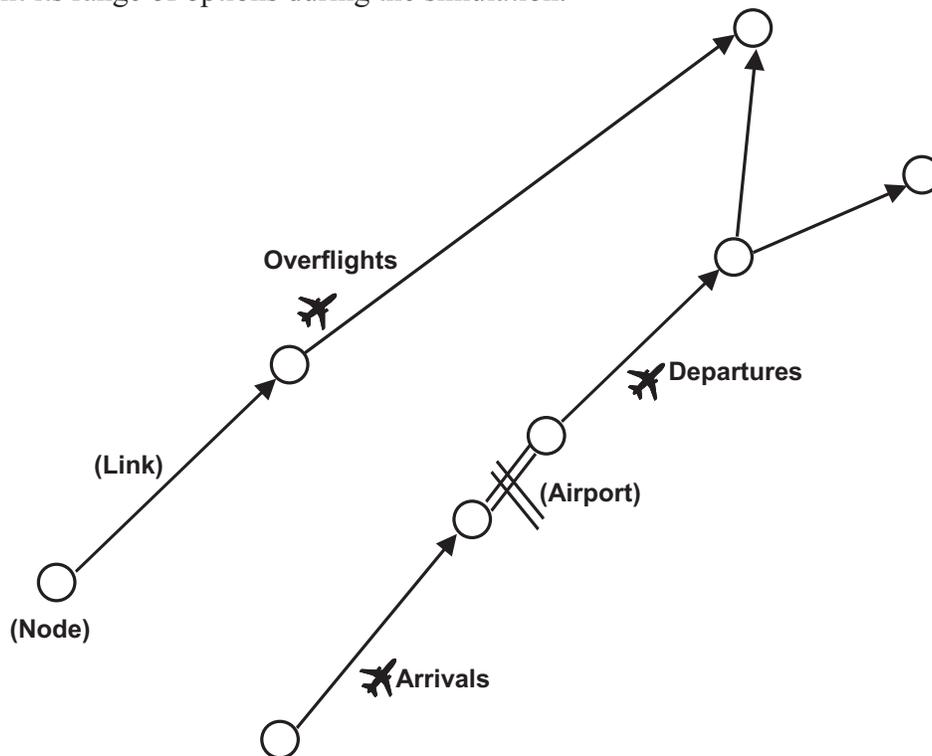
The first number in a random number stream is called the seed. The random number generator uses this seed to produce the ensuing stream. Starting with the same seed, the random number generator will always produce exactly the same random number stream (assuming that the simulation is run on a machine with the same processor). This is significant because it allows you to reproduce the simulation results achieved with a given data set. See Section 16, Stochastic Processes, for a further discussion of randomness in SIMMOD.

## 9 Flights

In SIMMOD, a flight is an aircraft with a unique identifier (ID) and a set of data, including its:

- Type of flight
- Starting time
- Airspace route.

Depending on the scenario being simulated, the user can define other data to restrict the flight's path and limit its range of options during the simulation.



**Figure 9-1: Flight Types**

As shown in Figure 9-1, there are three types of flights:

- Arrivals at an airport
- Departures from an airport
- Flights passing through the airspace that do not land (overflights).

### 9.1 Arrivals

A SIMMOD arrival flight always starts in the airspace. The basic arrival consists of a flight that:

- Traverses an airspace route
- Lands on a runway
- Taxies to a gate
- Unloads passengers (i.e., occupies a gate)
- Exits the simulation.

#### Turnaround

An arrival flight can also turn around at an airport and depart. This means an arrival and departure are dependent on each other. If the arrival is late, the dependent departure must await

the arrival before it can load passengers and depart. The arrival with a turnaround departure consists of a flight that:

- Traverses an airspace route
- Lands on a runway
- Taxis to a gate
- Unloads passengers
- Loads passengers
- Taxis to a runway
- Takes-off on a runway
- Traverses an airspace route
- Exits the simulation.

### **Overflights**

An overflight is an arrival that does not land at an airport. The end of the arrival route is not an airport, so the flight begins and ends in airspace. After traversing its airspace route, the overflight exits the simulation.

## **9.2 Degenerate airports**

When the simulation does not include an airfield in its scenario, arrivals use a “degenerate airport”. The physical attributes of such an airport, other than its location at the end of an airspace route, are not modeled. The arrival at a degenerate airport consists of a flight that traverses an airspace route and exits the simulation.

No ground simulation occurs at degenerate airports. If the SIMMOD analysis project concerns only the airspace, this frees the user from generating airfield data. The degenerate airport also offers greater flexibility, as arriving flights are not subject to restrictive airport landing procedures.

All arrivals in the simulation must be created in the airspace, so the simulation must include a minimally defined airspace structure even when the ground simulation is of primary concern. To simplify the airspace definition, it is possible to define the required airspace route as a single link. This link will represent the final approach path for arrivals and will allow the runway management to be accurately modeled.

## **9.3 Departures**

A SIMMOD departure flight always starts at an airport. The basic departure (which is called an "emplane" for the purposes of simulation) consists of a flight that is created at a gate and:

- Loads passengers
- Taxis to a runway
- Takes off on a runway
- Traverses the airspace route
- Exits the simulation.

A departure can be defined by input data to generate an arrival at another airport. When a departure reaches the end of its departure route, the simulation will create a dependent arrival on a route leading to that airport. The departure with a dependent arrival consists of a flight that is created at a gate and:

- Loads passengers
- Taxis to a runway

- Takes off on a runway
- Traverses the airspace departure route
- Becomes a dependent arrival
- Traverses an airspace arrival route
- Lands on a runway
- Taxis to a gate
- Unloads passengers
- Exits the simulation.

When the airfield is not included in the modeled scenario, the departures are created at a degenerate airport. The physical attributes of such an airport, other than its location at the end of an airspace route, are not modeled. A departure at a degenerate airport consists of a flight traversing an airspace route and either exiting the simulation or creating a dependent arrival.

No ground simulation occurs at degenerate airports.

All departures must enter the airspace before exiting the simulation, so a ground simulation must have at least a minimally defined airspace structure. To simplify the airspace definition, it is possible to define each route as a single link. This link will represent the initial airspace path for departures and allow the runway management to be accurately modeled.

#### **9.4 Multiple, Stochastically Generated Flights**

The flights described above are either created individually by the user as external events or as dependent arrivals and departures.

If individual flight information is not available or not required, the simulation can create flights using its multiple arrival (MULTARR) and multiple departure (MULTDEP) features. These allow the simulation to create a number of flights stochastically (i.e., randomly) over a given time period. The number of flights and the time period are defined by input data.

Stochastically generated aircraft can represent additional flights needed to model congestion in the modeled airport or airspace, e.g., general aviation (GA) flights or flights from other airlines. These aircraft can also represent projected scheduling where real numbers do not exist.

#### **9.5 Cloning**

Cloning selectively increases or decreases traffic on specific routes, and provides a convenient means of modifying the existing schedule while still reflecting the traffic pattern defined by the user's input data. Cloning is especially helpful in estimating traffic congestion based on a projected increase or decrease in scheduling.

Cloning replicates or removes individually defined arriving and departing flights. Stochastically generated flights, described above, may not be cloned.

Cloning can be set to take effect for any time interval on any route. For example, one may need to simulate the effects of a 20% increase in traffic on one route and a 10% decrease on another during peak morning hours. Cloning is an appropriate approach to this problem because there is no empirical data to represent the change, yet the analysis requires modeling of a specific increases and decreases in traffic on defined airspace routes during a given period of time.

The process of cloning involves a random draw, which applies a realistic uncertainty to the duplication of existing flights.

When initializing a flight, SIMMOD checks whether cloning is in effect. If cloning is in effect, the cloning probability is checked for the flight. This probability will range from -100 to 500 percent. If the probability is negative, the flight may be removed from the simulation.

If the cloning probability value is greater than 100, the flight will be cloned once for each 100 percent of probability defined. The number of clones may then be increased by one, based on the difference between the total percentage of probability and the same percentage rounded to the next lowest 100.

The simulation makes the decision to create this extra clone (increasing the number by one) by drawing a random number between 0 and 100. This number is then compared to the probability difference mentioned above. If the random number is less than the probability difference, the flight is cloned. If it is greater, no extra clone is created.

Thus, if the cloning probability for a flight is defined as 267 percent, the flight will be cloned at least twice because the probability exceeds 200 percent. If the random number drawn is less than 67, the flight will be cloned a total of three times.

The following chart shows the options based on the range of cloning probability values:

**Table 9-1: Cloning Flights**

<b>Probability</b>	<b>Number of aircraft cloned</b>
-100% to 0%	-1 or 0
0% to 100%	0 or 1
100% to 200%	1 or 2
200% to 300%	2 or 3
300% to 400%	3 or 4
400% to 500%	4 or 5

A cloned flight’s injection time is based on the lateness distribution specified for the original flight. Thus, if three clone flights are generated, they will be injected into the simulation at different times.

## **10 Aircraft Definition**

Every flight created by the simulation is identified as a certain aircraft model. This and other aircraft data allows the simulation to distinguish among different aircraft and to assign them the appropriate separation rules, sequencing, speeds, takeoff and landing characteristics, and limitations based on size.

The simulation references an aircraft in three ways: model number, airspace group number, and airfield group number. Each reference is discussed below.

### **10.1 Aircraft Models**

SIMMOD uses an aircraft index number to identify the aircraft model of each flight. The numbers refer to aircraft model definitions in the Integrated Noise Model (INM) Data Base No. 9, a copy of which is supplied with SIMMOD. (The INM is an FAA model that determines aircraft noise impact at and around airports.) For example, the aircraft number 01 refers to the first aircraft described in the INM list, a B747-100/JT9DBD.

If the INM database is deemed unacceptable for a specific user’s application, the simulation will accept a different file with the same format.

The aircraft number is also used to define gate blockage. For each gate in the application, the analyst can define which adjacent gates will become blocked (unavailable for use) when the gate is occupied by an aircraft of any given number (see Gate blocking in Section 13, Airfield Logic).

The aircraft model descriptions in the INM database include takeoff weights for trips of various lengths. As shown in the chart below, the maximum INM aircraft weights generally indicate the size category of the individual aircraft models. Thus, aircraft may be assigned to size groups according to their weight category, as in the following table.

**Table 10-1: Aircraft Size Groups**

<b>INM max. weight</b>	<b>Aircraft Group</b>	<b>Group Number</b>
< 10,000 lb.	Single engine / GA	1
10,000 lb. to 100,000 lb.	Twin engine / Small	2
100,000 lb. to 300,000 lb.	Commercial jet / Large	3
> 300,000 lb.	Wide body and jumbo jets / Heavy	4

## 10.2 Aircraft Groups in Airspace

Many different models of aircraft have roughly equivalent characteristics when airborne. For the purposes of simulating airspace operations, aircraft are therefore classified into groups.

SIMMOD does not restrict the number or definition of aircraft groups defined for the airspace portion of the model, nor does it require that these groups match the aircraft groups defined for the airfield.

Aircraft in the airspace model are typically classified into five basic types: Large/heavy, Heavy, Large, Small, and GA (for General Aviation). The aircraft models belonging to each group must be listed in the data input so the simulation can determine the aircraft group to which each flight belongs and appropriately model the flight's airspace characteristics.

Characteristics defined for each group include:

- Speed ranges by link type (defined in Links and Link Types in Section 11, Airspace Structure)
- Holding queue by node type (defined in Nodes and Node Types in Section 11, Airspace Structure)
- Minimum separation list for this group (see below)
- Intrail separation multiplier
- Wake turbulence sequencing.

### Aircraft Separation in Airspace

The simulation maintains the separation between aircraft along links and passing through nodes. Intrail separation requirements for each aircraft of a given group followed by another group (the wake vortex separation requirements) can be defined uniquely, as shown in the following example.

**Table 10-2: Separation Requirements in Nautical Miles**

Aircraft Size	Followed by:			
	Heavy	Large	Small	GA
Heavy	4	5	6	6
Large	3	3	4	3
Small	3	3	3	3
GA	3	3	3	3

Random variation can be added to each intrail separation as a multiplier (in the form of a stochastic variable) to account for the realistic variation in actual aircraft separation. (See the entry on “Intrail Separation Multiplier” in Section 16, Stochastic Processes.)

### **Aircraft Groups on the Airfield**

Aircraft are also classified into groups for the purposes of simulating airfield operations. Aircraft with similar characteristics in ground operations are defined to belong to the same group. SIMMOD does not restrict the number and definition of airfield groups nor does it require that these groups match the aircraft groups defined for the airspace.

Aircraft have typically been classified into four basic groups: Heavy, Large, Small, and GA (for General Aviation). The aircraft models belonging to each of these airfield aircraft groups must be listed so that the simulation can appropriately determine the ground movement characteristics of each flight.

Characteristics defined for each group include:

- Landing characteristics (described under Landing and Takeoff Roll Distances in Section 16, Stochastic Processes)
- Takeoff characteristics (described under Landing and Takeoff Roll Distances in Section 16, Stochastic Processes)
- Gate occupancy characteristics (described under Gate Service Times for Arrivals and Departures in Section 16, Stochastic Processes).

## **11 Airspace Structure**

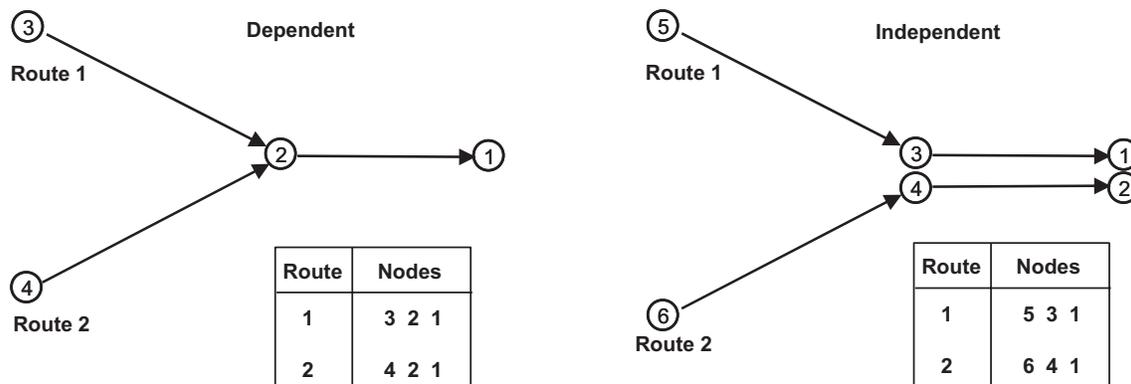
SIMMOD airspace is composed of an interrelated network of aircraft routes. These routes are defined by the analyst as a series of nodes and links. When two or more routes converge, some nodes and links will be held in common; that is to say, they will appear in the definition of more than one route.

All aircraft move in the airspace along these routes, and every flight entering the simulation must be assigned to a route in the input data. As an aircraft moves through the airspace, separation requirements are checked between it and other aircraft on the same path, merging paths, and crossing paths.

Unlike actual flights, aircraft in the simulation cannot deviate from their designated paths. This being the case, vertical and lateral separations are not checked by the simulation. These separation requirements are maintained insofar as routes are correctly defined by the user with vertical and lateral separation.

Each node on a path is given an altitude by definition; the simulation uses the altitude to calculate fuel consumption and speeds not given as true airspeed. Altitude is not checked or adjusted by the simulation to resolve conflicts.

Two links can occupy the same ground coordinates in the simulation but only at different altitudes. SIMMOD can handle these either as two independent paths that never affect each other or as two dependent paths that have all or part of a path in common. The decision to make paths dependent or independent is accomplished in the definition of the input data. See Figure 11-1, Dependent and Independent Routes.



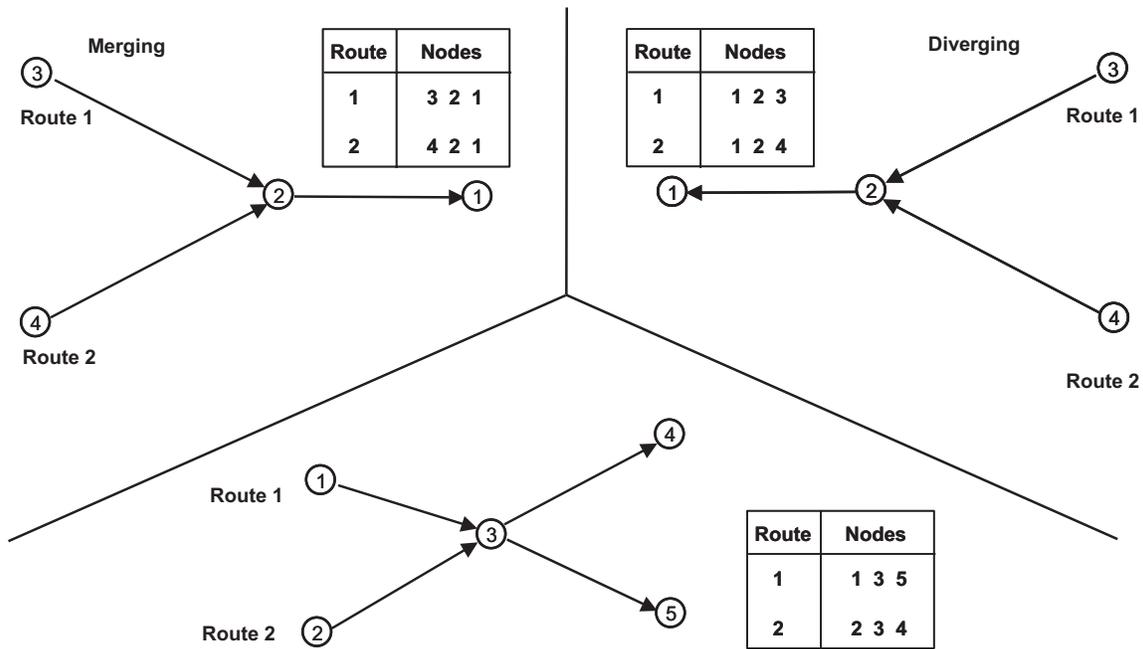
**Figure 11-1: Dependent and Independent Routes**

It is possible to make some changes in the structure of the airspace during the simulation. This is further explained in Section 15, Resetting Simulation Parameters.

## 11.1 Routes

A route is defined as a series of nodes connected by links listed sequentially in the direction of travel. A flight must be assigned to a route in the data input.

The interaction of routes is monitored by the links and nodes they hold in common. For example, two departure routes might share the first link and then diverge, or two routes crossing in the airspace might share a node. See Figure 11-2, Sample Routes: Crossing, Merging, and Diverging.



**Figure 11-2: Sample Routes: Crossing, Merging, and Diverging**

Airspace links and nodes restrict aircraft actions. That is, the node and link parameters are coordinated with each aircraft definition such that they affect the aircraft's performance and behavior. See Section 12, *Airspace Logic*, for additional details.

To simplify data requirements, links and nodes are grouped by type. Grouping nodes and links by type is not mandatory, but it does simplify data input and reduce simulation execution time by decreasing memory requirements. Links and nodes are usually grouped into types based on common data or data requirements.

### Nodes and Node Types

The definition of an airspace node determines several important control parameters for simulated aircraft passing through that node, including separation requirements, holding pattern characteristics, passing rules, and arrival and holding strategies. Additional information on these subjects is available in Section 10, *Aircraft Definition*, and Section 12, *Airspace Logic*.

### Holding

All holding in the airspace is done at airspace nodes. Flight patterns associated with holding aircraft are not explicitly modeled. Instead, SIMMOD models the effects of holding in terms of the time it uses, and applies holding to handle aircraft that are essentially waiting in a queue. The air route structure must therefore be defined by the user to accommodate any holding pattern implied by the holding queue characteristics at a node.

It is possible to define unique holding queue characteristics for each node. One such characteristic (which actually represents a set of characteristics) is the holding stack type. A node defined with a holding stack type allows each aircraft in the simulation to hold in a unique and appropriate manner.

Several sets of holding stack data, each representing a different stack type, can be defined for each aircraft group. During the simulation, each node defined with a holding stack type can thereby reference the corresponding holding stack data set for each aircraft.

For example, if a node is defined to use a holding stack of type 3, then each aircraft holding at that node uses the type 3 holding stack data for its group.

The definition of an airspace node must indicate which holding stack type it will employ, if any. The decision to use a certain holding stack type should be based on the node's position and function in the airspace model.

To simplify data input, it is convenient to group airspace nodes with the same holding queue characteristics as the same type of node.

If the definition of a node does not specify a particular type of holding stack, holding aircraft simply remain at the node until released — first in, first out — to proceed to the next node.

There is no minimum hold period in this case.

### **Holding Stack Characteristics**

Holding stack characteristics define the time restrictions imposed upon an aircraft waiting to leave a hold pattern at a node. The restrictions define two values: (1) the minimum holding time and (2) the holding pattern exit time interval. The latter time increment determines the intervals at which aircraft can exit the pattern.

For example, an aircraft at a node may be required to hold at least 5 minutes. It may exit at that time (five minutes) or at two-minute intervals thereafter, i.e., at seven, nine, eleven, etc. minutes. The holding stack type also determines the speed of the aircraft as it holds in a stack.

For additional information, see the entry on Airspace Holding in Section 12, Airspace Logic.

### **Interface Nodes**

Some nodes in the airspace are defined as interface nodes. Interface nodes indicate the transition between the ground and air simulation. They are typically near the end of a runway. If a node in an airspace arrival route is an interface node, the simulation will continue to handle the flight in the airport model or in a degenerate airport.

The first node of an airspace departure route is an interface node. After takeoff, the simulation continues a flight from that node. A complete description of the logic of interface nodes is provided in Section 14, Interface Logic.

### **Links and Link Types**

Aircraft move from one node to the next only along a defined link. Airspace links typically represent segments of a flight path. Routes are usually comprised of several links.

Links can be grouped by location to simulate airspace sectors. SIMMOD can monitor these sectors for occupancy and control them based on sector capacity.

To represent a set of path segments affected by the same winds, links can also be grouped (e.g., by altitude) into Wind Sets.

A link type defines a group of links with the same speed characteristics. The only data common to defined link types are the specified maximum, minimum, and nominal speeds an aircraft can travel along each link.

### **Link Speed Ranges**

Speed range is defined by a maximum speed, minimum speed and nominal speed. The nominal speed is the normal speed an aircraft would travel without any adjustments for conflict or congestion. The maximum speed is the upper bound and the minimum speed is the lower bound used to adjust aircraft speeds to resolve conflict or congestion. Depending on the speed

adjustment strategy in effect, any speed between the minimum and maximum may be used by the model. For a complete explanation of the speed adjustment, see Section 12, Airspace Logic. The simulation's standard measure of speed is true airspeed in knots. If speeds are entered as true airspeed, the simulation will perform no conversions. Speeds can also be input using indicated airspeed in knots or Mach number as the unit of measure. The simulation will convert these to true airspeed.

The conversion of indicated airspeed to true airspeed for a link is done by calculating the true airspeed at the altitude of each node and then averaging. The calculations performed by the simulation to determine true airspeed are given below.

The conversion from indicated airspeed to true airspeed is as follows:

Known values:

$V_{ind}$  =indicated airspeed in knots

$A$  =aircraft altitude in feet

Unknown values to be determined:

$V_{true}$  =true airspeed in knots

Sub-expressions:

$$\begin{aligned} \varepsilon &= 5 \text{ (speed of sound in knots at sea level at standard temperature and pressure)}^2 \\ &= 5 (661.4786)^2 \end{aligned}$$

$$\theta = 1 - 6.8755856 \times 10^{-6} A \text{ temperature ratio using the standard lapse rate}$$

For altitudes less than or equal to 36,089 feet (tropopause):

$$V_{true} = \sqrt{\varepsilon \theta \left( \left( \frac{\left( 1 + \frac{V_{ind}^2}{\varepsilon} \right)^{3.5} - 1}{\theta^{5.2558797}} + 1 \right)^{\frac{1}{3.5}} - 1 \right)}$$

For altitudes greater than 36,089 feet:

$$V_{true} = \sqrt{\varepsilon \theta \left( \left( \frac{\left( 1 + \frac{V_{ind}^2}{\varepsilon \theta} \right)^{3.5} - 1}{0.2234 e^{\left( \frac{36089 - A}{20805} \right)}} + 1 \right)^{\frac{1}{3.5}} - 1 \right)}$$

The conversion of Mach number to true airspeed for a link is done by calculating the true airspeed at the altitude of each node, and then averaging the two values. The calculations performed by the simulation to determine true airspeed are given below. The speed of sound is a function of temperature and pressure, which vary with ephemeral atmospheric conditions. SIMMOD assumes 59°F at sea level as standard.

For altitude less than or equal to 36,089 feet:

$$\text{Speed of Sound} = 290555 \sqrt{51867 - (0.003566 \text{ Altitude})}$$

For altitude greater than 36,089 feet:

$$\text{Speed of Sound} = 57385$$

For true airspeed:

$$V_{true} = \text{Speed of Sound} \times \text{Mach Number}$$

## 11.2 Wind Sets

The effects of wind can vary by link. Links can be grouped together such that they have the same wind effects. These groups are called wind sets. Wind sets are defined by data input. Typical examples of wind sets include:

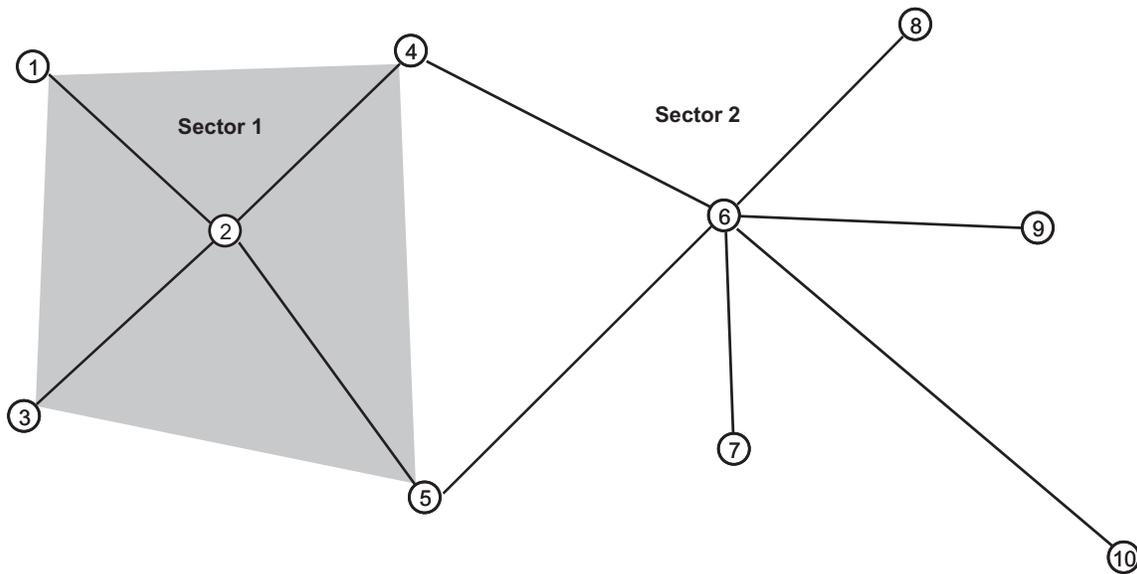
- High vs. low altitude links
- Groups linked by physical location
- Terminal approach vs. en route links.

If no windsets are defined by data input, all links are grouped into one windset. If some windsets are defined, the links not included in a windset are grouped together in a windset appearing at the end of the wind set list.

The simulation will consider the effects of wind in all speed calculations, including those which yield travel time and fuel consumption figures. The wind data includes speed in knots and direction.

## 11.3 Sectors

The simulation can measure the combined capacity of a group of links defined as a sector. The sector capacity includes the total number of aircraft (a) on links included in the sector or (b) holding at nodes included within the sector or (c) holding at nodes at the defined perimeter of the sector if the aircraft is exiting from the sector. If an aircraft is entering the sector and holding at a node at the defined perimeter of the sector, then it is still considered to be in the previous sector. Holding will occur at the nodes on the edge of a sector if it is full. See Figure 11-3, Sectors.



**Figure 11-3: Sectors**

Airspace sectors can be explicitly defined by data input. If no sectors are defined, all links are grouped into one sector. If some sectors are defined, the links not included in a sector are grouped together in a sector at the end of the sector list.

## 11.4 Plans

The definition of routes is more complex than a simple node and link list. A group of routes can be defined to handle different operations for an airport.

An airport might be operating with a southern flow and then, because of changing conditions or operational requirements, change to a northern flow. The simulation considers each type of operation a plan. Under a plan certain routes are available for use. If a plan changes, a different set of routes will become available.

If no plan information is defined by input the simulation considers all routes available in plan 1. For an explanation of the simulation logic involved in plan changes see Section 15, Resetting Simulation Parameters.

## 12 Airspace Logic

SIMMOD's airspace model logic manages the simultaneous movement of aircraft on all airspace routes. As noted earlier in this manual, routes are defined as a series of nodes connected by links. Airspace movement takes place along these links. As each aircraft traverses a link, it is required to maintain minimum separation from preceding and succeeding aircraft unless the link is defined to allow passing. Other considerations that restrict aircraft progress on a route include the capacity of a link or a sector, projections of downstream congestion, and aircraft movement controls on the links.

The simulation evaluates each aircraft's position with respect to other aircraft in the system while the aircraft is at a node. Based on data input, SIMMOD resolves air traffic control decisions (e.g., whether to allow another aircraft on a link, what intrail separation the aircraft will maintain) before each aircraft is allowed to enter the link. This section first considers some fundamental rules for aircraft movement on airspace links. It then addresses aircraft control at airspace nodes (holding and holding strategies), and three aircraft movement control strategy levels and their ramifications.

### 12.1 Aircraft Movement Rules on the Links

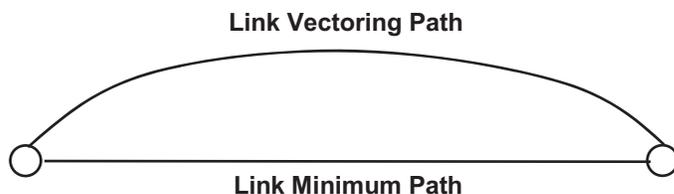
Links may be defined to model certain circumstances, opportunities, and limitations that an aircraft would encounter in airspace. The definition of a link may change the effects of an aircraft's control strategy.

#### Link Types and Aircraft Speeds

A link type defines a group of links with the same aircraft speed ranges. For the purposes of simulation, SIMMOD must determine the length of time each aircraft will use to traverse each link. This length of time is used to assign the flight a time of arrival (TOA) at the next node. The length of time is determined using three speed parameters: maximum speed, minimum speed, and nominal speed. These speeds are defined for each model of aircraft traveling on each type of link. The preferred speed is the nominal speed. The aircraft will always use the nominal speed if it is possible to do so. If it is not, the aircraft will use a speed between the maximum and the nominal, or between the nominal and the minimum, depending on the aircraft movement control strategy in effect.

## Link Time for Vectoring or Path Stretching

An aircraft may have to use up more time traversing a link than it can by merely traveling at the minimum speed. SIMMOD therefore allows the user to define for each link an amount of time that may be used in vectoring or path stretching. SIMMOD makes no distinction between vectoring and path stretching. The simulation decides to delay the flight based on time requirements. In effect, SIMMOD adds to the link distance by adding an appropriate amount of vectoring time. This extra time represents the vector distance added to the link traveled at the minimum speed. Since the vectoring time is defined by link and not by aircraft type, vectoring time should approximate a realistic delay for all aircraft types flying a link. See Figure 12-1, Link Vectoring or Path Stretching.



### Example:

The minimum length of the link is the straight line distance.

A link incorporating delay from vectoring or path stretching implies a curved path.

Maximum additional (vectoring) distance = Maximum vectoring delay time x Minimum speed

Figure 12-1: Link Vectoring or Path Stretching

## Wind

The effect of wind in airspace is considered in the calculation of an aircraft's TOA at a node. Every speed calculation is adjusted for wind effects. The wind calculation is:

$$\text{Headwind} = \cos(\text{Wind Heading} - \text{Average Link Heading}) \times \text{Wind Speed}$$

$$\text{CrabFactor} = 1 - \frac{\text{WindSpeed}^2 - \text{Headwind}^2}{V_{\text{true}}^2}$$

$$\text{GroundSpeed} = (V_{\text{true}} \times \text{CrabFactor}) - \text{Headwind}$$

This calculation of ground speed is an approximation that is accurate in situations where the wind speed is significantly less than the aircraft's true airspeed.

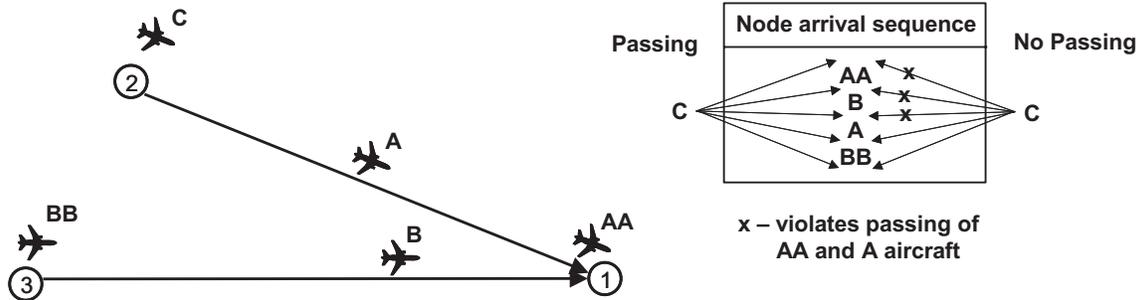
Setting the wind for the simulation is further explained in the Wind Sets entry of the previous section, and in Section 15, Resetting Simulation Parameters, under the entry Changing the Wind Characteristics.

## Passing and Wake Turbulence Along a Link

The link overtake setting allows aircraft to pass one another along a link. If passing is allowed on a link, an aircraft's position in the arrival queue sequence at the next node is not restricted by the TOA's of aircraft preceding it on the link. See Figure 12-2, Passing and the Node Arrival Queue.

The feasibility of any aircraft position in the next node's arrival queue is determined by the node's arrival control strategy and the aircraft's performance characteristics. For example, if the

next node applies a strict First-In-First-Out arrival queue strategy (SIMMOD’s “QFIFO” strategy) to aircraft arriving from a link, an aircraft traveling on that link will not be able to change its position in the node arrival queue, even though it technically has the ability to pass. On the other hand, if passing is not allowed on a link, and if there are no other links leading to the next node, then the only position available to an aircraft entering the node arrival queue is behind the preceding aircraft on the same link, i.e., in QFIFO order.



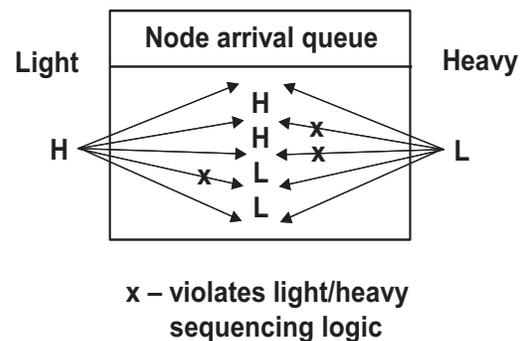
**Figure 12-2: Passing and the Node Arrival Queue**

In such circumstances, aircraft must therefore be allowed to pass in order to exploit the SIMMOD aircraft movement strategies beyond QFIFO. These strategies will be explained in full detail further below.

When aircraft are allowed to pass on a link, the simulation does not enforce the intrail separation requirements that would otherwise protect light aircraft from wake turbulence. However, the user can prevent a light aircraft from directly following a heavy aircraft on the same link by setting the wake turbulence flag. This light/heavy sequencing protocol essentially restricts the positions available to an aircraft entering the next node’s arrival queue. See Figure 12-3, Light/Heavy sequencing.

A light aircraft in the final position in the queue constitutes an exception to this protocol, because it must always be possible for an aircraft to be last in the queue.

The wake turbulence flag inhibits the option of vectoring on a link where wake turbulence (light/heavy sequencing) is in effect. Since the model does not precisely track the paths aircraft use while vectoring or path stretching, it would be impossible to protect them from wake turbulence.



**Figure 12-3: Light/Heavy Sequencing**

## 12.2 Aircraft Movement Control at the Nodes

Nodes are the gateways to the links and control points along the routes. An aircraft appears at an airspace node because it is a flight entering the simulation at this node or because it is arriving from a previous airspace or airfield node. Upon arrival at a node an aircraft has two options: either it is cleared to pass through the node or it is held at the node. To clear an aircraft, the simulation checks for the following conditions:

- Other aircraft holding at the current node
- Holding strategy at the next node

- Capacity of the next link
- Capacity of the sector (if entering a new sector at the current node)

The simulation maintains and references two aircraft queues at every node: (1) the holding queue, which lists aircraft holding at that node, and (2) the node arrival queue, which lists the aircraft approaching that node. The simulation refers to these queues in making the traffic projections that determine an aircraft's action.

If all conditions are favorable, the aircraft is released from the node to travel the link at a specified speed. The aircraft's TOA is entered in the arrival queue of the next node along the route and this event is added to the event schedule for the aircraft.

Each aircraft in the node arrival queue is listed by its TOA. This time of arrival is determined by the effects of the node's arrival control strategies and can be changed dynamically. Conditions may also force an aircraft to hold at the node. Depending on how the user defines the particular SIMMOD application, the simulation may not be able to exercise any other option. If the holding at a node seems unrealistic, the user may choose to manage holding earlier along the route or to apply different aircraft movement constraints. In most cases, users wish to facilitate basic aircraft movement in the airspace.

### **Airspace Holding**

The simplest aircraft movement control exerted at a node is holding. The initial decision to hold is made by determining if the release of an aircraft would contribute to congestion (indicated by holding) at the next node. Holding at a node is considered the last resort of the simulation. It is an option at a node if forward movement is restricted. Aircraft holding at a node are always queued to exit in first-in-first-out order. Each aircraft has a projected exit time from the queue, and aircraft cannot pass one another while holding. Holding can be further specified by defining holding stacks. Holding stacks define the length of time an aircraft will be held if holding is required. At the exit time, the simulation checks conditions to determine if it can safely release the aircraft from the node.

### **Airspace Holding Stacks Characteristics**

The holding stack characteristics define the time restrictions imposed upon an aircraft waiting to leave a hold pattern at a node. The restrictions define two values: (1) the minimum holding time and (2) the holding pattern exit time interval. The latter time increment determines the intervals at which aircraft can exit the pattern. For example, an aircraft at a node may be required to hold at least 5 minutes. It may exit at that time (five minutes) or at two-minute intervals thereafter, i.e., at seven, nine, eleven, etc. minutes.

A holding stack defines the minimum time an aircraft must spend in a holding stack and the time intervals after the minimum when the aircraft can leave. Suppose an aircraft enters a holding stack at a node. The holding stack may be defined to hold the aircraft a minimum of 2 minutes, and to release it thereafter only at 30 second intervals. If conditions still require the aircraft to hold at the end of these 2 minutes, this holding stack will allow the aircraft to exit as soon as conditions are favorable at one of the 30 second intervals thereafter, e.g., at 2 minutes and 30 seconds, 3 minutes, 3 minutes and 30 seconds, etc.

### **Airspace Holding Strategy**

To help control holding at downstream nodes, a holding strategy may be used at each node. The holding strategy is invoked at each node based on that node's aircraft movement control strategy,

the aircraft holding at the next node and the capacity of the next node. The three holding strategies are listed below in order of their increasing complexity. Depending on the strategy in effect at its current node, an aircraft holds if:

- Strategy 1 - There is an aircraft holding at the next node on the route
- Strategy 2 - The capacity of the next node's holding queue is full.
- Strategy 3 - The holding capacity of the next node is exceeded by the number of aircraft currently holding at the next node plus the number of aircraft approaching it.

All three strategies require that holding exist at the next node before any checking is done on the capacity or content of that node's holding queue. The first strategy is the default used at any node without a specified strategy. This is the simplest check to determine if holding is occurring at the next node. See Figure 12-4, Holding Strategy 1.

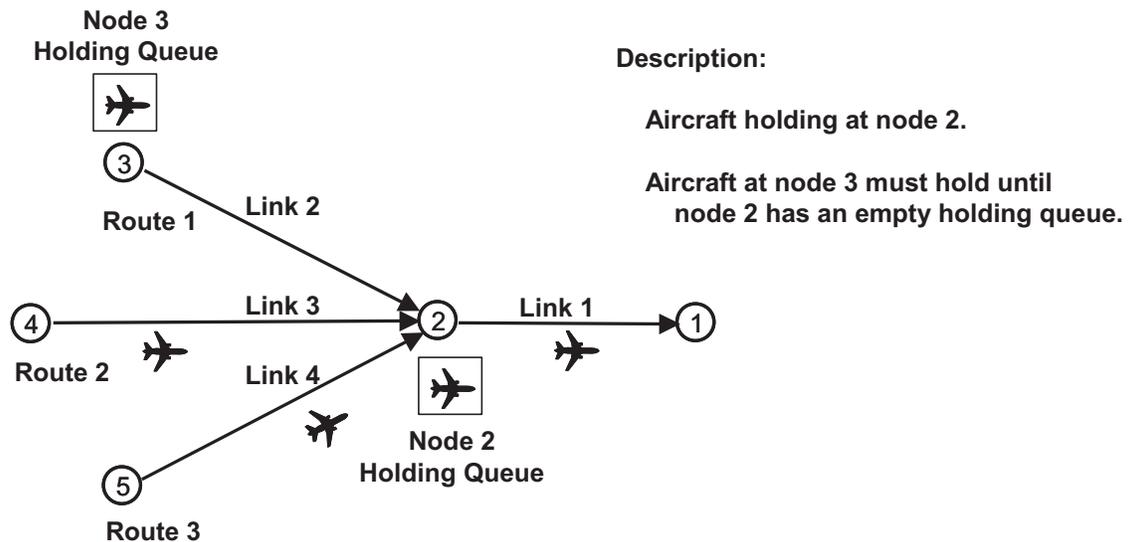


Figure 12-4: Holding Strategy 1

The second strategy takes the check a step further. If there is holding, then it determines whether the holding queue at the next node is full. See Figure 12-5, Holding Strategy 2.

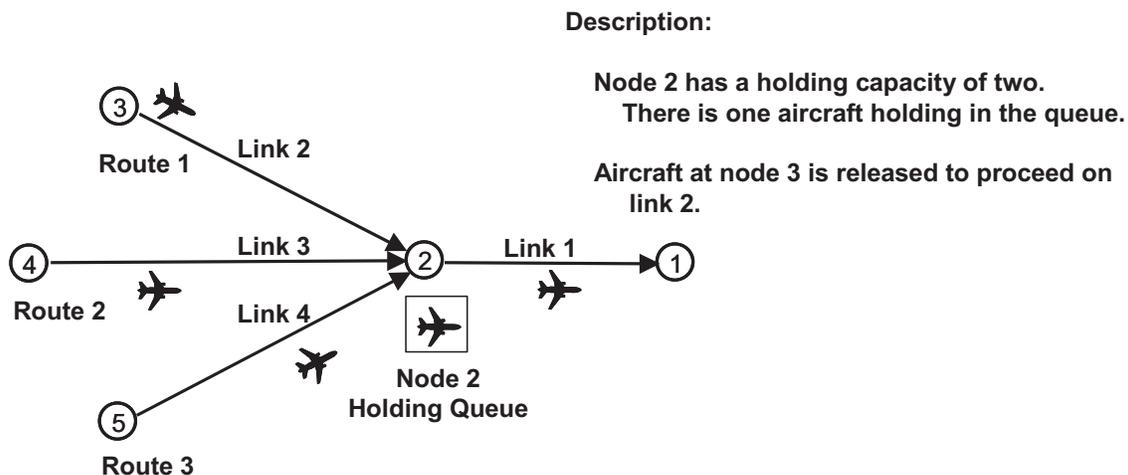


Figure 12-5: Holding Strategy 2

The third strategy, in addition to checking the number already holding at the next queue, considers the number approaching the node to see if the next holding queue is scheduled to be full by the time the aircraft would arrive. See Figure 12-6, Holding Strategy 3.

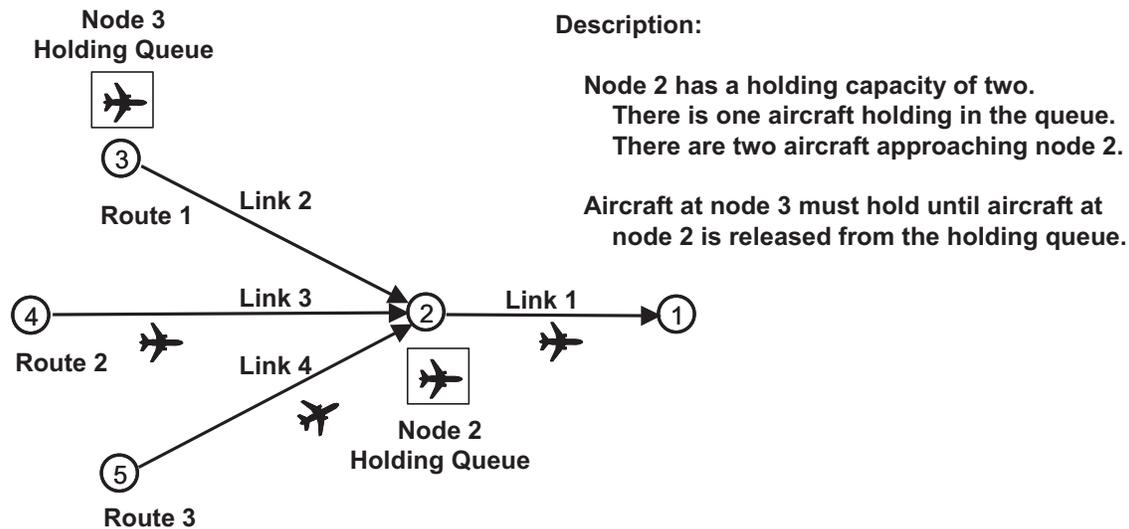


Figure 12-6: Holding Strategy 3

### 12.3 Aircraft Movement Control Strategies

There are three strategic levels of aircraft movement control. A movement control strategy is a logical approach to controlling aircraft traffic in airspace. Each different strategy or mix of strategies can have different effects on aircraft movement, and so may be applied to resolve different kinds of conflicts.

Control strategies will only be applied when an aircraft is ready to enter a new link, and this can only occur when the link has unfilled capacity.

The control strategies are as follows:

- Level I Node Arrival Control
  - Type 1: QFIFO
  - Type 2: SpeedFit
  - Type 3: MultiFit
- Level II Metering Control
  - Type 1: Basic
- Level III Flow Control
  - Type 1: Basic

Generally speaking, the higher numbered, more complex strategy levels use more sophisticated logic and require more data input. They are usually not needed at all nodes and will increase run time considerably if applied universally.

The Level I strategy is referred to as Node Arrival Control. Note, however, that all strategies involve node arrival decisions to some extent. This level of strategy includes three alternate strategy types. Each node can have a different type of Level I strategy, but one must be defined for each node.

Any Level I strategy will work in unison with the Level II and Level III strategies. Levels II and III include only one type of strategy each, and these are optional.

## 12.4 Level I: Node Arrival Control Strategies

### Type 1: QFIFO Node Arrival Control

The first node arrival control strategy, QFIFO, is the simplest. Called QFIFO to indicate that the first into the queue is the first out, this type of control is the default for the simulation and should be used unless more control is specifically required.

The QFIFO control is straightforward. Every aircraft approaching a node is always put at the end of the node arrival queue. This can back up the air traffic. An aircraft may be required to wait at its current node until proper separation can be achieved in relation to aircraft preceding it on the next link, or in relation to aircraft preceding it in the node arrival queue (e.g., aircraft arriving from other converging links).

QFIFO works best for nodes with a single link approaching them or in cases where passing is not allowed on a link. The QFIFO Logic searches for the last aircraft currently in the node arrival queue and reads its TOA.

To enter the node arrival queue, the next aircraft must have a later TOA and the difference in TOA's must be sufficiently large to ensure that minimum separation is maintained by the two aircraft.

The entering aircraft's TOA is calculated using the nominal speed. See Figure 12-7, QFIFO Logic with No Delay.

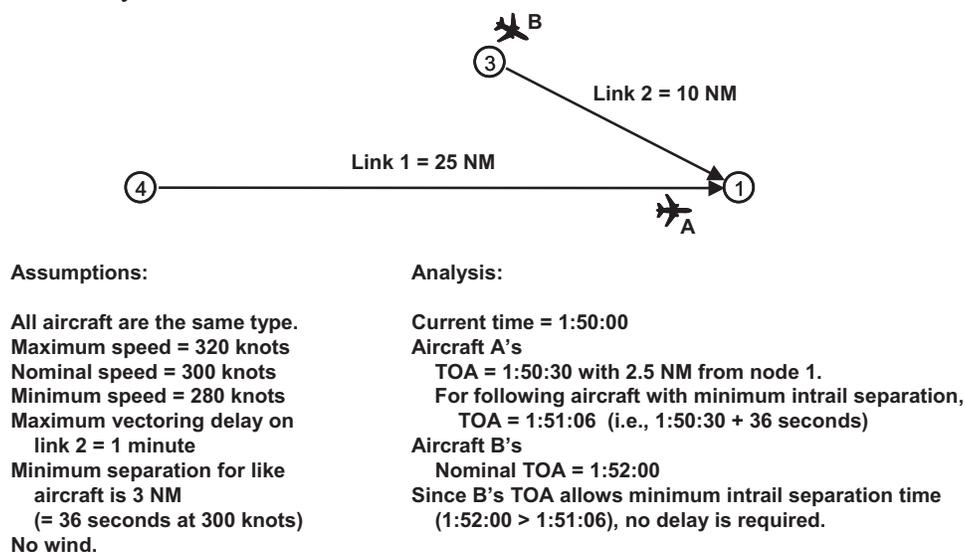
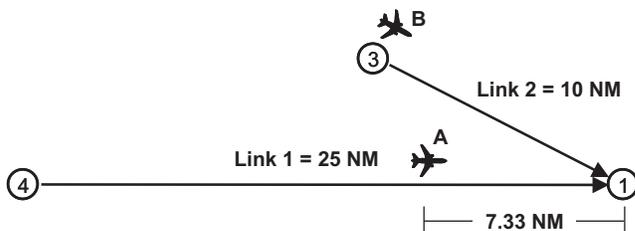


Figure 12-7: QFIFO Logic with No Delay

If this nominal time is still too early, the simulation delays (i.e., contrives to add time to) the TOA, as described below.

### Order of Delays Generated for QFIFO

The QFIFO strategy first creates delay by slowing down the aircraft so that it flies the link between the nominal speed and the minimum speed. The delay is the difference between the nominal time and the time required to fly the link at the lesser speed. See Figure 12-8, QFIFO Logic with Speed Delay.



**Assumptions:**

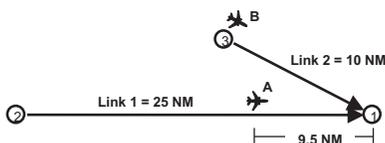
All aircraft are the same type.  
 Maximum speed = 320 knots  
 Nominal speed = 300 knots  
 Minimum speed = 280 knots  
 Maximum vectoring delay on link 2 = 1 minute  
 Minimum separation for like aircraft is 3 NM  
 (= 36 seconds at 300 knots)  
 No wind.

**Analysis:**

Current time = 1:50:00  
 Aircraft A's  
 TOA = 1:51:28 with 7.33 NM from node 1.  
 For following aircraft with minimum intrail separation,  
 TOA = 1:52:04.  
 Aircraft B's  
 TOA at nominal speed = 1:52:00  
 TOA at minimum speed = 1:52:09  
 Since B's minimum speed TOA is greater than  
 the minimum intrail separation time (1:52:09 > 1:52:04),  
 the delay can be absorbed with a speed decrease.  
 Aircraft B's TOA = 1:52:04 at a speed of 290 knots.

**Figure 12-8: QFIFO Logic with Speed Delay**

If this delay is not sufficient to yield adequate separation, the aircraft will fly the link at the minimum speed and attempt to create the additional delay by vectoring. The vectoring delay includes the delay from traveling at the minimum speed and the additional delay from vectoring (i.e., path stretching) on the link. See Figure 12-9, QFIFO Logic with Vectoring Delay.



**Assumptions:**

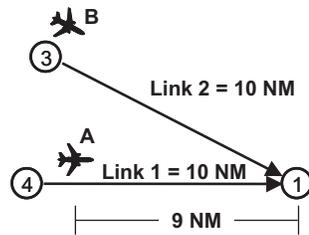
All aircraft are the same type.  
 Maximum speed = 320 knots  
 Nominal speed = 300 knots  
 Minimum speed = 280 knots  
 Maximum vectoring delay on link 2 = 1 minute  
 Minimum separation for like aircraft is 3 NM  
 (= 36 seconds at 300 knots)  
 No wind.

**Analysis:**

Current time = 1:50:00  
 Aircraft A's  
 TOA = 1:51:54 with 9.5 NM from node 1.  
 For following aircraft with minimum intrail separation,  
 TOA = 1:52:30.  
 Aircraft B's  
 TOA at nominal speed = 1:52:00  
 TOA at minimum speed = 1:52:09  
 TOA at minimum speed with vectoring = 1:53:09  
 Since B's minimum speed TOA is less than the minimum  
 intrail separation time (1:52:09 < 1:52:30),  
 the delay cannot be absorbed with a speed decrease.  
 Since B's vectored TOA is greater than the minimum  
 intrail separation time (1:53:09 > 1:52:30),  
 the delay can be absorbed with a speed decrease and  
 vector delay.  
 Aircraft B's TOA = 1:52:30 at a speed of 280 knots and 21  
 seconds of vectoring.

**Figure 12-9: QFIFO Logic with Vectoring Delay**

If this is still not sufficient, the aircraft will hold at the current node until it can traverse the link at the minimum speed with the vectoring delay. See Figure 12-10, QFIFO Logic with Holding Delay.



**Assumptions:**

All aircraft are the same type.  
 Maximum speed = 320 knots  
 Nominal speed = 300 knots  
 Minimum speed = 280 knots  
 No vectoring.  
 Minimum separation for like aircraft is 3 NM  
 (= 36 seconds at 300 knots)  
 No wind.

**Analysis:**

Current time = 1:50:00

**Aircraft A's**

TOA = 1:51:48 with 9 NM from node 1.

For following aircraft with minimum intrail separation,  
 TOA = 1:52:24.

**Aircraft B's**

TOA at nominal speed = 1:52:00

TOA at minimum speed = 1:52:09

Since vectoring is not allowed and the delay cannot be absorbed with a speed decrease, B must hold at node 3.

Aircraft B's TOA = 1:52:24 at a speed of 280 knots and 15 seconds of holding delay.

**Figure 12-10: QFIFO Logic with Holding Delay**

Using QFIFO thus involves the following restrictions:

- Each aircraft must maintain separation from the aircraft before it in the queue.
- Aircraft must not violate the minimum or maximum speeds.
- Aircraft TOA's can include the vectoring delay only if light/heavy sequencing is not in effect. (Since each new aircraft will be assigned to the last position in the queue, light/heavy sequencing provides no additional control and should not be used where QFIFO is in effect.)

QFIFO uses the following order of steps to fit an aircraft in the last position in a queue:

1. Try to make it fit at nominal speed.
2. Try to make it fit at decreased speed.
3. Try to make it fit at decreased speed using vectoring (if allowed).
4. Make it fit at decreased speed using vectoring (if allowed) and holding delay.

## 12.5 Level I: Node Arrival Control Strategies, Continued

### Type 2: SpeedFit Node Arrival Control

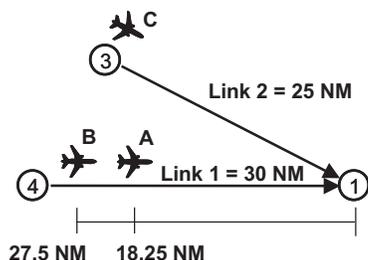
The second node arrival control strategy is called SpeedFit because any aircraft entering the node arrival queue can adjust its speed (within its allowable speed range) to fit into any position in the queue that allows adequate separation between the preceding and succeeding aircraft. SpeedFit only changes the speed of the aircraft entering the queue. It cannot change the speed or TOA of any other aircraft in the node arrival queue.

SpeedFit offers more possibilities and yields major improvements if certain conditions exist, e.g., if aircraft are approaching a node from more than one link, or if there is a mix of approaching aircraft with significantly large speed differences and sufficiently wide speed ranges to allow passing to occur.

In the SpeedFit logic, an aircraft's initial position in the queue is projected based on nominal speed. A check is done to see if this aircraft's "nominal" position violates any requirements set up for the airspace system. Such violations could include, for example: lack of separation with

an aircraft already in the queue; passing an aircraft on a link where passing is not allowed; or sequencing a light behind a heavy aircraft.

The simulation can attempt to fit the aircraft into this nominal position by adjusting the aircraft's speed (within its range) and by applying path stretching (if this is allowed). See 12-11, SpeedFit at Nominal Position with Speed Increase.



**Assumptions:**

- All aircraft are the same type.
- Maximum speed = 320 knots
- Nominal speed = 300 knots
- Minimum speed = 280 knots
- Maximum vectoring delay on link 2 = 1 minute
- Minimum separation for like aircraft is 3 NM (= 36 seconds at 300 knots)
- No wind.

**Analysis:**

Current time = 1:50:00

**Aircraft A's**

TOA = 1:53:39 with 18.25 NM from node 1.

For following aircraft with minimum intrail separation,

TOA = 1:54:15.

**Aircraft B's**

TOA = 1:55:30 with 27.5 NM from node 1.

For following aircraft with minimum intrail separation,

TOA = 1:56:06.

**Aircraft C's**

TOA at nominal speed = 1:55:00

For following aircraft with minimum intrail separation,

TOA at nominal speed = 1:55:36.

At the nominal speed, B does not have separation with C.

Increasing C's speed to 306 knots.

**Aircraft C's**

TOA = 1:54:54

For following aircraft with minimum intrail separation,

TOA at nominal speed = 1:55:30.

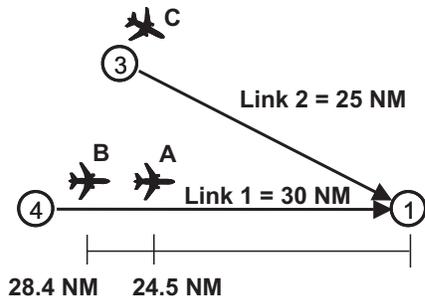
Can fit between A and B.

**Figure 12-11: SpeedFit at Nominal Position with Speed Increase**

If the initial projection of an aircraft's nominal position is not feasible, the simulation will try to find a fit forward in the queue, starting with the next position - unless passing is a violation, in which case all forward positions would constitute a passing violation. See Figure 12-12, SpeedFit at Forward Position with Speed Increase.

NOTE: A "feasible" position in this context consists of a ten second window in addition to the time guaranteeing the separation requirement. Calculations appearing in the section figures do not, however, take this requirement for a ten second window into account. This omission simplifies the explanation of SpeedFit and MultiFit node arrival control strategies and the processes they entail.

If no forward positions are viable, positions behind the nominal are tried. See Figure 12-13, SpeedFit at Nominal Position with Speed Decrease and Figure 12-14, SpeedFit at Backward Position with Speed Decrease, Vectoring and Holding Delay. As a final resort, the last position is always available.



**Assumptions:**

All aircraft are the same type.  
 Maximum speed = 350 knots  
 Nominal speed = 300 knots  
 Minimum speed = 280 knots  
 Maximum vectoring delay on link 2 = 1 minute  
 Minimum separation for like aircraft is 3 NM  
 (= 36 seconds at 300 knots)  
 No wind.

**Analysis:**

Current time = 1:50:00

**Aircraft A's**

TOA = 1:54:54 with 24.5 NM from node 1.  
 For following aircraft with minimum intrail separation,  
 TOA = 1:55:30.  
 For preceding aircraft with minimum intrail separation,  
 TOA = 1:54:18.

**Aircraft B's**

TOA = 1:55:41 with 28.4 NM from node 1.  
 For following aircraft with minimum intrail separation,  
 TOA = 1:56:17.

**Aircraft C's**

TOA at nominal speed = 1:55:00  
 For following aircraft with minimum intrail separation,  
 TOA at nominal speed = 1:55:36.

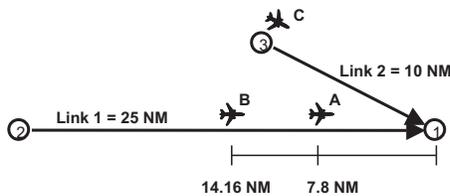
C does not fit between A and B.

Increase speed of C to fit forward position (before A).  
 Increasing C's speed to 349 knots.

**Aircraft C's**

TOA = 1:54:18  
 For following aircraft with minimum intrail separation,  
 TOA = 1:54:54.

Figure 12-12: SpeedFit at Forward Position with Speed Increase



**Assumptions:**

All aircraft are the same type.  
 Maximum speed = 320 knots  
 Nominal speed = 300 knots  
 Minimum speed = 280 knots  
 Maximum vectoring delay on link 2 = 1 minute  
 Minimum separation for like aircraft is 3 NM  
 (= 36 seconds at 300 knots)  
 No wind.

**Analysis:**

Current time = 1:50:00

**Aircraft A's**

TOA = 1:51:34 with 7.8 NM from node 1.  
 For following aircraft with minimum intrail separation,  
 TOA = 1:52:10.

**Aircraft B's**

TOA = 1:52:50 with 14.16 NM from node 1.  
 For following aircraft with minimum intrail separation,  
 TOA = 1:53:26.

**Aircraft C's**

TOA at nominal speed = 1:52:00  
 For following aircraft with minimum intrail separation,  
 TOA = 1:52:36.

At the nominal speed, C does not have separation with A.

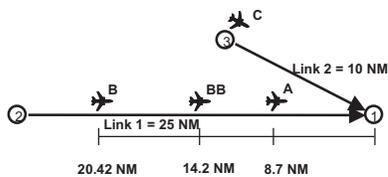
Decreasing C's speed to 280 knots creates some of the needed delay (separation).

TOA at minimum speed = 1:52:09

Need 1 second of vector delay.

Aircraft C's = 1:52:10 at a speed of 280 knots and 1 second of vectoring.

Figure 12-13: SpeedFit at Nominal Position with Speed Decrease



**Assumptions:**

All aircraft are the same type.  
 Maximum speed = 320 knots  
 Nominal speed = 300 knots  
 Minimum speed = 280 knots  
 Maximum vectoring delay on link 2 = 1 minute  
 Minimum separation for like aircraft is 3 NM  
 (= 36 seconds at 300 knots)  
 No wind.

**Analysis:**

Current time = 1:50:00

**Aircraft A's**

TOA = 1:51:44 with 8.7 NM from node 1.  
 For following aircraft with minimum intrail separation,  
 TOA = 1:52:20.

**Aircraft BB's**

TOA = 1:52:50 with 14.2 NM from node 1.  
 For following aircraft with minimum intrail separation,  
 TOA = 1:53:26.

**Aircraft B's**

TOA = 1:54:05 with 20.42 NM from node 1.  
 For following aircraft with minimum intrail separation,  
 TOA = 1:54:41.

**Aircraft C's**

TOA at nominal speed = 1:52:00  
 Nominal speed position has no fit between A and BB.  
 Earliest TOA with speed increase = 1:51:53  
 Speed increase is not enough to change position forwards.  
 Latest TOA with speed decrease = 1:52:09  
 Speed decrease is not enough to change position backwards.  
 Latest TOA with speed decrease and vector delay = 1:53:09  
 Speed decrease with vectoring offers a position change,  
 but position does not have separation with BB.  
 Add 18 seconds holding to speed decrease (280 knots) and  
 vector delay (1 minute) to yield TOA = 1:53:27.

**Figure 12-14: SpeedFit at Backward Position with Speed Decrease, Vectoring, and Holding Delay**

SIMMOD always adjusts the nominal speed by the minimum amount required to meet the separation requirements. If there is room for an aircraft at a position in the node arrival queue, the simulation will choose a speed closest to the nominal speed. If the position is forward in the queue, the speed will be the slowest speed allowing separation from the aircraft behind. Otherwise, the speed is the fastest speed allowing separation from the aircraft in front. Thus, each aircraft to be placed into a queue position by SpeedFit is subject to the following restrictions:

- Only the aircraft entering the queue can be adjusted to make a fit.
- Each aircraft must maintain separation from the aircraft before and after it in the queue.
- Aircraft must not violate the minimum or maximum speeds to achieve position.
- Aircraft position can include the vectoring delay where light/heavy sequencing is not in effect.
- Aircraft positions after the nominal position can include holding delay.

SpeedFit uses the following order of steps to fit an aircraft in a queue:

1. Try to fit in the nominal position in the queue:
  - Use nominal speed
  - Increase speed
  - Decrease speed
  - Decrease speed and apply vectoring (if allowed)
2. Try to fit in a position forward from the nominal in the queue:
  - Increase speed
  - Try to fit in a position back from the nominal in the queue:
    - Decrease speed
    - Decrease speed and apply vectoring (if allowed)
    - Decrease speed and apply vectoring (if allowed) and holding delay

3. Fit in the last position in the queue (always available):
  - Decrease speed
  - Decrease speed and apply vectoring (if allowed)
  - Decrease speed and apply vectoring (if allowed) and holding delay

## 12.6 Level I: Node Arrival Control Strategies, Continued

### Type 3: MultiFit Node Arrival Control

The third node arrival control strategy, MultiFit, takes the SpeedFit control a step further. MultiFit attempts to fit an aircraft at each position by adjusting other aircraft in the queue. To try to make a fit, MultiFit adjusts not only the speed of the individual aircraft entering the node arrival queue, but also the speeds of aircraft preceding and succeeding it in the queue.

If no fit is found at a position, the preceding and succeeding aircraft speeds are returned to their original values and the next position is tried. First the nominal position is tried for a fit, then the positions forward from nominal, and finally backwards from nominal. When attempting to fit forward or backward, positions are attempted one by one, starting with the position closest to the nominal. Each attempt to fit an entering aircraft into a given position involves an exhaustive application of the appropriate logic. The last resort for an entering aircraft is the end of the queue.

The speed adjustment of an aircraft already on a link is called a re-set. MultiFit control is one of two places where the simulation is allowed to adjust an aircraft's speed and vectoring in mid-link. Re-set can increase speed, decrease speed or add vectoring. The change of speed is only applied to the portion of the link that the aircraft has left to travel. Vectoring can add the total amount of vectoring time possible on the link, unless the aircraft in question is already vectoring; then only the portion not already allocated to vectoring can be added to the aircraft.

Thus, each time an aircraft is to be placed into a queue position by MultiFit, the following restrictions apply:

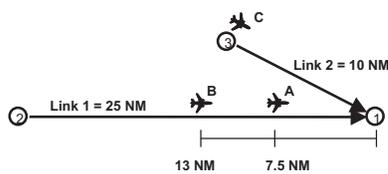
1. For each queue position, only three aircraft can be adjusted:
  - The aircraft entering the queue
  - The aircraft preceding the entering aircraft in the queue
  - The aircraft succeeding the entering aircraft in the queue.
  - Every adjusted aircraft must have separation with aircraft before and after it in the queue.
  - No adjusted aircraft may violate the minimum or maximum speeds to achieve position.
  - An aircraft's adjusted position can include the vectoring delay only if light/heavy sequencing is not in effect.
  - Only the aircraft entering the queue can include holding delay to achieve a position after nominal position.

MultiFit uses the following steps to fit an entering aircraft in a queue:

2. Try to fit the aircraft into its nominal position in queue:
  - Use nominal speed of entering aircraft
  - Increase speed of entering aircraft
  - Decrease speed of entering aircraft
  - Decrease speed of entering aircraft and add vectoring delay (if allowed)

- Increase speed of preceding aircraft; adjust entering aircraft by decreasing speed and vectoring (if allowed)
- Adjust the succeeding aircraft by decreasing speed and vectoring (if allowed); adjust the speed of the entering aircraft by decreasing its speed and vectoring (if allowed) or by increasing its speed
- Increase speed of preceding aircraft; adjust the succeeding aircraft by decreasing speed and vectoring (if allowed); adjust the speed of entering aircraft by decreasing speed and vectoring (if allowed) or increasing speed

Figure 12-15 shows how the MultiFit strategy logic might fit an aircraft in a nominal queue position by increasing the speed of the aircraft preceding it in the node arrival queue. Figure 12-16 shows how the MultiFit strategy logic might fit an aircraft in a nominal queue position by decreasing the speed of the aircraft succeeding it in the node arrival queue.



**Assumptions:**

All aircraft are the same type.  
 Maximum speed = 320 knots  
 Nominal speed = 300 knots  
 Minimum speed = 280 knots  
 Maximum vectoring delay on link 2 = 1 minute  
 Minimum separation for like aircraft is 3 NM  
 (= 36 seconds at 300 knots)  
 No wind.

**Analysis:**

Current time = 1:50:00

**Aircraft A's**

TOA = 1:51:30 with 7.5 NM from node 1.  
 For following aircraft with minimum intrail separation,  
 TOA = 1:52:06.

**Aircraft B's**

TOA = 1:52:36 with 13 NM from node 1.  
 For following aircraft with minimum intrail separation,  
 TOA = 1:53:26.

**Aircraft C's**

TOA at nominal speed = 1:52:00  
 For following aircraft with minimum intrail separation,  
 TOA = 1:52:36.

Aircraft C does not have separation with A.

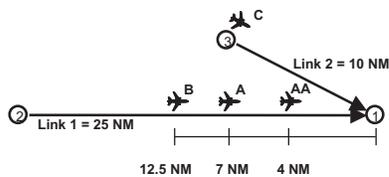
Increasing A's speed to 320 knots allows C to remain in position.  
 TOA at minimum speed = 1:52:09

**Aircraft A's new**

TOA = 1:51:24 at speed of 320 knots with 7.5 NM from node 1.  
 For following aircraft with minimum intrail separation,  
 TOA = 1:52:00.

**Figure 12-15: MultiFit at Nominal Position with Speed Increase of Preceding Aircraft**

3. Try to fit an aircraft into one of the positions forward from nominal in queue, starting with the position closest to nominal:
  - Increase speed of entering aircraft
  - Increase speed of preceding aircraft; adjust entering aircraft by increasing speed
  - Adjust the succeeding aircraft by decreasing speed and vectoring (if allowed); increase the speed of entering aircraft
  - Increase speed of preceding aircraft; adjust the succeeding aircraft by decreasing speed and vectoring (if allowed); increase the speed of entering aircraft



**Assumptions:**

All aircraft are the same type.  
 Maximum speed = 320 knots  
 Nominal speed = 300 knots  
 Minimum speed = 280 knots  
 Maximum vectoring delay on link 2 = 1 minute  
 Minimum separation for like aircraft is 3 NM  
 (= 36 seconds at 300 knots)  
 No wind.

**Analysis:**

Current time = 1:50:00

**Aircraft AA's**

TOA = 1:50:48 with 4 NM from node 1.

For following aircraft with minimum intrail separation,

TOA = 1:51:24.

**Aircraft A's**

TOA = 1:51:24 with 7 NM from node 1.

For following aircraft with minimum intrail separation,

TOA = 1:52:00.

**Aircraft B's**

TOA = 1:52:30 with 12.5 NM from node 1.

For following aircraft with minimum intrail separation,

TOA = 1:53:06.

**Aircraft C's**

TOA = 1:52:00 at nominal speed with 10 NM from node 1.

Nominal speed position has no fit between A and BB.

For following aircraft with minimum intrail separation,

TOA = 1:52:36.

Aircraft C does not have separation with B.

Adjustment of C's speed alone is not sufficient.

Adjustment of A's speed is not possible.

Adjustment of B's speed is possible.

**Aircraft B's new**

TOA = 1:52:36 at speed of 288 knots with 12.5 NM from node 1.

For following aircraft with minimum intrail separation,

TOA = 1:53:12.

Aircraft C, A, and AA remain unchanged.

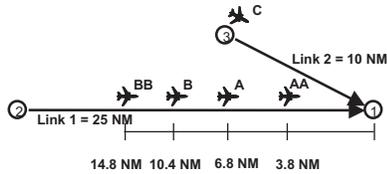
**Figure 12-16: MultiFit at Nominal Position with Speed Decrease of Succeeding Aircraft**

4. Try to fit an aircraft into one of the positions backward from nominal in queue, starting with the position closest to nominal:
  - Decrease speed of entering aircraft
  - Decrease speed of entering aircraft and add vectoring delay (if allowed)
  - Decrease speed of entering aircraft and add vectoring (if allowed) and holding delay
  - Increase speed of preceding aircraft; adjust entering aircraft by decreasing speed, vectoring (if allowed) and holding
  - Adjust the succeeding aircraft by decreasing speed and vectoring (if allowed); adjust the speed of entering aircraft by decreasing speed, vectoring (if allowed) and holding or by increasing speed
  - Increase speed of preceding aircraft and adjust the succeeding aircraft by decreasing speed and vectoring (if allowed); adjust the speed of entering aircraft by decreasing speed, vectoring (if allowed) and holding or by increasing speed.

Figure 12-17 shows how the MultiFit strategy logic might fit an aircraft at a position back in the queue by applying the following steps:

- Rule out a nominal or forward position in the queue.
- Attempt a fit behind the first aircraft succeeding the nominal position (i.e., behind aircraft B). Exhaust all approaches except the last.
- Increase the speed of aircraft B. For the purposes of this particular attempt, B is the preceding aircraft.
- Decrease the speed of aircraft BB, the succeeding aircraft for this attempt.
- Adjust entering aircraft C by decreasing its speed and adding vectoring delay.

- Aircraft C is fit between B and BB. No further attempts need be made. Aircraft C does not go to the end of the queue.



**Analysis:**

Current time = 1:50:00

Aircraft AA's

TOA = 1:50:46 with 3.8 NM from node 1.  
 For following aircraft with minimum intrail separation,  
 TOA = 1:51:24.

Aircraft A's

TOA = 1:51:22 with 6.8 NM from node 1.  
 For following aircraft with minimum intrail separation,  
 TOA = 1:52:00.

Aircraft B's

TOA = 1:52:05 with 10.4 NM from node 1.  
 For following aircraft with minimum intrail separation,  
 TOA = 1:53:06.

Aircraft BB's

TOA = 1:52:58 with 14.8 NM from node 1.  
 For following aircraft with minimum intrail separation,  
 TOA = 1:53:34.

Aircraft C's

TOA = 1:52:00 at nominal speed with 10 NM from node 1.  
 Nominal speed position has no fit between A and BB.  
 For following aircraft with minimum intrail separation,  
 TOA = 1:52:36.

No space at nominal position.

No possible to make either forward position

**Assumptions:**

All aircraft are the same type.  
 Maximum speed = 320 knots  
 Nominal speed = 300 knots  
 Minimum speed = 280 knots  
 Maximum vectoring delay on link 2 = 1 minute  
 Minimum separation for like aircraft is 3 NM  
 (= 36 seconds at 300 knots)  
 No wind.

**MultiFit searches for first backward position:**

Not enough space by adjusting B and C.  
 Not enough space by adjusting BB and C.  
 Enough space by adjusting B, BB, and C!

Aircraft B's new

TOA = 1:52:58 at 317 knots  
 For following aircraft with minimum intrail separation,  
 TOA = 1:52:34.

Aircraft C's

TOA = 1:52:34 at 280 knots with 25 seconds of vector delay  
 For following aircraft with minimum intrail separation,  
 TOA = 1:53:10.

Aircraft BB's

TOA = 1:53:10 at 280 knots  
 For following aircraft with minimum intrail separation,  
 TOA = 1:53:46.

Figure 12-17: MultiFit at Back Position, Speed Decrease of Succeeding Aircraft

5. Fit an aircraft into last position in queue:
  - Decrease speed of entering aircraft
  - Decrease speed plus vectoring (if allowed) of entering aircraft
  - Decrease speed plus vectoring (if allowed) and holding delay of entering aircraft
  - Increase speed of preceding aircraft and adjust entering aircraft by decreasing speed, adding vectoring (if allowed) and holding delay

## 12.7 Level II: Metering Strategy

Metering Strategy is an optional strategy that enhances the simulation's ability to control aircraft movement. It models the processes by which a controller looks ahead along the route network and handles projected downstream traffic.

The basic airspace movement rules allow the simulation to coordinate aircraft approaching each node on a route. Metering allows the simulation to see downstream congestion by projecting aircraft positions at key nodes along a route. Based on the projection, the simulation will attempt to space aircraft to minimize downstream congestion or divert aircraft to a less congested route.

Metering is limited in that its adjustments must be carried out within the constraints established by the operative node arrival control strategy; for example, it cannot change the sequence of an aircraft in the node arrival queue if this has already been established by the Level I node arrival control strategy.

Metering Strategy is discussed below in two sections: Metering Node Structure and Metering Logic.

### **Metering Node Structure**

To use meter control, certain nodes are designated as meter post nodes and others as meter nodes.

Post nodes are generally located at bottlenecks or critical merge points in the airspace network. Each post node is associated with a group of meter nodes, which are located before it on the airspace route(s). Specifically, meter nodes are located at points along a meter controlled route where the simulation should invoke the meter control (Level II) strategy. (Other routes, which are not heading for the meter post node and are not associated with it, may happen to include a meter node, but these routes are not controlled by the metering strategy.)

A meter post node should be located at a critical merge point in the airspace where normal airspace movement controls might not be capable of managing aircraft delay — the node immediately prior to the airport interface, for example. The simulation allows any airspace node to be defined as a meter post node.

Another consideration in locating the meter post node is this node's role in diverting aircraft to alternative routes. The analyst may cause metered aircraft to be diverted to an alternate route based on the number of aircraft expected to arrive at the meter post node. Similarly, the analyst may deny aircraft from other routes any access to a specific metered route based on the number of aircraft due to arrive at this post node.

A meter node should be located at a point where meter control logic can successfully initiate the control of downstream traffic, i.e., it should be a node sufficiently far upstream from the post node to allow aircraft adequate space and time to meet minimum separation requirements by the time they reach the post node.

Note, however, that meter nodes should not be located at the outermost airspace nodes: aircraft should not enter the simulation at a meter node.

### **Meter Logic**

Any aircraft between a meter node and its post node on a meter-controlled route is subject to meter control actions.

Meter controls function in cooperation with the (Level I) node arrival control. As each flight progresses along a route, node arrival control is exercised at each node. If the flight is on a metered route, metering control is in effect from the time the flight approaches the meter node on a link until it passes through the post node.

When an aircraft subject to metering arrives at a node, the node arrival control logic is first applied to determine the aircraft's movement to the next node along the route and its position in the next node's arrival queue.

Next, the meter control logic (Level II) "looks ahead" to the post node for the route. It projects the status of traffic at this post node at the aircraft's arrival time, assuming the aircraft will travel at nominal speed beyond the next node. All traffic converging on the post node is considered in this look-ahead.

Meter control must maintain each aircraft's relative position in the next node's arrival queue as established by the node arrival control. If a future conflict with another aircraft is projected, meter control actions are taken to fully or partially alleviate the conflict.

Each meter post node has a meter queue in addition to its normal node arrival queue and node holding queue. An aircraft enters the meter post queue only if it is on a metered route and only after being placed in the node arrival queue for the initial meter node.

The meter post queue lists the aircraft approaching a meter post node in order of their projected TOA. The earlier the TOA at the post node, the more advanced an aircraft will be in the meter post queue. This projected TOA for an aircraft is the sum of the TOA in the node arrival queue at the meter node plus the time required by the aircraft to traverse all links between the meter node and meter post node at nominal speed.

However, the meter control cannot change the relative position (i.e., the sequence) of the new aircraft in the meter queue, nor can it change the nominal time required by the aircraft to traverse the intermediate links. Consequently, any adjustment of a TOA in the meter queue must be made by changing the aircraft's TOA in the current node arrival queue.

Meter control adjustments cannot change the sequence of any aircraft in the node arrival queue either, but they can change the entering aircraft's specific TOA insofar as is possible without violating its separation requirements. If the new aircraft in the meter queue is inadequately separated from only one aircraft in that queue (either the immediately preceding or succeeding aircraft), the meter control will attempt to fill this separation requirement by reducing the new aircraft's separation from the other aircraft to the minimum required. The difference saved will provide extra — but not necessarily adequate — separation time where it is needed.

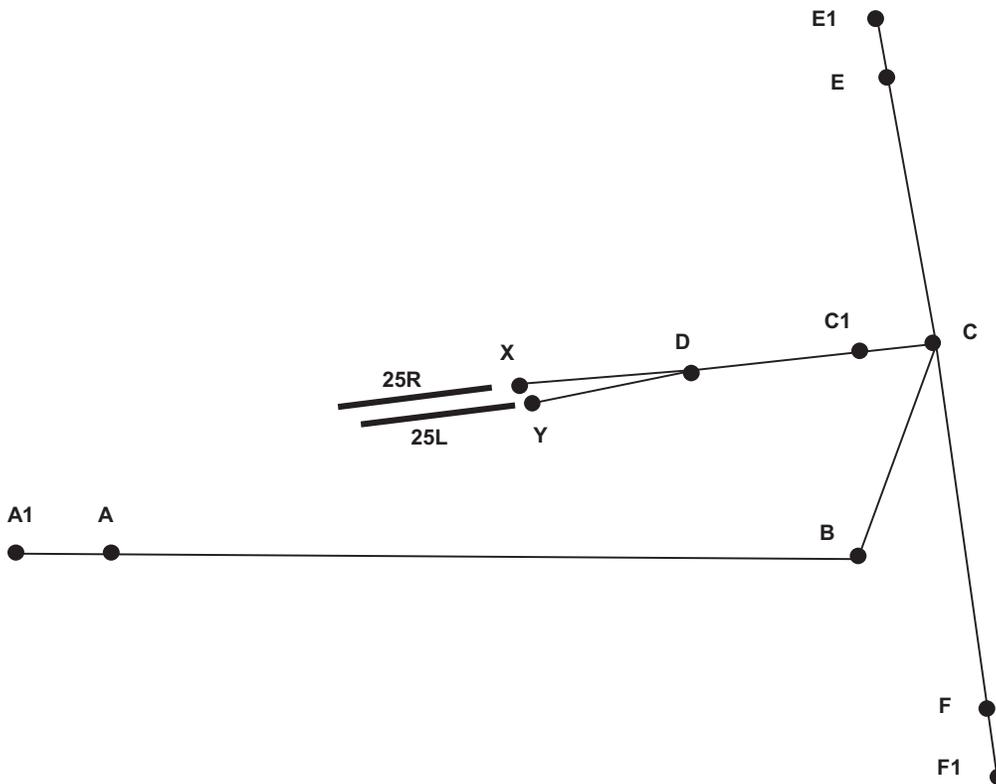
If an aircraft entering the meter queue has inadequate separation from both the preceding and succeeding aircraft, the meter control will not attempt to ensure separation from either. The node arrival logic always limits the meter logic somewhat by establishing the sequence of aircraft in the node arrival queue. The node arrival control strategy that least inhibits the meter logic is QFIFO, because QFIFO always places the aircraft last in the node arrival queue. With an aircraft in this position, the meter logic can always attempt to resolve conflicts at the post node by delaying the aircraft at the current node.

The other node arrival control strategies, SpeedFit and MultiFit, might place the aircraft in a position that does not allow for the resolution of the metering queue separation requirements. Although only one meter control has been programmed to date, the modular design of the program will facilitate the addition of new types of meter control.

The following example describes the application of metering logic shown in Figure 12-18:

Assume that the aircraft flying through nodes A, E, and F, should be metered so that they will be separated at C and not be delayed on the final approach. To achieve metering:

- Define A, E, and F as meter nodes. Since injection nodes should not be meter nodes, injection nodes A1, E1, and F1 should not be meter nodes.
- Define C as a post node.
- Assign each route that passes through C that you want metered, to the corresponding meter node that it also transits.



**Figure 12-18: Metering Example Airspace Structure**

The aircraft will hold at A, B, E, and F so that they are separated when they reach C. It is not possible to prevent the metering logic from imposing delay at B should the logic determine that delay is necessary.

As mentioned above, aircraft may be diverted to alternate routes depending on the level of congestion projected at the meter post node. This projection is based on the number of aircraft in the meter post queue. In order for aircraft to be diverted to alternate routes, the user must specify the change to an alternate route via the plan record. (For more details, see “Dynamic Airspace” in Section 14, Interface Logic).

## 13 Airfield Logic

Nodes and connecting links define SIMMOD airfield structures:

- Runways
- Departure queues for holding and sequencing aircraft departing on runways
- Gates for loading and unloading aircraft
- Taxiways for aircraft movement between gates and runways
- Towing areas
- Runway crossings
- Dynamic single direction (DSD) paths

Airfield nodes are points in a two-dimensional coordinate system (corresponding to a basically flat airfield). The only attributes assigned to every node are its latitude and longitude (the X,Y coordinates). Airfield nodes typically mark taxiway intersections, runway exits or crossings, gates, towing areas, de-icing areas, staging areas, taxi checkpoints, and departure queues.

Airfield links describe taxiways, runways, runway exit paths, and departure queue paths. Figure 13-1 shows an airfield network with gates, taxiways, runways, and runway exits.

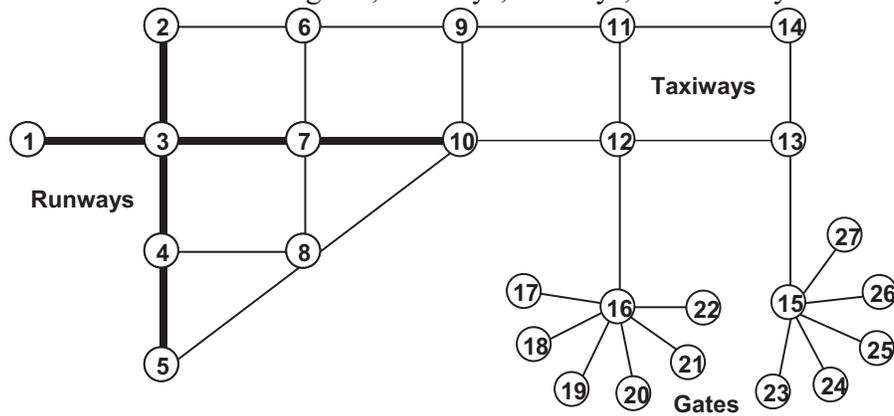


Figure 13-1: Airfield Network

The link is the more consequential of airfield structures. The attributes of the airfield link are significant in determining aircraft movement. Links are defined by their initial and final nodes; in addition to its length, each link has the following attributes:

- Assignment to arriving aircraft, departing aircraft, or both
- Maximum number of aircraft allowed on the link
- Passing rules (no passing, passing in one direction only, passing in both directions)
- Aircraft groups allowed on the link
- Direction of aircraft movement (from initial to final node, from final to initial node, or both directions)
- Taxi speed on the link, in knots

### 13.1 Runways

A runway is defined as a list of links from one end to the other. Runway links are defined in one direction and may be used in both directions.

#### Runway Exits

Runway exits can be defined at the end of each link on a runway. Which runway exit is used for any simulated landing depends on where the aircraft finishes its landing roll. Any exit reached after completion of the roll is a viable exit. The taxipath optimization logic selects the specific exit.

A link connected to the runway may be defined as a high-speed exit. These are frequently used to reduce runway occupancy time. The difference between the headings of the link and the runway determines the amount of the landing roll that may be completed on the high-speed exit.

### Amount of Landing Roll Completed on High Speed Exit

Change in Heading	% of Roll Completed
10 deg	20%
20 deg	15%
30 deg	10%
40 deg	5%

### Runway Blockage

When an aircraft is performing a takeoff or landing roll on a runway, all links and nodes of the runway are blocked to other aircraft taking off or landing (see Figure 13-2).

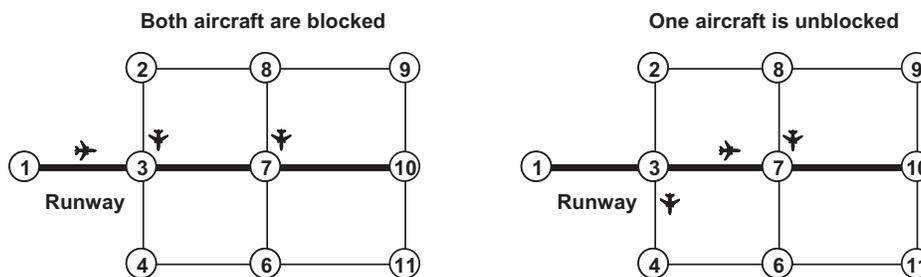


Figure 13-1: Runway and Taxiway Interaction

Before an arriving flight lands on the runway, it also blocks aircraft from crossing the runway. This blocking begins before the arrival lands on the runway by a period of time defined as the expected runway delay time. This is generally equivalent to the length of time an aircraft requires to cross the runway, because all runway crossings should be finished by the time the arrival touches down. The blocking ends at each node on the runway as the landing aircraft passes it, unless a user-defined procedure determines otherwise (see Section 14, Interface Logic). For departures, blocking begins before the takeoff is cleared to depart by the expected runway delay time. The blocking ends at each node on the runway as the departing aircraft passes it, unless a user-defined procedure determines otherwise.

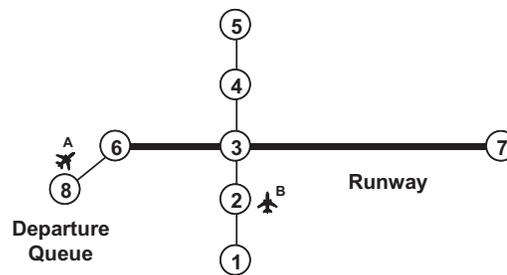
### Runway Crossing Priority

Landing and departing aircraft have priority for using the runway and may continually block aircraft from crossing based on the spacing of the landings and takeoffs. However, the simulation parameters can be reset to give priority to runway crossings. Depending on the number of aircraft waiting to cross and the time any aircraft has been forced to wait, SIMMOD can adjust the intrail separation of arriving aircraft to force a break. Small changes are made dynamically as needed. The change is specified as either an increase to aircraft inter-arrival times, or as an increase in the separation distance between arrivals. The requirements for establishing larger changes in arrivals are covered in the section "Setting Conditions for a Forced Runway Crossing" in Section 15, Resetting Simulation Parameters.

### Runway Crossing Times

Optional logic exists in SIMMOD to allow the user to model runway crossings from hold line to hold line. The optional RWYCROSS record of the Airfield file allows the user to assign specific

runway crossing times and characteristics to each runway crossing. SIMMOD's runway crossing logic will check departing and arriving aircraft before allowing an aircraft to cross the runway. If the RWYCROSS logic is not used, SIMMOD uses the DefDly value of the RUNWAY record to determine the amount of time a plane needs to cross the runway. A plane will be allowed to cross the runway as long as it can finish crossing by the time an arrival or departure reaches the same crossing node. The SETXNG record may also be used to give priority to crossing planes. Refer to Section 15, Resetting Simulation Parameters, of this manual for details of SETXNG. A global variable, RC\_Fudge\_Time, in the GLOBAL record provides additional flexibility when using the runway crossing logic. This constant value, in seconds, is subtracted from the total time SIMMOD has calculated for an aircraft to cross a runway. This allows the user to reduce the window of time an aircraft requires before it may attempt a runway cross. Once an aircraft is crossing the runway, the departure queue will hold departing aircraft until the runway crossing is completed (see figure 13-3). This global option only works with the runway crossing logic enabled. Coordination of runway usage is discussed in Section 14, Interface Logic.



**Assumptions:**

Time = 1.0 hours  
 Aircraft B takes 0.2 hours to cross the runway (node 2 to node 4)  
 Aircraft A is scheduled to leave the queue at 1.15 hours

**Analysis:**

Aircraft B will hold at node 2 because  
 $1.0 + 0.2 = 1.2 > 1.15$   
 thus there is not enough time for aircraft B to complete the runway crossing before aircraft A leaves the departure queue.

Setting the global variable rc\_fudge\_time to a value of 6 minutes (0.1 hours) will switch the priority from the departure queue to the runway crossing. Since it will take aircraft B 0.2 hours to cross the runway, aircraft A will not leave the departure queue until 1.2 hours.

Figure 13-3: Runway Crossing Logic Using the RC\_FUDGE\_TIME Global Variable

### 13.2 Gates

Every aircraft entering the airfield is assigned to a gate. All gates are located at defined airfield nodes. Aircraft enter or leave the simulation at gates or towing areas. Once at the gate, a flight is held for the duration of at least one gate service time in order to model gate occupancy. This period of time accounts for either the loading or unloading of aircraft passengers and supplies. Gates are defined with a user-specified service time distribution, which is sampled for both loading and unloading.

If a flight uses towing (optional), aircraft will hold at the gate for an additional amount of time. This time is chosen from the gate towing time distribution in the TAMPS record and accounts for the additional time spent during the towing procedure. If the aircraft is a turnaround flight, the aircraft is held for the duration of both times, i.e., both are applied to the aircraft. Gate access

restrictions can be set according to aircraft model or TAMPS group in the optional GATEUSE record. SIMMOD no longer restricts gate use by aircraft size.

### Gate Ownership

A gate can belong to an airline, to a group of airlines, or to all airlines. Every flight that enters the system is identified with an airline, and it can use only the gates available to that airline. Such gates include those specifically assigned to the airline and those available to all airlines. An arriving flight can be assigned to a specific gate, any gate owned by its airline, or left unassigned (i.e., able to use any available gate). A gate's alternate-gate flag (the GFLAG field in the GATE record) can be set to let arrivals assigned to that gate seek another gate if the assigned gate is unavailable.

### Multiple Gates Modeled as One

SIMMOD allows you to represent a group of gates as a single gate with a capacity greater than one aircraft. Figure 13-4 shows the relationship between multiple gates and a group gate representing them. The group gate might be located at the approximate center of the actual group or at the end of a link connecting them all.

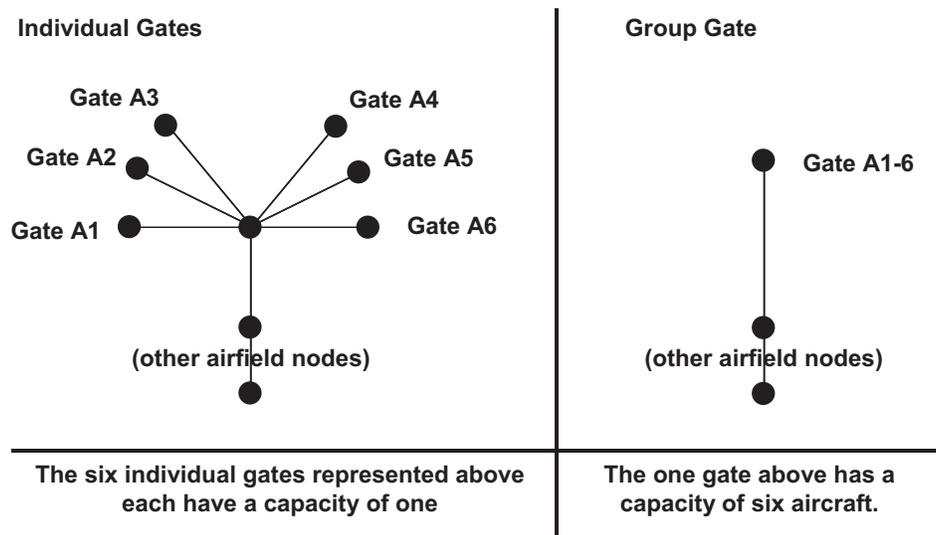


Figure 13-4: Gate Groups

This gate-grouping technique allows SIMMOD to maintain accurate statistics for overall gate utilization, but does not show how individual gates are actually used. Grouping reduces program execution time and generally should be used unless you have a particular interest in individual gates.

### Grouping Gates into Concourses

A concourse is a collection of gates. The user can reference a concourse rather than a single gate in the following records where a gate is referenced: EXIT, GATERWY, and GATETOW.

### Gate Blocking and Gate Restrictions

Gate blocking and gate use parameters have been removed from the GATES record and combined into a new record called GATEUSE. Gate blocking occurs when a power-back or

push-back operation at the gate blocks access to other gates. The gate can allow push-backs only, or allow both push-backs and power-backs, or not model push/power-backs for this gate. If no push/power-backs are modeled, then the taxi speed for the link is used for time calculations. The GATEUSE record is used to specify gate blocking. Model blocking occurs when a specific model of aircraft (e.g., a jumbo jet) occupying a gate blocks access to other gates. SIMMOD allows users to specify that only certain model types at related gates may be blocked. For each gate, a list is created which specifies all the aircraft types that block other gates, along with the gates blocked and models blocked. This list also determines which aircraft can occupy the gate. This list is developed based on data fields provided in a new record called GATEUSE. Link blocking occurs when a power-back or push-back operation at the gate blocks access to ground links near the gate. Each gate may define a list of links that are blocked during a power-back or push-back. When an arriving flight's assigned gate is blocked, it can wait until the gate is available, or if the definition of the gate permits, it can seek an alternate gate. SIMMOD models gate access using aircraft model type to determine whether or not an aircraft may use a gate. This allows the user more flexibility in determining which aircraft may access which gates. Some gates at airports are not equipped to handle certain types of aircraft because of aircraft size or the need for specialized equipment.

### **Gate Logic**

Gate assignments for an individual flight can be determined by data input or left to SIMMOD. If gate blocking has occurred and alternate gating is an option, gate ownership limits the available choices in the simulation. The model of the aircraft is also a consideration. When a flight is created at an airfield or enters the airfield from the airspace, SIMMOD performs a gate check. The process for the check is as follows:

1. SIMMOD checks to see whether the flight is assigned to a specific gate.
2. If not, SIMMOD checks for available gates and assigns the flight to one at the first possible time.
3. If the flight is assigned, SIMMOD checks the availability of that gate.
4. If the gate is available, it is reserved for the flight at its gate arrival time.
5. If the gate is not available, SIMMOD checks whether it is allowed to find an alternative to this gate.
6. If SIMMOD cannot seek an alternate gate, it will reserve this gate at the first available time.
7. If SIMMOD can seek alternates, it will search for available gates and reserve one for the flight.

### **13.3 Links**

Certain aircraft may be restricted from using some airfield taxiways because of weight or size restrictions. SIMMOD allows users to specify link travel restrictions based on aircraft model types or TAMPS groups. The default value allows all aircraft to use all links.

### **Taxipaths and TaxiPlanning**

Taxiplans define the routing between runways and gates, and towing areas and gates. The SIMMOD taxiPlanning logic automatically determines an optimal series of sequential links in the direction of travel. The selection of the optimal taxiplan can be influenced by the user through the use of the TAXIPATH inputs.

## Optimized Taxiplans

SIMMOD will optimize a flight's taxiplan between a gate and runway based on the defined link characteristics, projected congestion, taxi time, and any taxipaths defined for the flight.

SIMMOD evaluates the costs (in aircraft taxi time) of possible taxiplans and selects the one with the lowest cost.

The cost that each link adds to the total taxiplan cost is based on the time an aircraft requires to traverse the link, plus any expected delay on the link at the time the aircraft is projected to be on it.

During the taxiplanning optimization, taxiplans are created by considering each link that extends from the origin node. The cost (taxi time) of reaching the node at the other end of each feasible link is calculated and stored as a list of taxiplans in order of least cost. Then the logic considers all the links that extend from the end node of the taxiplan with the lowest cost. This taxiplan is removed from the list and replaced by the taxiplans resulting from the addition of the links attached to this end node. The cost of these added taxiplans equals the total cost of the removed taxiplan and the incremental cost of the ground link that is added to create each new taxiplan.

The list of taxiplans is then resorted in order of least cost.

The logic repeats this process until a taxiplan has been found between the origin and destination ground nodes and all other plans in the list have a greater total cost. Such a taxiplan is optimal. The number of taxiplans and number of links in each taxiplan can be limited by input, thus limiting the number of possibilities SIMMOD will try. SIMMOD will halt the simulation if no taxiplan can be found.

Each link in a user-defined taxipath assigned to an aircraft is given a zero cost. A fully defined taxipath (one that includes all the links between the origin and destination nodes) will almost always be considered the optimal taxiplan for a flight because of its zero cost. It is sometimes beneficial to define a partial taxipath and specify it for a flight. SIMMOD will complete the taxiplan with its optimizer. Links explicitly listed in the definition of a taxipath — even a partial taxipath — have zero cost. See Figure 13-5, Taxiplan Optimization.

In this example, the taxiplanning optimization routine will attempt to determine the lowest cost (i.e., time-efficient) plan for a given aircraft taxiing from its gate to its departure queue represented by node 1 and node 7 respectively.

	Node Sequence	Total Cost per Complete Plan
Plan A	1 2 4 7	6 minutes
Plan B	1 2 4 6 7	8 minutes
Plan C	1 2 4 5 7	10 minutes
Plan D	1 2 4 5 7	8 minutes

Plan A will be the first plan completed by the optimizer. The other plans will be compared against A as they are developed. Because plan A has the lowest cost, it will be selected as the optimal plan.

Note that a give link (e.g., 5-7) may have different costs at different times depending on an aircraft's projected time of arrival.

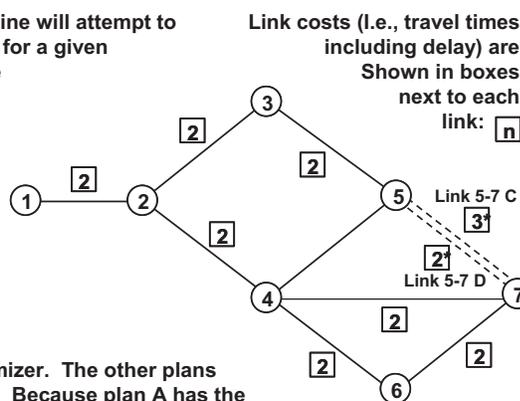
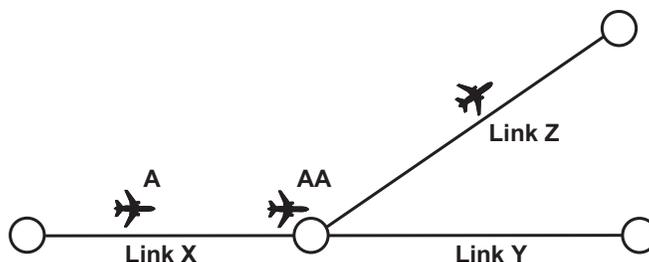


Figure 13-5: Taxiplan Optimization

## Taxipaths and Link Characteristics

The optimization of a flight's taxiplan considers the characteristics of each potential link to determine which are viable. It then considers the costs of viable links at the time when the aircraft is projected to enter them.

If the link is flagged for arrivals only and the flight is a departure, the link will not be considered viable for the optimized path. Similarly, departure-only links will not be considered for arrivals. If the link allows travel in only one direction, e.g., from the initial to the final node of the link, the link will not be considered for aircraft traveling in the opposite direction. If an aircraft exceeds a link's size restriction, the link will not be included in the optimization.



Link X does not allow passing.  
Link Z has a capacity of 1 aircraft.  
Aircraft AA must hold on link X until it can proceed onto link Z.  
Aircraft A must wait behind AA on link X to get to link Y.

Figure 13-6: Taxiplan Link Characteristics

The capacity of a link affects its cost by incurring delay. If a link does not have the capacity to accept an aircraft at its projected time of entry, the aircraft may be delayed until space is available. The delay will be added to the cost of the developing path.

Passing restrictions can also affect the cost of using a link by incurring and compounding the delays of other aircraft traversing the link.

Suppose, for example, that a link X, which allows no passing, connects to links Y and Z (see Figure 13-6). It is projected that when aircraft A will need to traverse link X to reach an open link Y, aircraft AA will be holding on X until space is available on link Z. Aircraft A will be required to hold on link X until AA clears X. The delay incurred will be included in the cost of using X as part of A's taxiplan.

## Assigning Taxipaths Depending on Gate, Runway, and Arrival/Departure Status

SIMMOD's optional GATERWY record allows the user to make gate/runway/arr\_dep specific assignments in order to supplement the taxi planning optimization logic. The GATERWY record may be used to assign a taxipath when the gate assignment is not known in advance. For instance, if SIMMOD chooses a random gate available to the flight's airline, the model may reference the GATERWY record to find a taxipath or partial taxipath that is suitable for that gate/runway/arr\_dep combination.

## Assigning Taxipaths Depending on Gate, Towing, and Arrival/Departure Status (Towing aircraft only)

SIMMOD's optional GATETOW record allows the user to make gate/towing/arr\_dep specific assignments in order to supplement the taxi planning logic. The GATETOW record may be used to assign a taxipath when the gate or towing area assignment is, or is not, known in advance. For

instance, if SIMMOD chooses a random gate/towing area available to the flight's airline, the model may reference the GATETOW record to use a taxi path that is suitable for that gate/towing/arr\_dep combination.

### 13.4 Departure Queues

To get to its runway, a departing aircraft must enter a departure queue. The usual choice for the location of a queue is at the end of a runway, but this location is not mandatory. Any taxi path specified for the flight will be applied to the taxi planning between the queue location and the point on the runway where the takeoff roll is to begin. If specified, SIMMOD will model the movement of the aircraft to the begin-roll point.

A departure queue may be associated with many routes, and can service more than one runway. Furthermore, a given runway may be serviced by several departure queues. However, a route can have only one departure queue.

SIMMOD allows you to define fragmentary routes. A fragmentary route consists of the links beginning at the interface node adjacent to the secondary runway and ending at a node belonging to the primary route. You must assign the fragmentary route to the departure queue servicing the alternate runway. See Figure 13-7, Alternate Runway Departures with Fragmentary Route Sections. Thus, when a departure flight is rerouted to an alternate runway, it enters the airspace at an interface node adjacent to the end of the runway and subsequently joins the original departure route. (For more details, see "Dynamic Airspace" in Section 14, Interface Logic). A departure queue is defined at a node, but aircraft waiting in the departure queue hold on the links leading into it. An aircraft cannot enter the queue unless it reaches the end of one of the links immediately leading into the queue.

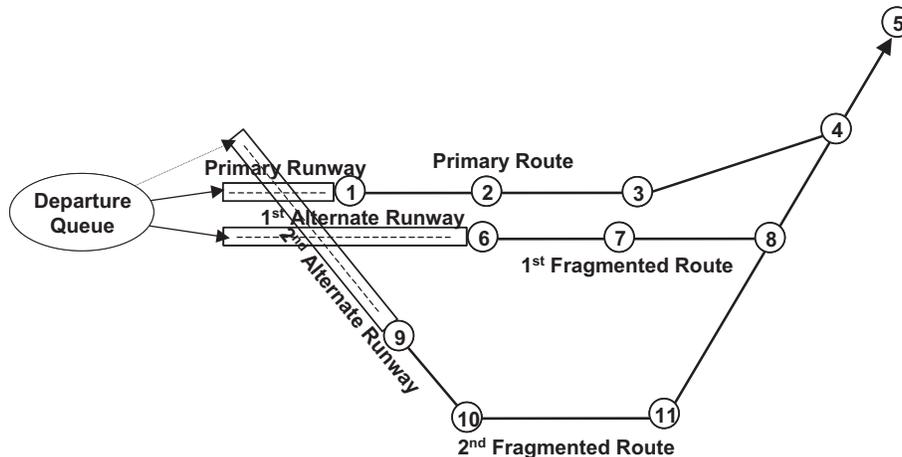


Figure 13-7: Alternate Runway Departures with Fragmentary Route Sections

The capacity of the departure queue is independent of the capacity of the ground links leading into it. If the sum of capacities of the ground links that enter the departure queue node exceeds the capacity of queue, it is possible for aircraft to be waiting at the ends of the entry links without actually being entered into the departure queue. In this case, the waiting aircraft are filed into a departure queue waiting list. Such an aircraft will automatically enter the departure queue when a space becomes available due to the release of another aircraft already in the queue.

Only aircraft in the departure queue are candidates to be released for takeoff. There are two departure queue strategies. The first is a strict first in, first out (FIFO) queue. The first aircraft in the queue is the only aircraft eligible to move out of the queue. No passing is allowed.

The second strategy allows an aircraft other than the first be released from the queue and taxi to its runway. This passing feature is dependent on runway or departure-queue availability, the aircraft's position in the queue, and the aircraft's minimum and maximum departure time (if specified in the SLOTWINDOW record). If the first aircraft is not cleared to depart, SIMMOD will search the queue to the depth permitted by the definition in the Departure Queue record.

### 13.5 Aircraft Groups

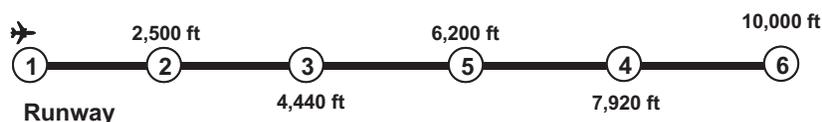
#### Landing-Roll Distance and Time

The landing roll distance required by an arriving aircraft is stochastically determined for each flight based on the aircraft type and the user-specified landing roll distance probability distribution for that aircraft group. See Figure 13-8 , Runway Roll Times.

The distance *d* is the random distance drawn from the linear roll distance distributions, defined in the TAMPS card, for aircraft group.

For landings, the final roll velocity is the runway taxi speed. For takeoffs, it is the nominal link speed on the first airspace link.

For landings, the initial velocity is the aircraft's velocity across its last airspace link. For takeoffs, it is zero knots.



**Roll distance of 6,200 ft  
Roll time of 50 seconds**

Node	Cumulative Roll Distance	TOA at node
1	0	1:50:00
2	2,500	1:50:20
3	4,440	1:50:36
4	6,200	1:50:50

Figure 13-8: Runway Roll Times

SIMMOD can model landings using a greater landing threshold or landing roll distance for a particular gate/runway/model combination. In these cases, an aircraft may be forced to use a fixed percentage of the runway length as a roll distance.

SIMMOD will always choose as the actual landing roll the larger of either the landing roll distribution or the exit runway percentage length.

Note: The distance rolled for takeoff and landing is the distance from starting point of the roll to the first node at which the cumulative length traveled exceeds the distance taken from the takeoff or landing roll distribution.

This means if your runway is a single 10,000 foot link, all rolls will be 10,000 feet. Or, if the first node is at 2,000 feet, the second node is at 6,000 feet, and the roll is to be 2,050 feet, then the roll will be 6,000, since that is the first node beyond the roll distance.

The time at which an aircraft crosses each runway node is in proportion to the runway roll distance completed. Consequently, deceleration is assumed to be constant.

## **Takeoff Roll Distance and Time**

The takeoff roll distance required by a departing aircraft is stochastically determined for each flight based on the aircraft type and the user-specified takeoff roll distance probability distribution for that aircraft group. As with the landing rolls, the time at which an aircraft crosses each runway node is in proportion to the runway roll distance completed. Acceleration is constant.

## **Power-Back and Push-Back Aircraft.**

Each gate in an airfield may have an associated power-back and push-back distribution. These distributions allow the model to take into account the different times required for each of these activities. SIMMOD airfield links only allow one speed type; however, the time it takes to enter a gate is much briefer than the time it takes to “push-back” and unhook the tow. Power back and push back distributions adjust the simulation to account for these time differences. Users must specify which aircraft will push-back or power-back in the PPBACK record.

## **13.6 Congestion, Holdcycles, Checkpoints, and Staging Areas for Arrivals**

A congestion area is a set of links and gates used in staging operations when determining an aircraft’s movement after completion of its landing roll. If the congestion area is saturated, the aircraft will go to a staging area or holdcycle, depending on the level of staging you are modeling.

Staging areas are specified areas on the airfield where aircraft can park while waiting for their gates to become free or for congestion levels in the gate area to decrease. In the GATES record users can specify that aircraft waiting for their gates to become free must go to a staging area before reevaluating gate status.

All staging areas must have an airfield node location and an airfield link that enters the node location. A staging area is defined as either a pad or a checkpoint.

A staging pad is an open area on the airfield where aircraft park to wait for an open gate. A pad must have a maximum number of aircraft allowed on the pad at one time, and a maximum number of aircraft allowed to queue for available room on the pad.

A checkpoint is usually located at the beginning of each taxipath contained in a holdcycle. A holdcycle is a group of user-defined taxipaths that form a circular path.

Upon the aircraft’s arrival at a checkpoint during a simulation, a decision is made by SIMMOD concerning the next movement of the aircraft. The decision depends on the current status of the congestion areas. The CHECKPOINT record includes the holdcycle number in which the checkpoint is a member, and a list of gates that can be accessed from this checkpoint.

Holdcycles are used in SIMMOD as a staging operation when an aircraft is not free to taxi directly to a gate upon completion of its landing roll. If the gate area is blocked, occupied, or congested, the aircraft will taxi in a defined holdcycle.

All holdcycles are defined by several checkpoint and taxipath pairs. Note that a holdcycle should be defined by taxipaths containing one link each. If a holdcycle consists of any taxipaths defined with two or more links, SIMMOD will use the taxi-planning optimization algorithm to obtain the shortest route. This will defeat the intention of defining a specific holdcycle and will also increase computation time.

## **Staging Areas, Congestion and Taxi Checkpoints for Departures**

The departure staging logic of SIMMOD provides for routing the departing aircraft to a departure staging area if the number of aircraft assigned to the departure queue reaches the user-defined threshold level.

A departure staging area is an open area (pad) on the airfield where aircraft park. A staging area must have a maximum number of aircraft allowed on the pad at one time, and a maximum number of aircraft allowed to queue for available room on the pad.

If the capacity of the staging area and the capacity of the queue are exceeded, aircraft are sent to the staging area with a warning message. When the departure queue level drops below the threshold level, any aircraft in the staging area awaiting movement towards the departure queue is released to taxi from the staging area. Users should not use departure staging logic without taxi checkpoints.

Using departure congestion areas in conjunction with departure staging areas imposes an additional restriction on the movement of departing aircraft. A departure congestion area is a set of links leading to the departure queue. Although the departure queue level may be below the threshold level, high levels of congestion near the departure queue may necessitate the staging of the departing aircraft until the congestion eases.

If the departure queue, departure staging area, and departure congestion links are all at capacity, aircraft are held at their gates as a last resort. An aircraft is allowed to taxi towards the designated departure queue only if the departure queue level is below the threshold level and the related congestion levels are below the specified maximum levels.

Taxi checkpoints add additional controls for routing departure traffic. When a gate is assigned a departure staging area, aircraft exit the gate and taxi to the assigned staging area. Taxi checkpoints are user-specified ground nodes where aircraft can reevaluate their taxipath and choose either to continue to the departure staging area or taxi directly to the departure queue. Users can specify multiple taxi checkpoints along each taxipath, and the departure staging/departure queue decision will be reevaluated at each taxi checkpoint node.

### **13.7 Departure De-icing**

The SIMMOD de-icing logic models the de-icing process on a flight-by-flight basis at designated de-icing areas. For each departure queue, airport plan, and gate combination, the user specifies a list of the de-icing areas that can be used by a departing aircraft destined towards the queue. The de-icing area consists of a pad area where de-icing is performed and a queue area where the aircraft waits for de-icing.

When an aircraft requires de-icing, it is routed towards the chosen de-icing area. The first listed de-icing area would be chosen if there is room for the aircraft either in the pad area or the queue area. The user-specified de-icing list specifies the various de-icing areas to be used in order of priority. If all the de-icing areas are full, the aircraft will be sent to the last defined de-icing area in the list regardless of the de-icing area constraints. However, a warning message will be printed in the log file indicating that an aircraft was sent to a de-icing area in violation of its capacity restrictions.

Upon arrival of an aircraft at the de-icing area, it is decided if there is capacity available at the pad area. Otherwise the aircraft occupies the queue area awaiting its de-icing turn. Upon completion of the user-specified delay for de-icing, the aircraft leaves the de-icing pad area and taxis towards the departure queue or the staging area based on departure staging considerations (see "Departure Staging Areas and Departure Congestion," above).

The de-icing logic also incorporates the capability of re-routing the aircraft for revisiting another de-icing station, if a user-specified de-icing active time has expired and SIMMOD has not yet scheduled the aircraft for takeoff. Re-routing is done if the aircraft is at the departure queue, staging area, or anywhere enroute to these waiting areas. If the aircraft is in the staging area queue, then it is rerouted once it occupies the passing position at the staging area. If an aircraft traversing a link is to revisit a de-icing station, rerouting is done upon its arrival at the next node.

### 13.8 Flight Banks

SIMMOD uses flight banks to model the dependency of arriving and departing flights in hub operations. See Figure 13-9, Bank Logic.

Bank Flights	Maximum Delay Time (minutes)	Connecting Passengers	Transfer Time for Passenger (minutes)	Current Status
A	15	Yes	10.2	Ready to depart
B	15	Yes	10.9	At gate, has been unloading for 2 minutes
C	15	Yes	11.0	At gate, has been unloading for 15 minutes
D	15	Yes	11.65	Will arrive in 12 minutes

Aircraft A will depart in 8.9 minutes based on the following information:

B delays A by 8.9 minutes (10.9 – 2 minutes).

C delays A by 0 minutes (11.0 – 15 minutes).

D would delay A 23.6 minutes (12 + 11.6 minutes); however, this exceeds the maximum delay time and is, therefore, disregarded.

Aircraft B's 8.9 minute delay of aircraft A remains effective, so A will depart in 8.9 minutes.

**Figure 13-9: Bank Logic**

A bank is a group of flights that are to be at their gates at the same time to allow passengers to make connections. SIMMOD does not limit the number of flights in a bank. If delays are encountered during the simulation, flights may not be able to make their connections at a hub. If a flight is part of a bank and is delayed in landing, some or even all other aircraft in the bank might be kept on the ground waiting for the late flight. Each departing flight in a bank makes the decision to wait based on four values:

- The user-defined maximum delay time that the flight can be held
- The transfer time required by the connecting passengers after their plane has landed (a stochastically adjusted value)
- The probability that there will be a sufficient number of connecting passengers from a heavy aircraft to justify delaying the departure
- The probability that there will be a sufficient number of connecting passengers from a large aircraft to justify delaying the departure

The following list represents the holding logic based on these values. Each dependent departure (i.e., each aircraft loaded and ready to pull back from its gate) considers the relative progress of each dependent arrival included in the mutual bank. The progress of the dependent arrivals is treated as follows:

- For a traveling flight (a flight traveling in the airspace/airport system but not currently at the gate), the departing flight is held if there are connecting passengers and the time

- required to land and transfer the passengers is less than the maximum hold time for the flight. Projected travel time is updated during the flight.
- For a queued flight (a flight queued for a gate but unable to unload), the departing flight is not held.
  - For a gated flight (a flight at the gate but not finished unloading), the departing flight is not held if there are no connecting passengers or if the remaining transfer time is greater than the maximum hold time for the flight.
  - The departing flight is held if there are connecting passengers and if the transfer time for the connecting passengers is less than the maximum hold time for the flight.
  - For a flight finished unloading, the departing flight is not held.

### **13.9 DSDPaths**

Dynamic Single Direction Paths (DSDPaths) are specified groups of connected ground links which may have multiple entry and exit points. DSDPath logic was implemented to allow a more realistic modeling of cul-de-sac and gate terminal traffic. Aircraft travel freely through the links with the restriction that no aircraft may pass another taxiing in an opposite direction. Each DSDPath has a specified capacity value to restrict the number of aircraft allowed on the path at a time. A capacity of one aircraft provides the same logic as the Metalink (see Metalink description above). If the capacity field is left blank or a 0 is specified, SIMMOD will not limit the capacity of the DSDPath.

## **14 Interface Logic**

The interface logic controls aircraft as they land or take off, making the transition from airspace to airport or from airport to airspace. Coordinating flights, routes, and runways involved in these transitions involves complex processes. This section discusses airspace procedures and their relationship to runway management. It describes how SIMMOD models procedures, landings, missed approaches, takeoffs, and various route and runway management issues. Related procedures, “mated” aircraft landings on parallel runways, and SIMMOD’s dynamic airspace re-routing capabilities are discussed in the latter part of this section. A landing aircraft requires a clear runway when it approaches within a certain distance of the airport. After touching down, it must then keep its path on the runway blocked to other aircraft until it has exited. The blocking prevents other aircraft from using the runway and may also limit the use of associated paths and runways. In SIMMOD’s airspace structure, a route usually ends or starts at an airport; overflights are an exception. If the airport is described in detail the route will be “connected” to its runway using an interface node. This node signals the simulation that an aircraft arriving via this airspace route will continue by landing at an airport, and that appropriate control measures must be coordinated. A flight departing from an airport will be scheduled to pass through an interface node at the end of the runway. This signals the simulation that the airspace portion of the model must coordinate with the airport in controlling the flight.

### **14.1 Procedures**

In addition to any other node characteristics, an interface node is defined to have procedures associated with it. The procedures, in turn, define time and distance restrictions necessary to

maintain a clear runway, plus any restrictions required to manage associated runways. A procedure is defined for every landing or takeoff that will be executed during the simulation. A procedure can be linked to other procedures to block other takeoffs or landings while the primary procedure is being executed. For example, a procedure may dictate the manner in which takeoffs and landings on the same runway exclude each other. An arriving aircraft coming within, say, two miles of the airport may block takeoffs from being released until the arrival is on the ground and has exited the runway. As soon as the landing has cleared, the takeoff is released. This takeoff might not only block landings while on the runway, it could also block all subsequent takeoffs until it was three miles away from the airport. A procedure constitutes permission for an aircraft to start a takeoff or landing. When an aircraft receives its permission, it blocks other aircraft from receiving theirs. The exact time and order for blocking and unblocking other aircraft determines the complex interactions required for runway management. The definition of any procedure includes two basic items of information: (1) the distance from the airport within which an aircraft will block runways and related procedures, and (2) the time interval after the start of an aircraft's runway roll within which the runway and related procedures must remain blocked. Additional information on these subjects appears under the description of Runway Blockage in Section 13, Airfield Logic.

### **Landing**

For a landing aircraft, coming within a given distance from the airport constitutes a request for permission to land and initiates the blocking of certain other aircraft while the landing is occurring. Different distances can signal different procedures and restrictions for other types of runway management. For example, an arrival may request a procedure, i.e., permission to land, when three miles out from an airport and start blocking takeoffs at two miles out. The release of blocking procedures varies according to the procedure. For example, a departure may be blocked from a runway for 60 seconds to account for the completion of a landing roll, while a departure on a crossing runway may be blocked for only 20 seconds to account for the landing aircraft passing by the intersection.

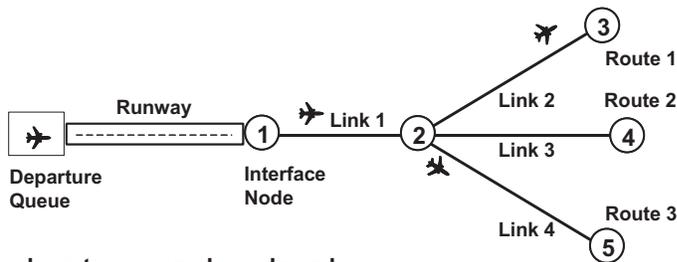
### **Missed Approach**

If a landing does not receive a clear procedure in response to a request, the missed approach logic is invoked. This may force the flight to pass the first interface node on a continuation route. The continuation route can circle back, perhaps reconnect with the route at some point, and allow the flight to try again; or it may divert the flight to another airport, or divert it to an exit in the airspace. If no continuation route exists, the aircraft will exit the simulation at that point. A low airport ceiling or too little runway visual range can also trigger a missed approach. Changes to weather conditions are considered in Section 15, Resetting Simulation Parameters.

### **Takeoff**

A takeoff is usually restricted by landings. Landings, however, are rarely restricted by takeoffs. Landing aircraft naturally have priority for using the runway, the separation between landing aircraft having already been determined to a large extent by airspace separation requirements. Arriving flights could continually block aircraft from taking off or crossing a runway based on the spacing of the landings, except that simulation parameters can be reset to establish a priority for runway crossing. This is also discussed under Runway Crossing Priority in Section 13, Airfield Logic. A takeoff does block successive takeoffs, and this serves to keep departing aircraft separated in the airspace. However, it is also possible to develop different departure

procedures that impose a sequence upon diverging departures. See Figure 14-1, Departure Control.



In this figure, three departure procedures have been defined for departures on routes 1, 2, and 3, respectively. For each pair of procedures, this establishes a blocking distance which will allow for the sequencing of takeoffs from one route to the next.

The takeoff procedure for a Route 1 flight blocks all other departures using this procedure (and route 1) until the aircraft reaches node 3. However, this same procedure only blocks the takeoff procedures on Route 2 and Route 3 until the aircraft using route 1 reaches node 2.

If a single procedure is used to control all departures, then the aircraft on Route 1 would block departures on all routes until it reached node 3.

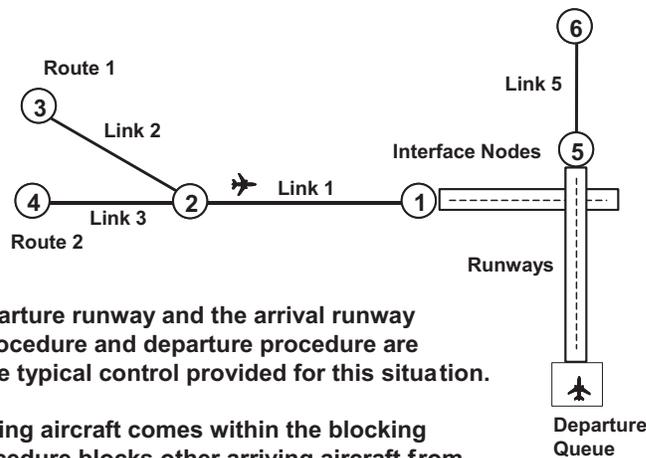
In the scenario given above, three departures queues — all located at the same geographic location — have been defined to allow the model to use separate procedures.

Figure 14-1: Departure Control

Takeoffs are also affected by other data values. Improperly set link capacities on the first airspace link, or mismatches between node separation requirements at the interface node can result in increased departure holding. Additionally, airspace separation requirements for aircraft on specific routes can be set. These are in addition to the node and wake separation requirements and will be enforced on sequentially departing aircraft onto the same airspace route.

### Related Procedures

Related procedures use either the same runway, or closely spaced, parallel, or crossing runways. Where there is potential for interference, procedures should be defined as related to give the simulation a chance to resolve conflicts. Relating procedures gives the simulation knowledge about how aircraft interact during airspace/airport transition. If two procedures are not related, their operations are treated as independent of each other and will be handled by the simulation without incurring any blockage. See Figure 14-2, Takeoff/Landing Control.



In this scenario, the departure runway and the arrival runway intersect. The arrival procedure and departure procedure are coordinated to model the typical control provided for this situation.

When the first approaching aircraft comes within the blocking distance, the arrival procedure blocks other arriving aircraft from executing an arrival approach until the first aircraft has reached the airport. More importantly, the arrival will block the departure runway, which will ensure that no departures are released until the landing aircraft has cleared the runway intersection.

Similarly, when the departing aircraft takes off, the departure procedure will block the arrival runway until the departure clears the runway intersection.

Figure 14-2: Takeoff/Landing Controls

## 14.2 Arrival Spreading

The separations of aircraft in an arrival flow should generally assume that the arrival runway is dedicated only to arrival operations. Consequently the intrail separations that are specified for the final nodes of an arrival route need only ensure that the arrivals are separated by the maximum runway occupancy time. This will ensure that a leading arrival will have sufficient time to exit the runway prior to a trailing arrival reaching the interface node. Of course, minimum wake turbulence separations must be maintained as well.

However, for runways with mixed operations, it is necessary that the arrival aircraft “spread” out to allow for departures. This is achieved by defining spread thresholds for the departure queue that will trigger a spreading of the arrival flow as well as specifying spread factors for the affected arrival interface nodes.

Each departure queue has two spread-related thresholds:

- Number of aircraft waiting. When the number of aircraft in the departure queue reaches this level, related arrivals will be spread.
- Waiting time. When an aircraft waits this amount of time (in seconds) in the departure queue, related arrivals will be spread.

When a departure queue activates spread, all the arrival procedures in the same related group as the affected departure procedure are notified that spread is in effect. Since each arrival procedure is associated with an arrival interface node, each affected arrival interface node is directed to impose spread.

Each interface node has three spread-related inputs:

- Duration. The amount of time that the spread remains in effect after it has been activated.
- Spread type. This indicates the way the spread value is determined. There are three spread types:
  - Percentage multiplier

- Increment value
- Absolute value
- Spread value. The value is interpreted depending on the spread type.
  - If the type is a percentage multiplier, the separation is product of the normal intrail separation and the percentage multiplier. So if you have 3.0 NM of separation and the multiplier is 2.0, arrivals will be separated  $3.0 \times 2.5 = 7.5$  NM when spread is active.
  - If the type is an increment value, the separation is increased by the amount of increment. So if you have 3.0 NM of separation and the increment is 4.0, arrivals will be separated 7.0 NM when spread is active.
  - If the type is an absolute value, the separation is the value specified and the normal separation is disregarded until spread is no longer active.

### 14.3 Staggered Runway Approaches

Instrument approaches on parallel runways that are separated by 2,500 to 4,300 feet may be conducted as long as diagonal separation of two nautical miles is maintained between adjacent aircraft. Additionally, simultaneous approaches to converging or intersecting runways (runways whose centerlines or extended centerlines intersect) are presently authorized during Visual Meteorological Conditions (VMC). Both types of approaches require coordination and control to interleave the arrivals. The logic of the staggering option emulates the decision process of the ATC controller as aircraft are vectored onto final approach. To obtain proper spacing and sequencing in SIMMOD, the staggering of arrivals is exercised over the airspace links prior to the stagger control nodes. A step-by-step overview of the staggering logic is presented in the following paragraphs.

Each stagger control node may have one or more other stagger control nodes associated with it, and a time separation array is defined for each such node pairing and for each aircraft group. This will be explained more fully below. By selectively grouping the stagger control nodes, different runway approaches may be “related” to each other, independently of other stagger control node groups.

The distance between each interface node and the node before its associated stagger control node should be approximately equal within each related stagger control group. Actually, the nominal travel times between these nodes should be nearly equal. Conversely, if there is a large discrepancy between these distances (or nominal travel times), then wildly erratic separations may result.

As an aircraft travels along its route toward an arrival runway, each airspace node that it passes through is examined to see if the next node has been designated as stagger control node. If not, then the simulation controls the aircraft's movement to the next node along the route in the usual manner. But whenever a pre-stagger control node is encountered, the stagger logic is called into play at that point to impose additional delay on the aircraft, if necessary, so as to separate it from other aircraft being staggered within the same related group.

Stagger control logic is imposed on any individual aircraft once, and only once, at the time it passes through the pre-stagger control node of its route. Once the aircraft begins traveling away from the pre-stagger control node, it is no longer subject to further action by the stagger separation logic. In essence, the model looks at the latest (anticipated) time of arrival at the airport interface node for all other aircraft that have already passed through their respective pre-

stagger control nodes (within the related group); and from that (latest) time, it determines the earliest time that this aircraft may be allowed to arrive at its own airport inter node (considering the inter-type time separations that have been specified in the stagger inputs).

Because a stagger controlled aircraft is never allowed to reach its own arrival interface node earlier than the anticipated time of arrival of a related stagger control aircraft, the travel times and distances should be nearly the same within a related group. For example, even if the aircraft being considered could reach its own interface node well before all other aircraft already scheduled to arrive at their interface node, the “newcomer” will be held at his pre-stagger control node long enough to ensure that his (anticipated) interface node arrival is sufficiently later than all others. This is a limitation of the current stagger methodology, but once the limitation is understood it should not be unduly restricting.

Another way to look at SIMMOD’s stagger logic is that each subsequent interface node arrival is always scheduled so that it will be properly separated, later in time, from all other presently anticipated interface node arrivals within its related stagger group. The stagger separation is always (and only) accomplished by imposing additional delay (as necessary) on an aircraft as it passes through its own re-stagger control node. In this manner, arrivals on crossing runways (or even arrivals at different airports) can be “staggered” as desired, without having to create “mated” final approach links or having to define overly complicated arrival procedures.

This entire approach relies on the ability of the model to accurately predict the amount of time that will be required for each aircraft to travel along the links between its pre-stagger control node and its airport interface node (considering proper separation from the aircraft, if any, preceding it along the same sequence of links). Anything that causes the actual time of arrival at the interface node to differ from the anticipated time of arrival (calculated at the instant of departure from the stagger control node) will negate the effectiveness of this methodology.

Another limitation of the stagger control logic is that each “node mate” defined for a stagger control node must itself also be defined as a stagger control node. Any such node mate that is not a stagger control node will have no effect.

#### **14.4 Dynamic Airspace Re-routing Due to Runway Congestion**

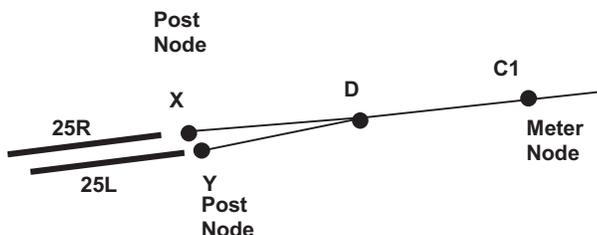
Given (a) the appropriate metering logic to look ahead to the primary route’s interface node and (b) a plan that incorporates a transitional route to another interface node, the simulation will try to re-route an aircraft in the airspace if the queue for its interface node is above capacity. Each route is allowed to define one or more transitional (i.e., re-routing) routes to another interface node. The metering control logic is used to check the flow and capacity at interface nodes to determine the need for re-routing. This requires that the interface nodes be meter post nodes. (Refer to Section 12, Airspace Logic, for information on metering.) The primary meter post node queue has a capacity; when that is exceeded the model checks the transitional route.

Transitional routes are defined by setting up one plan specifically to handle re-routing. The user must also specify the plan used for re-routing the traffic in the global data record `DIVERSION` `PLAN`. If this record doesn’t exist, you must create it. If there is sufficient capacity at the alternate metering post node (typically an interface node) to which the aircraft would be re-routed, and if the aircraft is not beyond the transition point for the transitional route, then the aircraft will be re-routed, i.e., it will take the transitional route.

The following example describes the application of metering logic:

- Define C1 as a meter node.
- Define X and Y as a post nodes. These nodes must be interface nodes.
- Define a reroute plan.

- Define the re-route transitions.
- In this example, the `MAX_QUEUE_SIZE` is 1 for Y. This means that if 1 aircraft is flying from C1 to Y, the next aircraft will re-route to X. The aircraft will alternate between the two runways.



**Figure 14-3: Terminal Rerouting Airspace**

This example is a continuation of the metering example shown in Figure 12-18. It is possible to combine both metering and dynamic re-routing in the same arrival flows to an airport.

## 14.5 Dynamic Departure Re-routing Due to Runway Congestion

When a flight is about to depart from its gate for its departure queue, a determination is made as to the departure queue length. If the length is greater than the departure queue's re-routing threshold, the simulation will attempt to reroute the flight. Re-routing will occur if a diversion route to the flight's assigned route has been specified in Dynamic Airspace Re-routing (as well as in the global data record `DIVERSION_PLAN`), and if the diversion route's corresponding departure queue's length does not exceed its maximum length for accepting diverted flights. Since the diversion is on the ground, the movement to the alternative queue is handled by the simulation taxipath optimization utility.

## 14.6 Touch-and-Go Operations

Touch-and-go patterns can be modeled using SIMMOD. First the user must create a touch-and-go set using the `TNGSET` record of the Airspace file, or by using the following menu path in the Network Builder: `Edit:Airspace:Route: TOUCH_AND_GO_SET`. A touch-and-go set consists of one or more patterns (sets of airspace nodes) which may be flown one or more times. The touch-and-go set must be assigned to an arrival flight in order for that flight to travel the touch-and-go patterns. The individual touch-and-go patterns must be created for each touch-and-go set. The touch-and-go patterns are defined with the `PATTERN` record of the Airspace file, or by the following menu path of the Network Builder: `Edit:Airspace:Route:TOUCH_AND_GO`. The touch-and-go pattern consists of a list of nodes and a threshold capacity value. Aircraft are allowed to travel the touch-and-go pattern only if the number of aircraft traveling the pattern (and traveling on a list of associated routes) is below this threshold value.

## 15 Resetting Simulation Parameters

The conditions initially defined for a simulation may not hold true for the entire duration of the simulation. Some airports, for example, use the same runway for landings at one time and takeoffs at another. Similarly, a reversal of wind direction may require a significant change in plans. SIMMOD provides a means to model such changes.

When defining the simulation, it is possible to trigger changes to some of the conditions initially established for the airport and airspace system. This is accomplished with user-defined external

events, usually just called “EVENTS” in the definition of input data. (External events are discussed in Section 8, How SIMMOD Works.)

The user defines each such event and the time at which it is to occur in the simulation. These changes cannot be triggered by another condition, e.g., the length of a holding queue.

### **15.1 Flight Cloning (SETCLONE)**

This is described in Section 0, Flights.

### **15.2 Changing Gate Characteristics (SETGATE)**

The gate characteristics that may be changed include:

- Capacity
- List of airlines permitted to use the gate
- Arrival unloading time distribution
- Departure boarding time distribution

### **15.3 Inhibiting Injections (SETINJ)**

This feature essentially halts the creation of new arrival air traffic. All injections of aircraft into the airspace during the simulation are inhibited, i.e., annulled, for set periods of time. Using this single control can thus produce the same results as much more extensive changes to the arrival and departure data. The arrival and departure schedule for the time period is removed from the event schedule.

### **15.4 Changing Link Characteristics (SETLINK)**

The link characteristics that may be changed include:

- Length
- Heading
- Overtake flag
- Wake turbulence flag
- Capacity
- Vectoring delay

Links are typically changed because of increased congestion occurring at certain times of the day or because of other changes in airport operations.

### **15.5 Changing Meter Post Node Characteristics (SETMETER)**

There is only one characteristic for a meter post node: the intrail separation for the node under metering logic.

### **15.6 Changing Node Characteristics (SETNODE)**

The node characteristics that may be changed include:

- Node arrival control strategy
- Arrival strategy option flag
- Intrail separation
- Holding capacity
- Holding strategy
- Holding stack type

Nodes are typically changed because of increased congestion occurring at certain times of the day or because of other changes in airport operations.

### **15.7 Changing Plans (SETPLAN)**

An initial plan must be defined with the SETPLAN event before any flights occur. You are allowed one or more plan changes during the course of the simulation.

A plan change allows the operations of the simulation to change. The new plan may change airspace routes and divert aircraft to transitional or new routes.

Routes may be changed selectively. The plan change transition logic allows each route to have a transition path for aircraft already on the route. It also designates a “point of no return” beyond which a flight cannot make the transition. Flights beyond this point must finish on the original route. In addition, departures can be held until all aircraft in the airspace have either completed their transitions or exited the airspace.

An optional value of the SETPLAN record, PrePlanTime, may be used to transition departures to the new departure queue before the plan changes. Departure flights waiting in the old departure queue when the new plan takes effect will not be transitioned to the new departure queue by SIMMOD.

### **15.8 Inhibiting Procedures (SETPROC)**

Inhibiting procedures during the simulation blocks certain procedures from being used. Starting from the set simulation time, this feature prevents certain normally scheduled operations (designated by the user) from occurring. This generally results in missed approaches in the airspace and causes aircraft to be queued on the airfield at departure queues. This option may be used by the analyst to investigate a specific situation, such as when a runway is blocked by a stalled or damaged aircraft.

### **15.9 Changing Route Characteristics (SETROUTE)**

The only characteristic of a route that changes is its intrail separation. This can also be defined by plan for each route.

### **15.10 Changing Runway Characteristics (SETRWY)**

The runway characteristics that may be changed include:

- Landing roll distribution
- Takeoff roll distribution
- Taxi prohibitions when active

### **15.11 Changing Sector Characteristics (SETSECT)**

The only sector characteristic that changes is its capacity.

### **15.12 Controlling Touch-and-Go (SETTNG)**

This record provides the capability to control touch-and-go during the simulation. When the touch-and-go flag is on, the aircraft will perform touch-and-go, and will be removed from the simulation at the eject node of the touch-and-go pattern after completion of the touch-and-go.

### **15.13 Changing the Wind Characteristics (SETWIND)**

Wind conditions can be changed at any time during a simulation. Either the heading or speed or both can change.

### **15.14 Changing the Weather (SETWX)**

A change in the simulated weather involves changes to the airport ceiling or runway visual range. These can affect the landing and takeoff procedures and cause missed approaches. A change in the weather does not increase separations or spacing to account for weather conditions.

### **15.15 Setting Conditions for a Forced Runway Crossing (SETXNG)**

Setting a crossing resets the simulation to allow a flight to cross a runway when arriving flights would otherwise block the runway and all paths crossing it. Based on user-defined parameters, the simulation adjusts the intrail separation of arriving flights to create a break in the incoming traffic; this in turn creates an opening to cross the runway.

When the simulation is set to allow runway crossing, it monitors the number of aircraft waiting to cross and the period each waiting aircraft has been delayed. If a certain number of aircraft are waiting to cross, or if at least one aircraft has been waiting a certain amount of time, the simulation will force a break to allow a runway crossing.

## **16 Stochastic Processes**

SIMMOD is a stochastic model. Stochastic processes, in the form of random variables, are introduced into every iteration of SIMMOD to produce unique output representing day-to-day variations in air traffic phenomena. The amount of variation is determined by the input data. Because SIMMOD is designed to produce realistic — not “ideal” — results from any single iteration of a defined application data set, it is often necessary to run several iterations with a single data set in order to establish statistically significant tendencies. For a run of several consecutive iterations, the SIMMOD Report Post-processor will produce aggregate values and, where appropriate, averages and standard deviations. For example, where frequencies indicating the usage of a given facility are reported for a single-iteration run, average frequencies will be reported for a multiple-iteration run.

### **16.1 Random Linear Variables**

SIMMOD uses random linear variables to reproduce the effects of random variation in airport and airspace system phenomena. The lateness of any given arrival or departure, for example, is introduced into the simulation processes as a random linear variable.

The simulation generates a random real number between zero and one. It then uses this number to draw from cumulative distributions to determine the values used in the simulation.

Distributions can be delineated piecewise using pairs of numbers. The first number in the pair represents  $P(x)$ , the probability of a value less than or equal to  $x$  occurring; the second defines the corresponding value  $x$ .

By defining at least two such pairs, one indicating zero percent probability for the lower value in the distribution range and one indicating 100 percent probability for the upper value, an analyst can describe a basic linear function. Additional pairs allow more detailed specification of a distribution function. This approach to defining linear functions facilitates description of empirical data distributions. See Figure 16-1, Random Linear Variables.

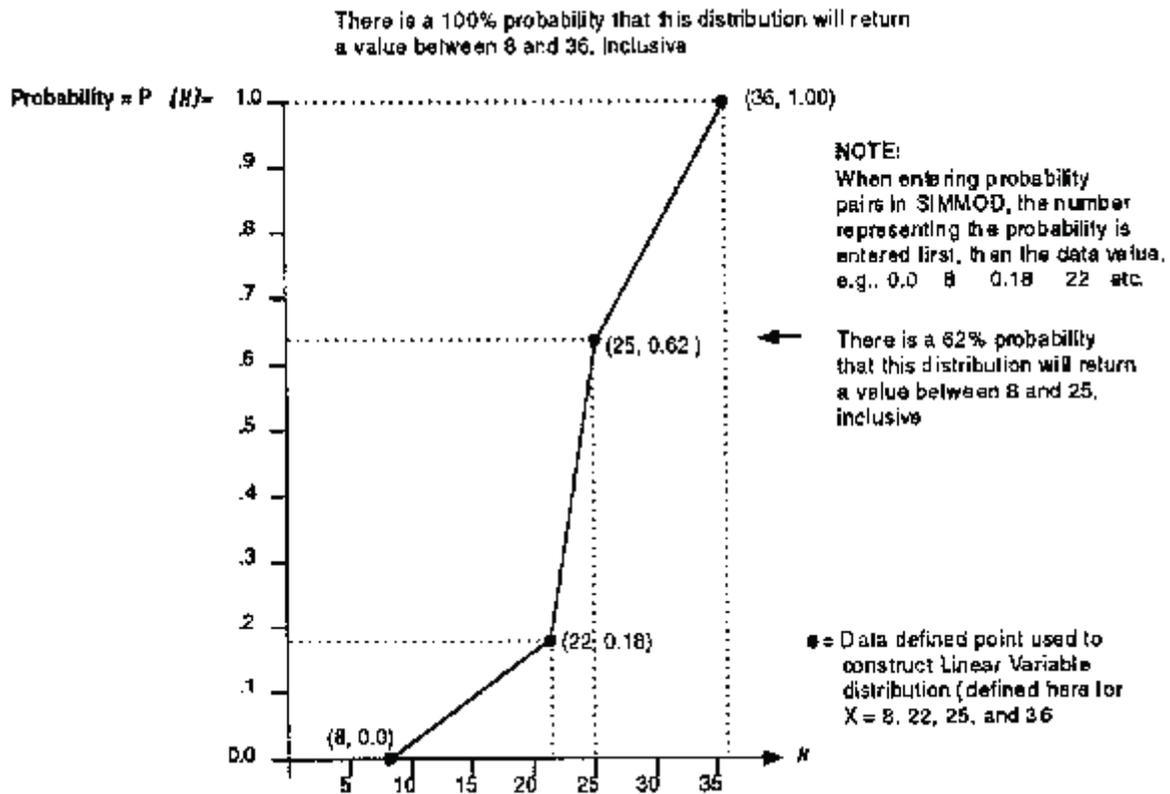


Figure 16-1: Cumulative Random Linear Variables

## 16.2 Random Number Streams and Seeds

When SIMMOD returns a value for a random linear variable to be used in the simulation, it does so by performing a function on a random real number between zero and one. Because many such random numbers are used in every run of a SIMMOD application data set, the simulation must create a sequence or "stream" of many random numbers for each iteration. These random number streams are created by a random number generator built into SIMSCRIPT II.5, the language in which SIMMOD is programmed.

Traditionally, the first number in a random number stream is called the "seed" of the sequence. This seed is used by the random number generator to produce the ensuing stream.

Starting with the same seed, the random number generator will always produce exactly the same random number stream (assuming that the simulation is run on a machine with the same processor). To this extent, then, the randomization process is controllable. This is significant because it allows the analyst to reproduce the simulation results achieved with a given data set.

As with all SIMSCRIPT II.5 programs, SIMMOD is initialized with ten random number streams, each consisting of a single random number seed. These ten streams are used within the simulation as follows:

- Stream 1 - Gate selection and occupancy time generation
- Stream 2 - Multiple arrival generation
- Stream 3 - Multiple departure generation
- Stream 4 - Takeoff and landing roll distance generation
- Stream 5 - Inter-aircraft separation multiplier generation
- Stream 6 - Arrival and departure lateness generation
- Stream 7 - Bank late flight holding probability generation

- Stream 8 - Bank late transfer time generation
- Stream 9 - Arrival and departure clone generation
- Stream 10 - PC version initial color display

Unless the analyst changes the seeds provided with SIMMOD, these original seeds will be used to generate numbers for the first iteration in every run of an application. At the end of the first iteration in a run, the last random number created in each stream is written to the Standard Report (the RPTSTD file) as output.

If the run does not end with one iteration, however, these last random numbers will become the seeds in the next iteration's random number streams. And at the end of that iteration, the last random number in each stream will again be written to the Standard Report. If yet another iteration is to follow, these last random numbers will become that next iteration's seeds, and so on and so on, until no iterations follow.

Different seeds produce different streams, and therefore different results from an iteration. If the analyst wishes to isolate the results of the second or third iteration in a specific run, it will be necessary to re-run that application data set using exactly the same random number stream seeds supplied for the iteration. As mentioned above, a record of these seeds is included in the Standard Report for the original multi-iteration run.

The ten random number streams listed above are all assigned to supply numbers for specific variables. These variables are discussed individually below.

## **16.3 Stochastic Process Descriptions**

### **Gate Service (Occupancy) Times for Arrivals and Departures**

Each flight has a gate occupancy time, also known as a gate service time. This represents the period of time required to perform common gate operations for an aircraft, notably loading and unloading. The amount of time will depend on the type of flight and the type of aircraft represented.

After an arriving flight arrives at its gate, it requires a gate occupancy time to unload passengers. After a departing aircraft is created at the gate, it needs a gate occupancy time to load passengers. The time at which departures are created should be early enough to allow for this loading time. Both the loading and unloading time distribution can be set to zero if not needed.

### **Multiple Arrivals and Departures**

Multiple arrivals and departures allow aircraft to be created without defining each flight individually. The characteristics of the flights will be the same but the time at which they are created is different. A single multiple arrival definition will create a data defined number of flights over a data defined time interval. The simulation determines a random start time for each flight created within the time interval.

### **Landing and Takeoff Roll Distances**

Landing and takeoff roll distances used in the simulation are based on observed probabilities that are translated into cumulative distributions. These probabilities are linked to aircraft type.

### **Intrail (Inter-Aircraft) Separation Multiplier**

The minimum intrail separation between a given aircraft and the succeeding aircraft due to the wake vortex requirements on the link is normally based on the model types of the leading and

trailing aircraft. The Intrail Separation Multiplier allows the analyst to adjust the separation value assigned to an aircraft and thereby to approximate more closely the actual behavior of aircraft.

### **Lateness of Scheduled Flights**

A lateness time can be added to a flight arriving or being created to account for delay not otherwise modeled. The amount of lateness (in minutes) is drawn from a cumulative probability distribution. The time must be a positive delay; a negative delay will cause an error because the simulation cannot move backwards in time. This lateness is not considered as a delay for the delay statistics.

### **Bank Holding Probability and Transfer Time**

Three distributions from two random number streams are used for bank logic.

One determines the probable time required for a passenger to transfer planes if a flight has been held for them. The transfer time is more specifically defined as the interval starting with the late flight's arrival at the gate and ending when the flight being held can be released to depart from the gate.

The other two distributions determine whether there are enough connecting passengers to justify holding the departure of an aircraft in the bank. One distribution is for passengers from a heavy aircraft, the other is for passengers from a large aircraft.

If there are enough connecting passengers from an aircraft to justify holding a departure, the model tests to see if the passenger transfer time would cause the flight to hold for more than its maximum delay time. If the maximum delay would be exceeded, the justification for holding is annulled and the departing aircraft reverts to its previous departure time.

If there are not enough passengers to justify holding, the departure time remains unchanged.

### **Arrival and Departure Clone Generation**

Arrival and departure clone generation is described in Section 0, Flights.

# APPENDIX B – Generic Metroplex Scheduling Study



## Introduction

---

The goal of the Generic Metroplex scheduling study is to compare optimized arrival and departure scheduling to baseline TMA scheduling using simplified “generic” metroplexes whose characteristics can be easily varied, while observing the resulting effects on the scheduling benefits. In our previous Metroplex NRA, we used a single Generic Metroplex runway layout, consisting of two airports having one arrival runway and one departure runway each, and having all runways parallel to each other. The objective of this report is to expand this to a set of Generic Metroplexes which is still simple enough to be easily analyzed, but which better represents the geometric relationships present in existing NAS metroplexes. To this end, we maintain the condition of having only two airports, with one arrival and one departure runway each, while the condition that all runways be parallel is lifted. Given these conditions, a set of runway layouts is developed by analyzing the runway geometries of current metroplexes. Using this set of runway layouts, the complete set of Generic Metroplex Airspace Geometries is constructed containing arrival/departure routes corresponding to either single or double cornerpost fix geometries (Geometries 1 and 3 from our previous Metroplex NRA). Finally, we present two sets of comparative analyses using these Generic Metroplexes. The first computes volume intersections of arrival and departure flows to compare the airspace complexity for each Generic Metroplex using a variety of airspace geometries. Finally, fuel burn and emissions metrics are computed for each Generic Metroplex with each Airspace Geometry (using a single input set of arrival/departure demand data) in order to compare the relative environmental impact of these alternative airspace designs. Metroplex Runway Layout Spanning Set

### Spanning Set Development Methodology

To derive a spanning set of runway layouts, we first developed a set of two-airport runway configurations directly from existing metroplexes. All configurations used in each of the 77 ASPM airports for 2006-2010 were the data source for this analysis.

Eight Metroplexes having at least one ASPM airport were represented in the ASPM data:

- New York: JFK, LGA, EWR, TEB
- San Francisco: SFO, OAK, SJC
- Dallas: DFW, DAL
- Los Angeles: LAX, VNY, BUR, SNA, ONT, LGB
- Miami: MIA, FLL
- Balt/Wash: IAD, DCA, BWI
- Chicago: ORD, MDW, GYY
- Houston: IAH, HOU

From this list, nine two-airport “metroplexes” were identified:

- New York1: JFK, LGA
- New York2: EWR, TEB
- San Francisco: SFO, OAK
- Dallas: DFW, DAL
- Los Angeles: LAX, LGB
- Miami: MIA, FLL

- Balt/Wash: IAD, BWI
- Chicago: ORD, MDW
- Houston: IAH, HOU

For each of the two-airport metroplexes listed above, the most-used single arrival runway and single departure runway configurations were determined from the ASPM Hourly database.

Methods considered for obtaining these configurations include:

1. Determine most-used configuration at each airport by total operations count or total hours count, choose representative arrival and departure runways.
2. For each runway at an airport, find “runway value”: sum hours (or operations) counts for all configurations in which this runway appears as an arrival (or departure) runway, take runway having maximal value.
3. Combination of the two approaches above: find most-used configuration (using operations or hours counts), then identify representative arrival and departure runways informed by runway value.

The approach used for this analysis is #3 above; as it turned out the most-used configurations identified were the same using either operations counts or hour’s counts. Results were obtained for each calendar year 2006-2010. The configurations used were chosen from the recent years: 2008 and 2009. As it turned out, the only significant differences in most-used 2-runway configurations between these two years were at LGA, ORD and MDW.

Using this set of runway layouts, the geometric relationships of the arrival and departure runways within each airport, and the two airports to each other were analyzed. Each runway layout in the spanning set is described using a set of parameters which completely defines the geometry of each generic metroplex runway layout. Finally, metroplexes having similar runway layout geometries by visual inspection are grouped, and a representative metroplex for each group is given.

### Most-Used Full and 2-Runway Metroplex Configurations

In this section the most-used configuration for each of the nine 2-airport metroplexes is presented. In the cases where these configurations differ significantly between 2008 and 2009, results from both years are shown. Finally, for each metroplex, the simplified configuration is given, consisting of one arrival runway and one departure runway in each airport. In all of the images below, runway lengths are shown at 5 times their actual length, and the runway usage is color coded according to the following legend:



Figure 1: Color-coding of runway usage

## Configurations for New York Metroplex 1: JFK, LGA

The most-used configurations for the New York Metroplex 1 in 2008 and 2009 differ in the flow directions and usage at LGA. The simplified runway layout for this metroplex follows the 2009 configuration.

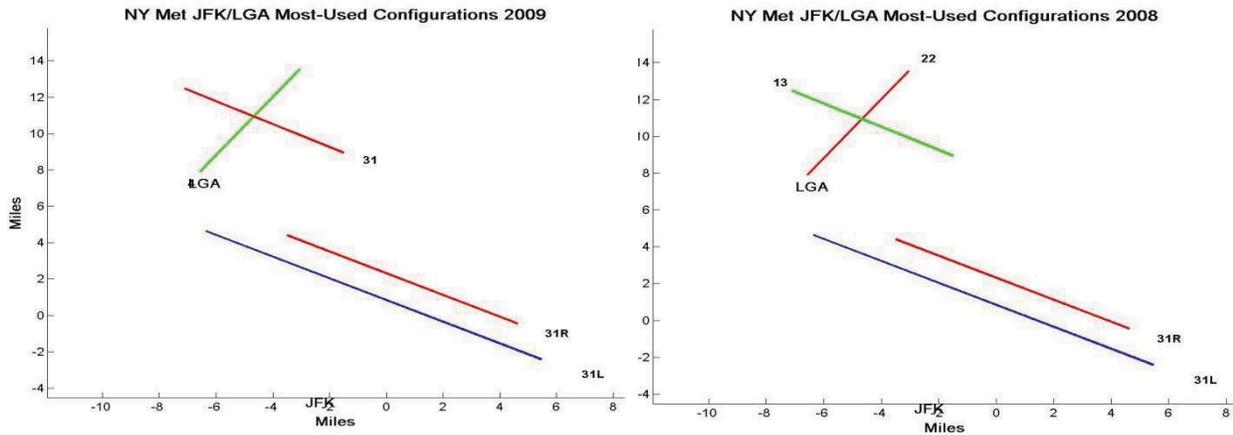


Figure 2: JFK/LGA Most-Used Configurations for 2008 and 2009

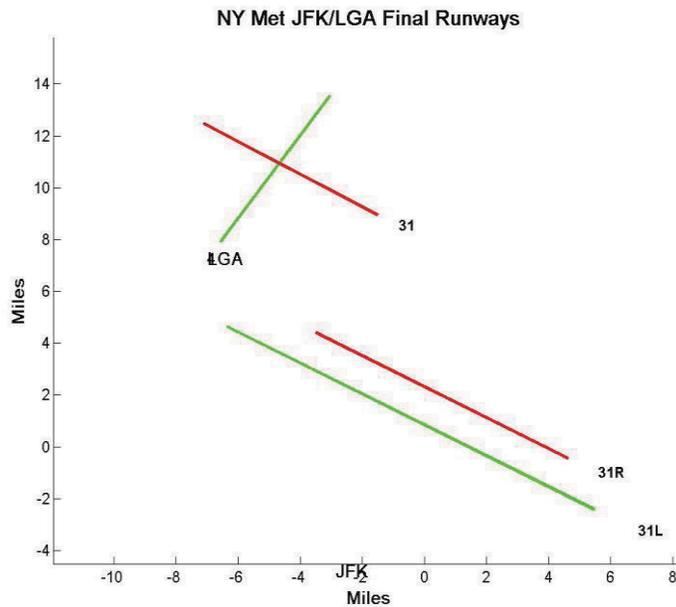


Figure 3: JFK/LGA Simplified Configuration

## Configurations for New York Metroplex 2: EWR, TEB

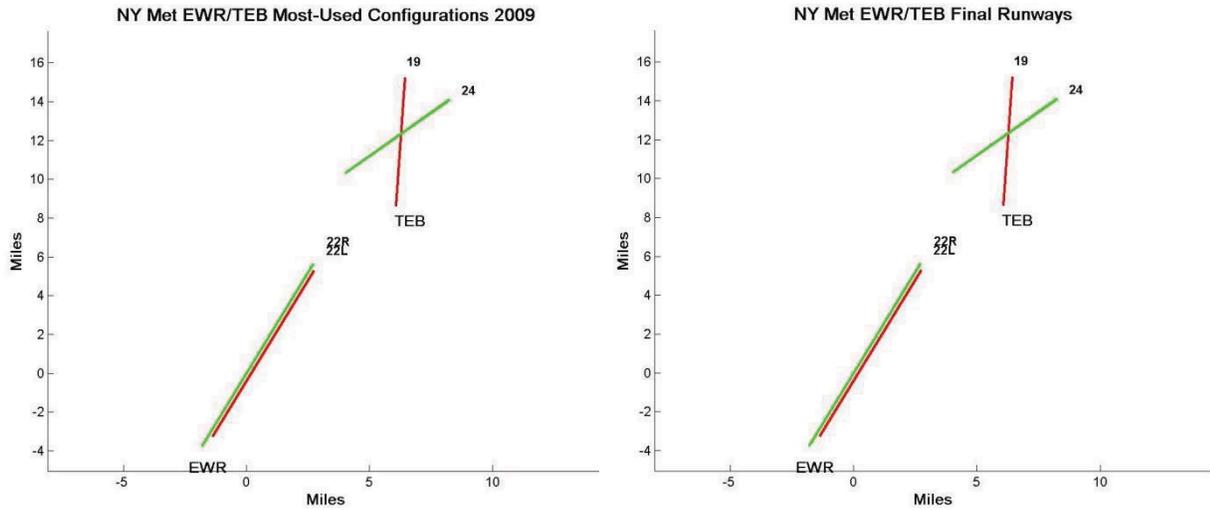


Figure 4: EWR/TEB Most-Used and Simplified Configurations

## Configurations for San Francisco Metroplex: SFO, OAK

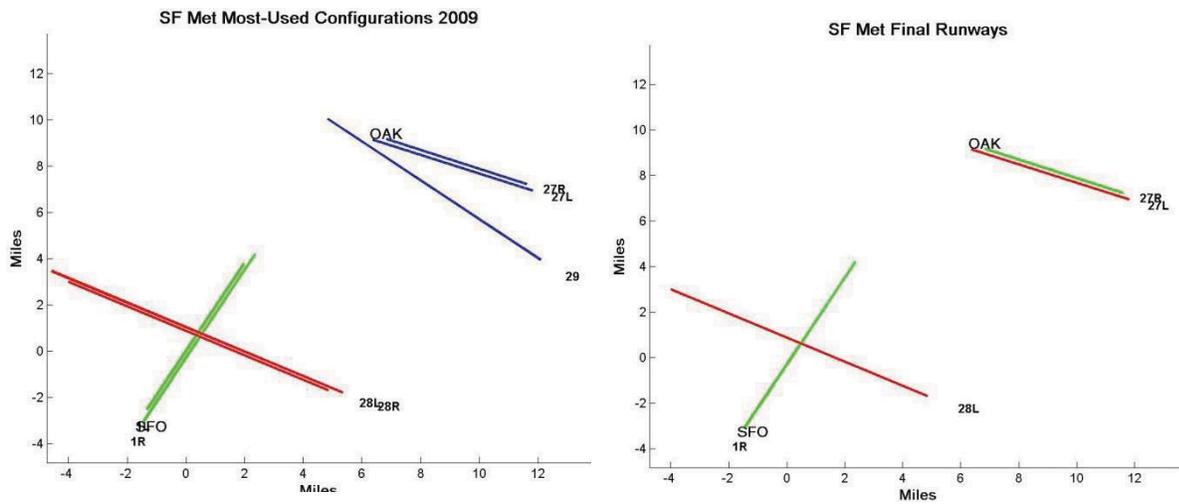


Figure 5: SFO/OAK Most-Used and Simplified Configurations

## Configurations for Dallas Metroplex: DFW, DAL

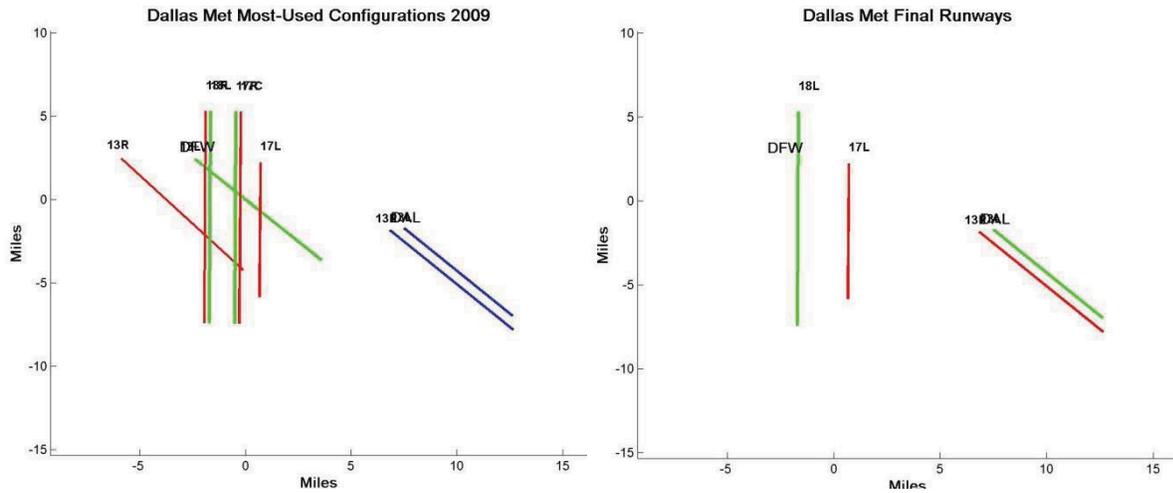


Figure 6: DFW/DAL Most-Used and Simplified Configurations

## Configurations for Los Angeles Metroplex: LAX, LGB

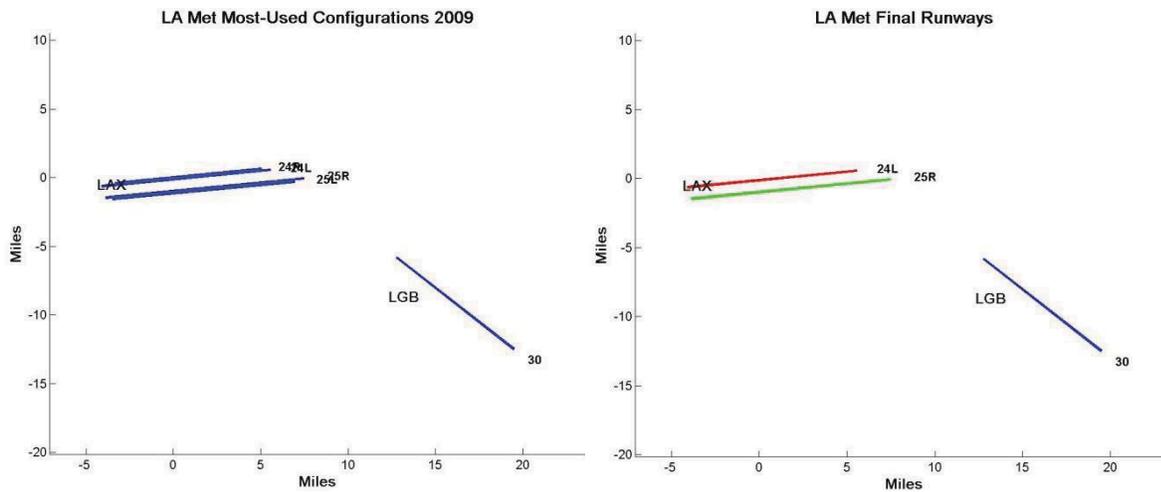


Figure 7: LAX/LGB Most-Used and Simplified Configurations

## Configurations for Miami Metroplex: MIA, FLL

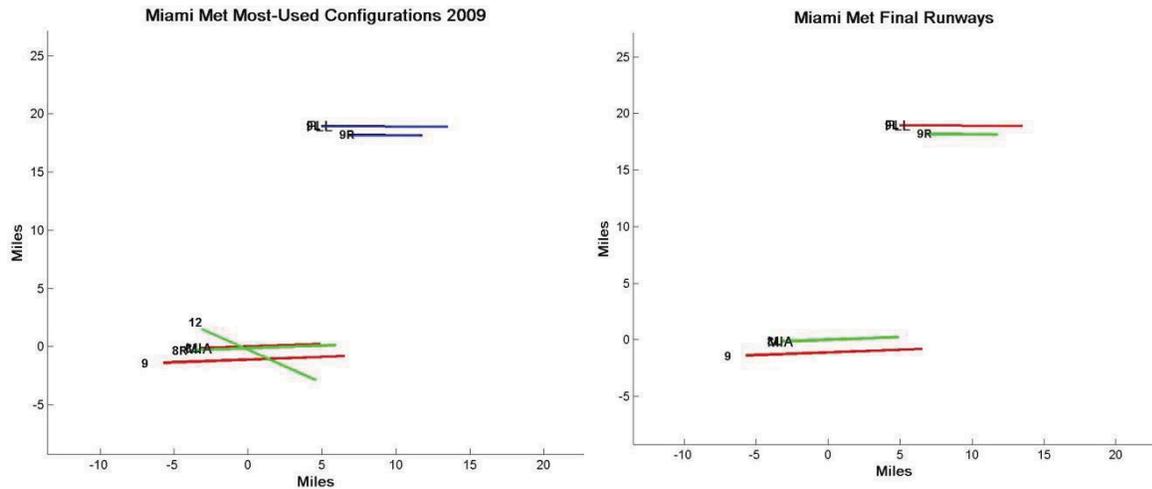


Figure 8: MIA/FLL Most-Used and Simplified Configurations

## Configurations for DC Metroplex: IAD, DCA

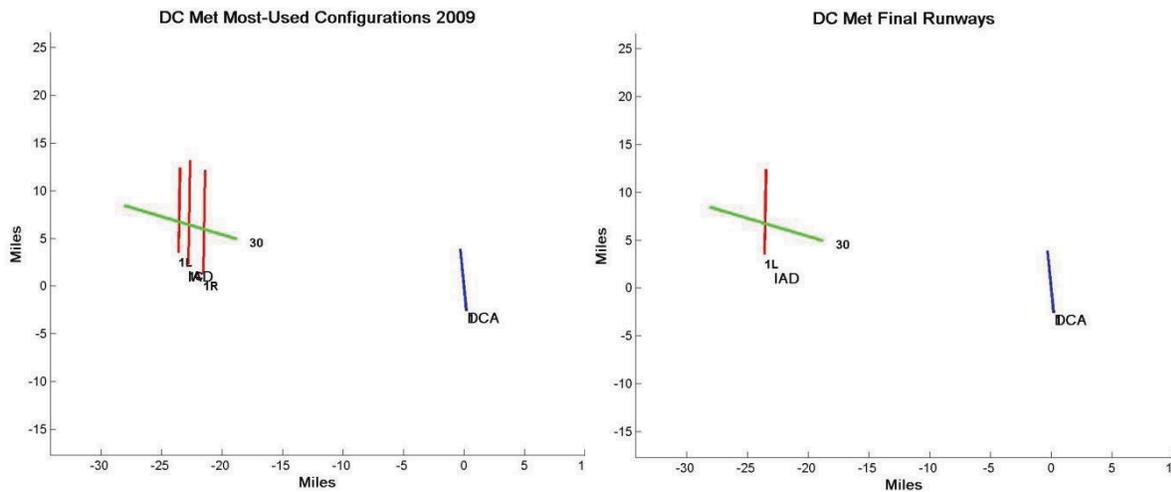


Figure 9: IAD/DCA Most-Used and Simplified Configurations

### Configurations for Chicago Metroplex: ORD, MDW

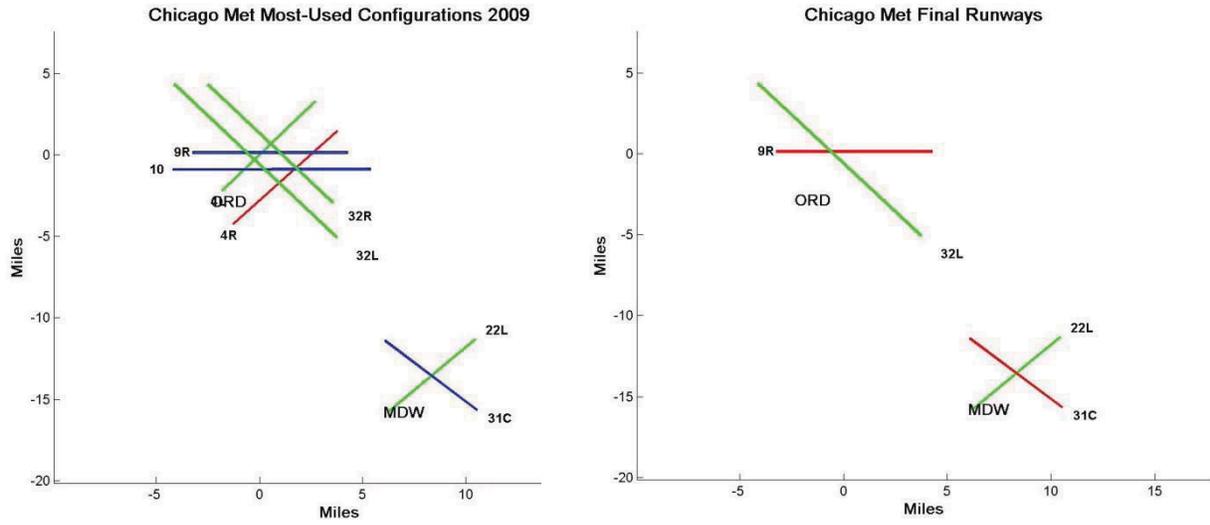


Figure 10: ORD/MDW Most-Used and Simplified Configurations

### Configurations for Houston Metroplex: IAH, HOU

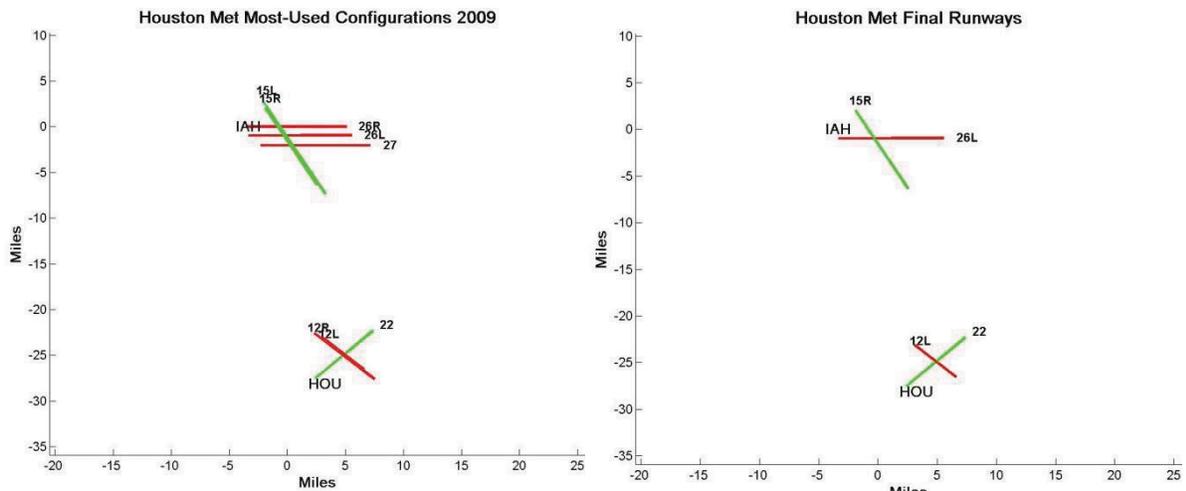


Figure 11: IAH/HOU Most-Used and Simplified Configurations

# Parameterization of Runway Layout Geometries

## Full Set of Parameter Definitions for Runway Layout Geometries

The following figure displays the full set of parameters which could be used to completely describe an arbitrary metroplex runway layout geometry. Our current case is much simpler, since we only have two airports, with one arrival and one departure runway in each. The following figures identify the subset of notation which is sufficient to describe the runway layout geometries in our case. Finally, all of these values for the metroplexes identified above are given.

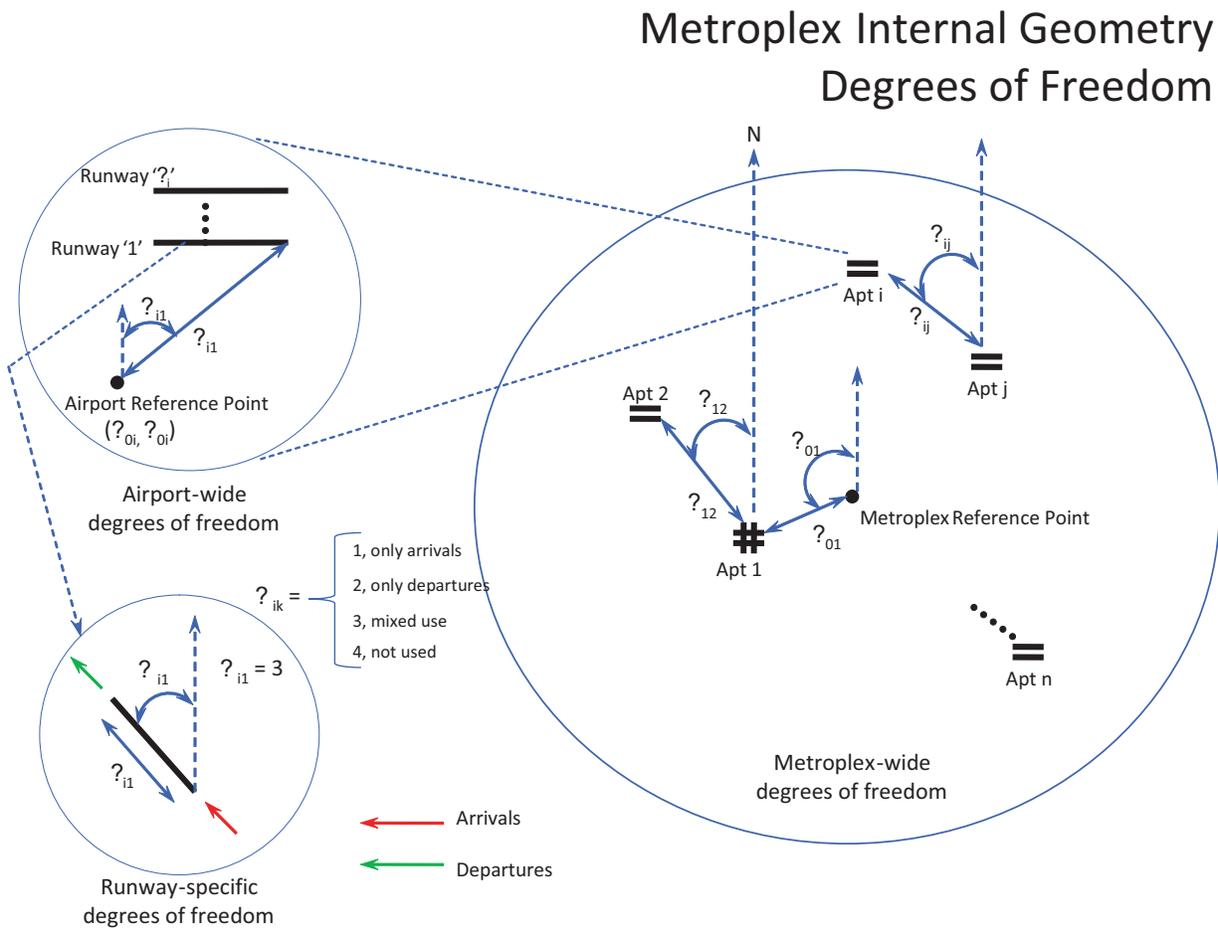


Figure 12: Aditya Saraf's Parameter Definitions for Metroplex Runway Layouts

## Reduced Set of Parameter Definitions for Two-Airport Geometries

### Two-Airport Metroplex Parameters

#### Airport-to-Airport

$(\delta_{im}, \theta_{im})$  = Location of airport m wrt airport i (distance, orientation)

Identify each airport's reference point with its 1<sup>st</sup> runway start point.

Identify Metroplex reference point (i=0) with starting point of 1<sup>st</sup> airport's 1<sup>st</sup> runway.

(So our only case becomes i=1, m=2.)

#### Airport-to-Runway

$(\rho_{ik}, \sigma_{ik})$  = Location of k<sup>th</sup> runway start wrt airport i reference point (distance, orientation)

#### Runway

$(\lambda_{ik}, \phi_{ik}, \mu_{ik})$  = (Length, orientation, operations) for k<sup>th</sup> runway at airport i.

Figure 13: Parameters used to describe Metroplex Runway Layout Geometries in the two-airport case.

### Airport-to-Airport Location

metName	delta	theta
NY Met JFK/LGA	10.9	323
SF Met	12.6	50
Dallas Met	10.3	112
LA Met	18.1	123
Miami Met	20.7	24
DC Met	24.6	287
Chicago Met	16	146
Houston Met	23.7	172
NY Met EWR/TEB	12	27

Figure 14: Parameterization of Metroplex Airport-to-Airport Geometries

		<u>A-to-R</u>		<u>Runway1</u>			<u>Runway2</u>		
metName	airport1	rho	sigma	mu1	lambda1	phi1	mu2	lambda2	phi2
NY Met JFK/LGA	JFK	6697	211	Arr	9979	310	Dep	14541	310
SF Met	SFO	6703	75	Dep	8650	10	Arr	10580	280
Dallas Met	DFW	10219	277	Dep	8996	130	Arr	13427	180
LA Met	LAX	7615	120	Arr	10256	240	Dep	12056	250
Miami Met	MIA	7517	215	Dep	8580	80	Arr	12968	90
DC Met	DCA	0	90	Mix	6876	10	Mix	6876	10
Chicago Met	ORD	8723	150	Arr	7940	90	Dep	12984	320
Houston Met	IAH	11591	260	Arr	9377	260	Dep	10012	150
NY Met EWR/TEB	EWR	1010	96	Dep	10999	220	Arr	9999	220
metName	airport2	rho	sigma	mu1	lambda1	phi1	mu2	lambda2	phi2
NY Met JFK/LGA	LGA	7541	82	Dep	7002	40	Arr	6992	310
SF Met	OAK	1000	202	Dep	5446	270	Arr	6203	270
Dallas Met	DAL	2976	223	Dep	7751	130	Arr	8799	130
LA Met	LGB	0	90	Mix	9997	300	Mix	9997	300
Miami Met	FLL	4492	153	Arr	8989	90	Dep	5269	90
DC Met	IAD	6770	128	Arr	9409	10	Dep	10471	300
Chicago Met	MDW	4654	182	Dep	6430	220	Arr	6510	310
Houston Met	HOU	4827	263	Dep	7594	220	Arr	5144	120
NY Met EWR/TEB	TEB	1341	92	Arr	7004	190	Dep	6006	240

Figure 15: Parameterization of Metroplex Airport-to-Runway and Runway Geometries

## Results: Spanning Set of Metroplex Runway Layouts

The nine simplified metroplex configurations are combined into six groups below, according to their geometric similarity. The groups are as follows, with the chosen representative metroplex listed first in each group.

- Group 1: EWR/TEB
- Group 2: DFW/DAL, LAX/LGB
- Group 3: JFK/LGA, IAD/DCA, SFO/OAK
- Group 4: ORD/MDW
- Group 5: IAH/HOU
- Group 6: MIA/FLL

In addition, a Generic Metroplex consisting of runways in series (as opposed to the parallel runways in the Miami Metroplex) is represented by Group 7: “Half SFO/SJC” Metroplex.

# Metroplex Runway Layout Group 1: EWR/TEB

## Runway Layout Group 1: EWR/TEB

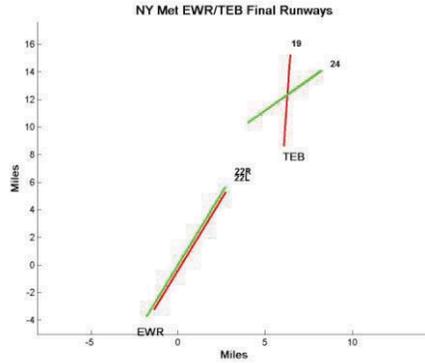


Figure 16: Metroplex Runway Layout Group 1: EWR/TEB

# Metroplex Runway Layout Group 2: DFW/DAL

## Runway Layout Group 2: Dallas

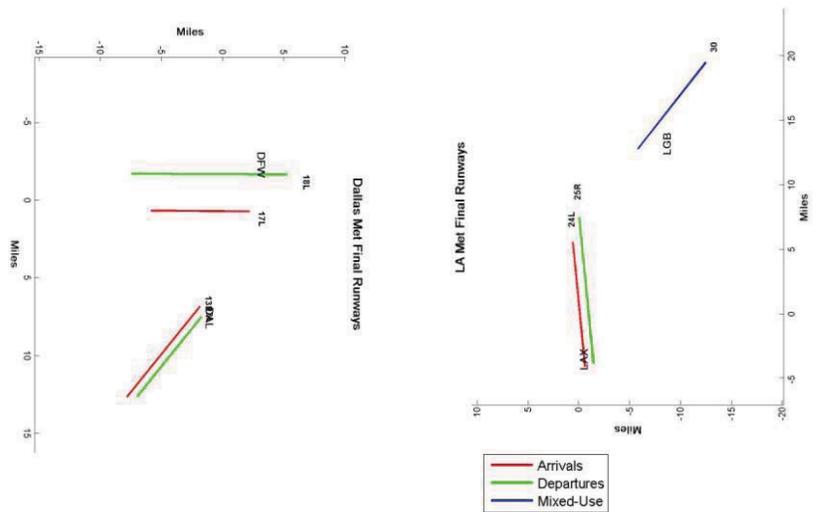


Figure 17: Metroplex Runway Layout Group 2: DFW/DAL

# Metroplex Runway Layout Group 3: JFK/LGA

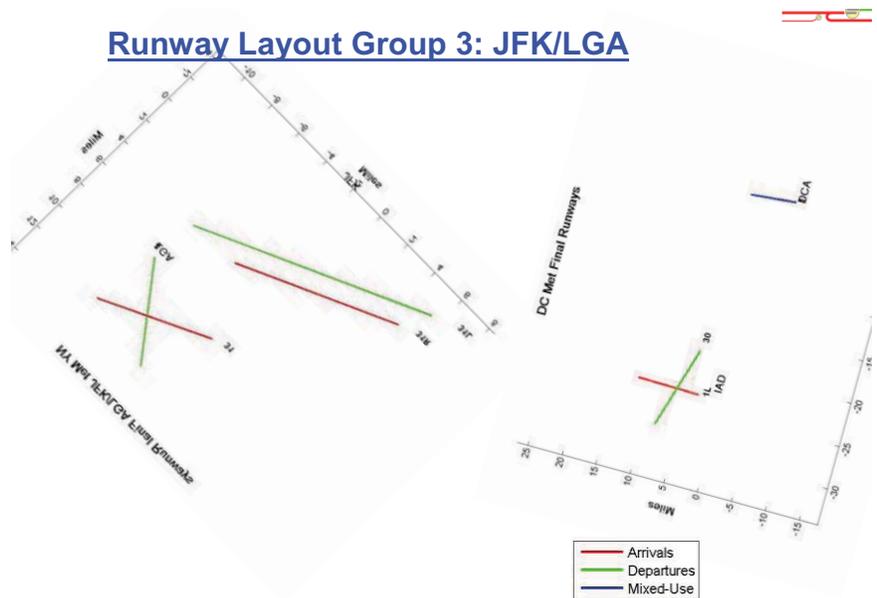


Figure 18: Metroplex Runway Layout Group 3: JFK/LGA, part 1

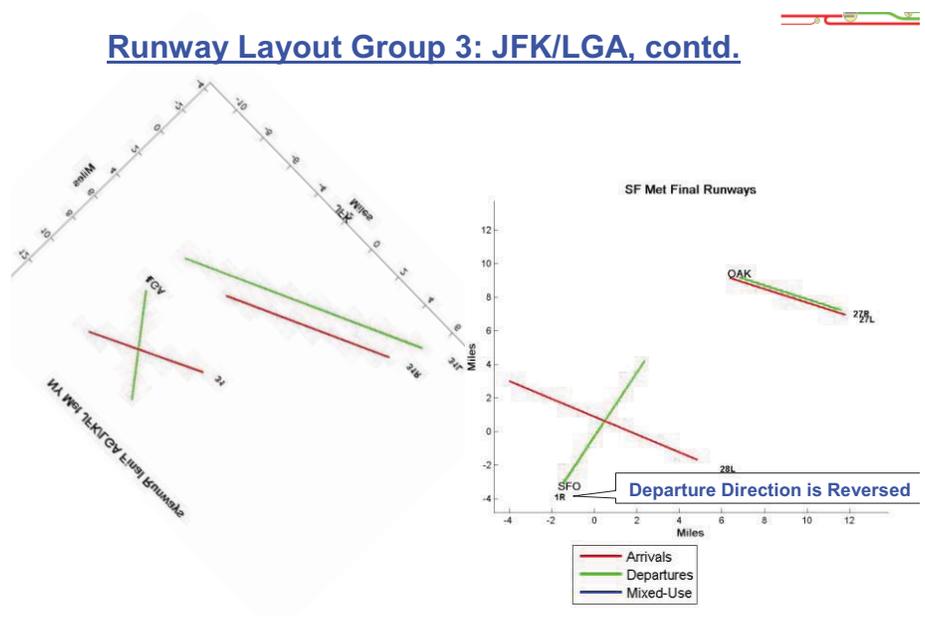


Figure 19: Metroplex Runway Layout Group 3: JFK/LGA, part 2

## Metroplex Runway Layout Group 4: ORD/MDW

### Runway Layout Group 4: Chicago

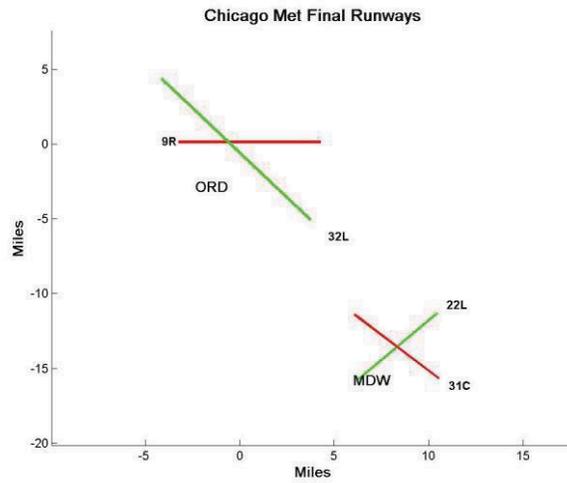


Figure 20: Metroplex Runway Layout Group 4: ORD/MDW

## Metroplex Runway Layout Group 5: IAH/HOU

### Runway Layout Group 5: Houston

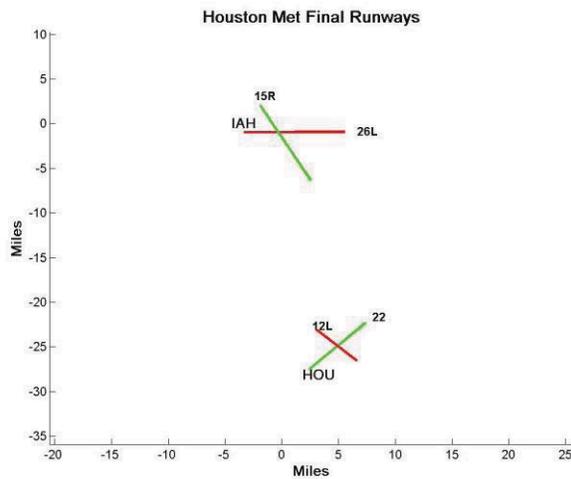


Figure 21: Metroplex Runway Layout Group 5: HOU/IAH

## Metroplex Runway Layout Group 6: MIA/FLL

### Runway Layout Group 6: Miami

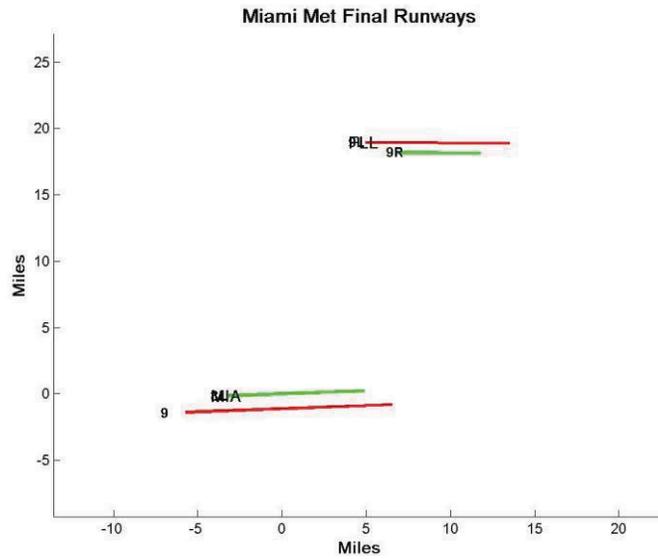


Figure 22: Metroplex Runway Layout Group 6: MIA/FLL

## Metroplex Runway Layout Group 7: Half SFO/SJC

### Runway Layout Group 7: "Half SFO/SJC"

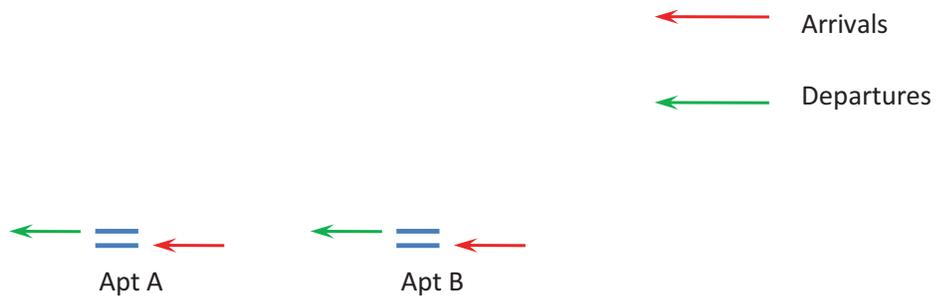


Figure 23: Metroplex Runway Layout Group 7: "Half SFO/SJC"

## **Arrival/Departure Paths for Generic Metroplexes**

This section describes the method of construction and the resulting arrival/departure paths for the Generic Metroplexes.

### **Generic Metroplex Runway and Airspace Geometries**

The seven Metroplex runway layouts chosen in the previous section are now paired with each of two Airspace Geometries, representing different levels of spatial segregation:

- **Airspace Geometry 1:** There are four equally spaced arrival fixes and four departure fixes located at the 40 NM ring from TRACON center. Each fix is shared by both airports.
- **Airspace Geometry 3:** There are four pairs of arrival fixes and four pairs of departure fixes at the 40 NM ring. Each fix in the pair is used by only one airport.

### **Construction Method for Lateral Arrival and Departure Paths**

Arrival/departure paths for each of these 14 runway/fix geometry combinations are constructed using the following rules, which are simplified from those used in our original study.

- Start with given fix and runway start/end threshold locations
- Construct exactly one turn in each path using the following criteria
  - For arrivals: turn ends at Final Approach Fix (5 mi from runway start)
  - For departures: turn begins 2 miles from runway end
  - Turning radii for arrivals: 2.616 miles, for departures: 2.9 miles

### **Results: Lateral Arrival and Departure Paths for Generic Metroplexes**

The following figures show the resulting arrival paths (in red) and departure paths (in blue), where the x and y axes are in nautical miles.

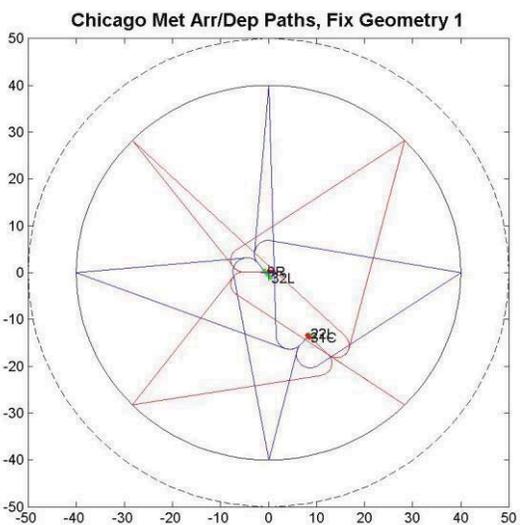
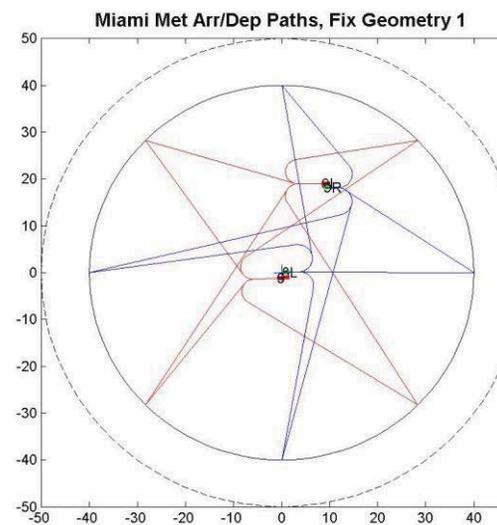
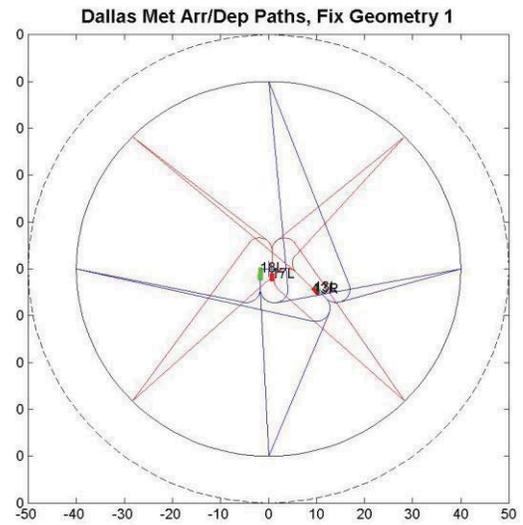
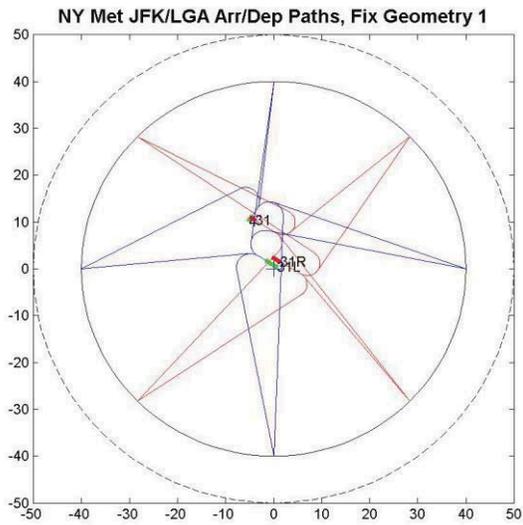


Figure 24: Lateral arrival/departure paths for the Generic New York 1, Dallas, Miami, and Chicago Metroplexes using Airspace Geometry 1.

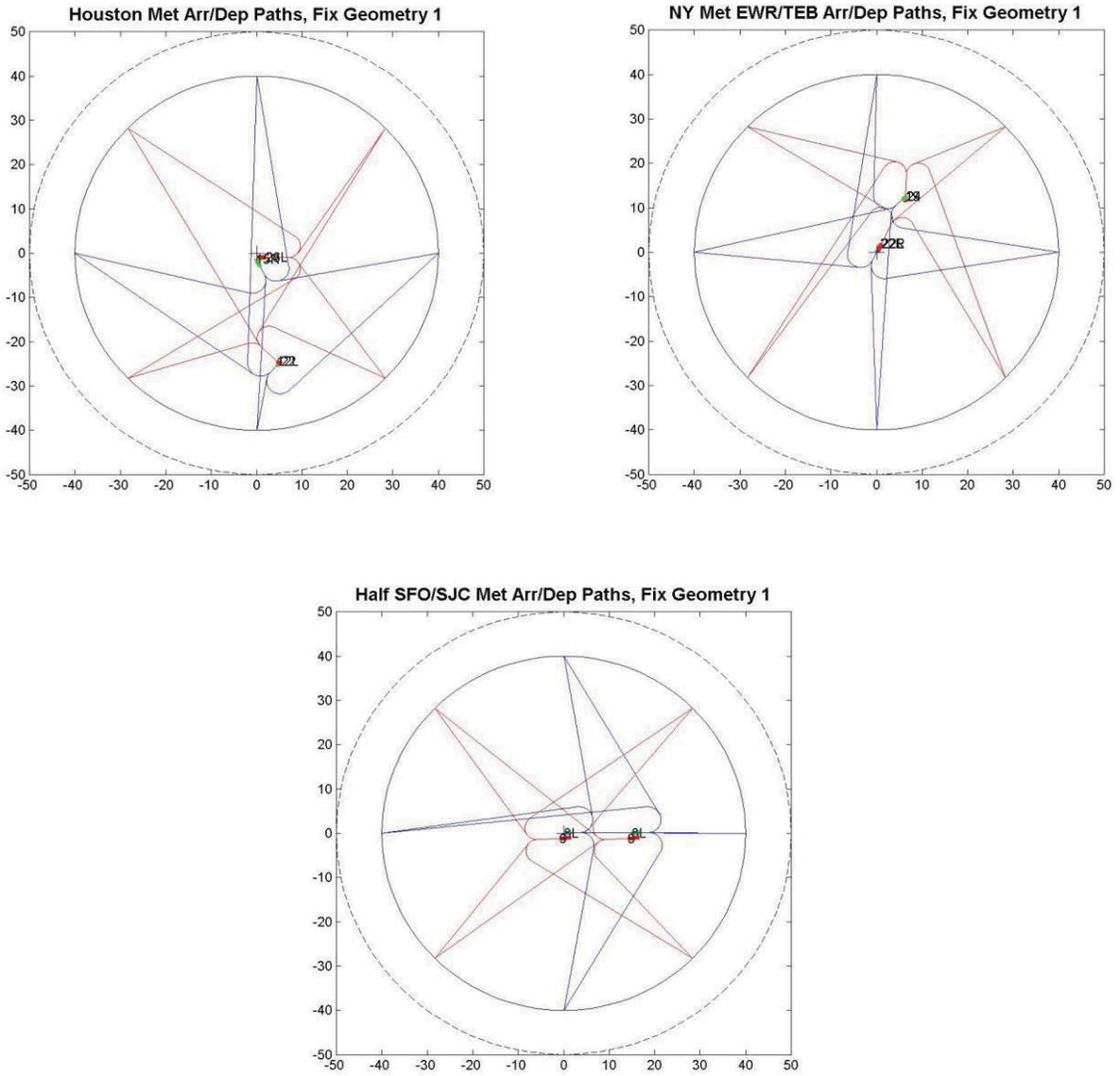
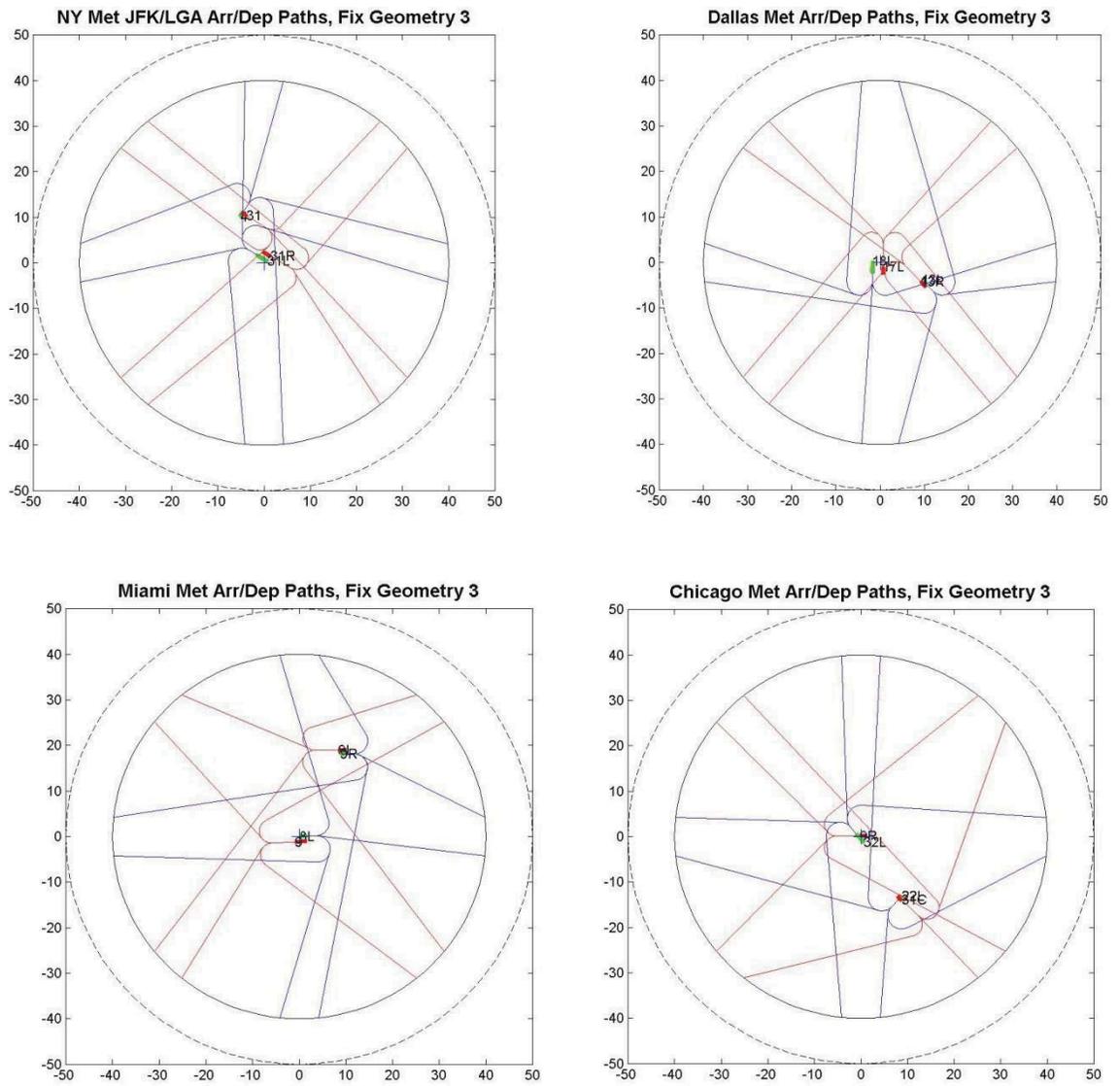
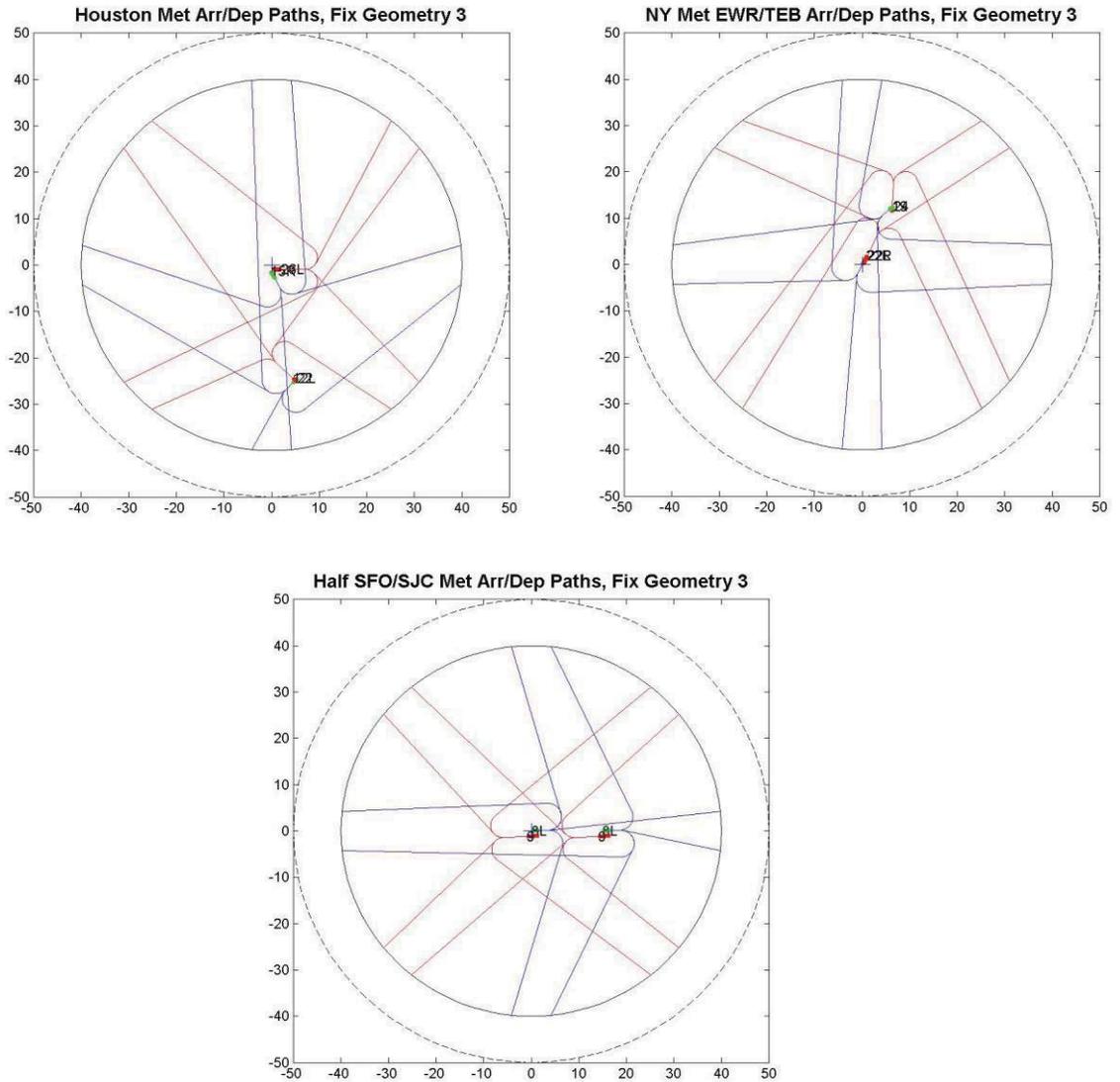


Figure 25: Lateral arrival/departure paths for the Generic Houston, New York 2, and Half SFO/SJC Metroplexes using Airspace Geometry 1.



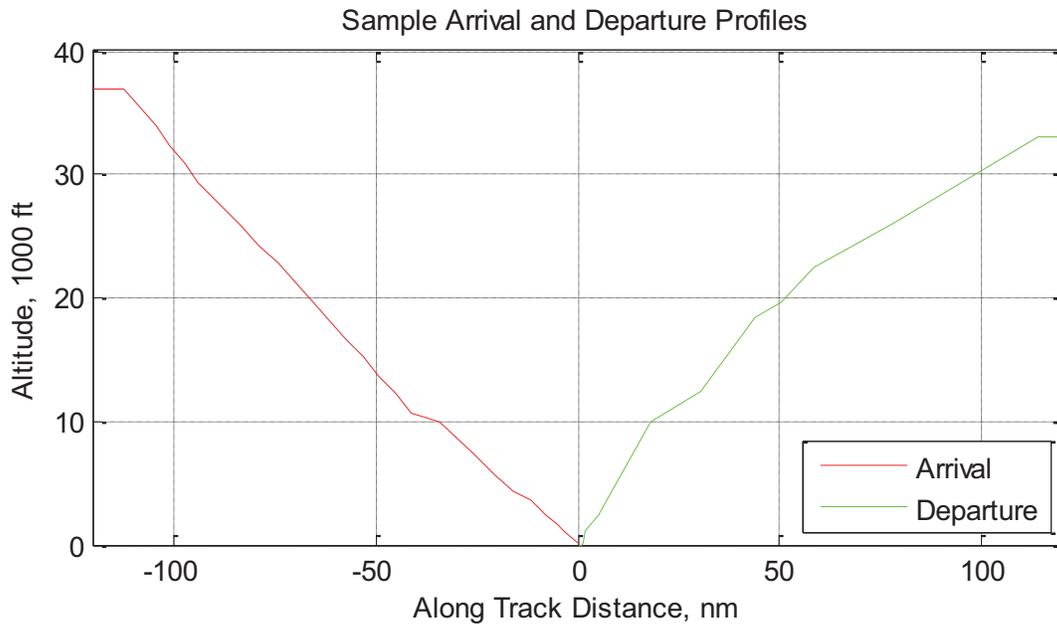
**Figure 26: Lateral arrival/departure paths for the Generic New York 1, Dallas, Miami, and Chicago Metroplexes using Airspace Geometry 3.**



**Figure 27: Lateral arrival/departure paths for the Generic Houston, New York 2, and Half SFO/SJC Metroplexes using Airspace Geometry 3.**

## **Altitude Profiles for Arrival and Departure Paths**

The altitude profiles applied to the arrival and departure paths follows the same method used in our previous report. The unrestricted arrival profile represents a CDA-type vertical profile. An example of the unrestricted arrival and departure profiles for optimal efficiency is shown



**Figure 28: Example of unrestricted arrival and departure profiles.**

The average arrival and departure path lengths for the Generic Metroplexes are 49.3 and 47.0 nmi, respectively. The average arrival and departure fix crossing altitudes are 13600 ft and 18400 feet, respectively.

### Traffic Demand for Generic Metroplexes

The traffic demand set was provided by Sensis Corporation; we only outline it here. The seed data set is an ETMS-derived record of IFR flights for September 26, 2006. This data set is “grown” to 3 times the total traffic volume in accordance with 2008 TAF forecasts. From the grown traffic demand set, KATL traffic is used to create traffic demand sets for both Generic Metroplex airports A and B. Flights are randomly pruned from the data set to achieve a demand/capacity ratio of 0.7 for airport A and 0.35 for airport B, assuming each airport’s capacity is 60 arrivals/hour and 60 departures/hour.

## Terminal Airspace Complexity for Generic Metroplexes

---

In this section the Flow Envelope Intersection metrics are computed for each of the Generic Metroplexes using various airspace geometries. The relative complexity for each airspace design is shown in overhead views of the resulting volume intersections. Tabular comparisons of volume intersections are also shown for the two fix geometries as well as the flows using arrival/departure routings before and after the application of spatial deconfliction.

### Metroplex Flow Envelope Intersection Metric Definition

The purpose of the Flow Envelope Intersection metric is to quantify the complexity of metroplex airspaces. Complexity of the airspace surrounding two or more closely-spaced airports will increase with the amount of overlap between their *aircraft flows*, defined as aggregations of flights following a perceptible pattern. Flights are grouped into flows by the proximity of their trajectories in space and time.

In order to quantify the interaction of flows, the notion of an *aircraft flow envelope* is developed and used to define two metrics for flow interactions: *flow envelope intersections*, and *flight-pairs*.

### Aircraft Flow Envelopes

For analysis of existing metroplexes, historical track data can be used to define aircraft flows. All of the tracks occurring during a specified window of time can be displayed in three dimensions using Metron Aviation's Airspace Design Tool (ADT). The grouping of tracks into flows can be determined visually, or in an automated way using clustering algorithms within ADT. For future metroplex design studies, the planned 3D paths from the arrival and departure fixes to the runways are employed to define the metroplex flows.

The *aircraft flow envelope* is a “minimal” volume of airspace encompassing most or all of the traffic in a flow. For the Generic Metroplexes, flow envelopes are created by first dividing each arrival and departure path into equal-length sections, defining “nodes” along the path, and assuming the paths are linear between these nodes. Vertical and horizontal dimensions are then added to each node in accordance with a specified RNP standard.

In order to find the intersections of these flow envelopes using the IntersectFlows algorithm, they must first be divided into convex polyhedra. The method for doing this is already suggested by the division of the centerline paths into linear segments divided by nodes, as described above. In the Generic Metroplex case the RNP vertical and horizontal designations at each node along the path naturally define the division of the flow into convex polyhedra. Figure 29 shows the overhead view of a set of flow envelopes, which have been divided into convex polyhedra using this method.

The Flow Envelope Intersections Metric is simply the sum of all pairwise intersection volumes of distinct flow envelopes in the metroplex. The formula for the intersection of one such pair of flows is given by the following:

- Let  $I(j,k)$  be the volume of the intersection of the  $j$ th and  $k$ th convex polyhedra from Flow1 and Flow2, respectively.
- The envelope intersection is defined as  $\sum_j \sum_k I(j,k)$

The sums are taken over the polyhedra of Flow1 and Flow2, respectively.

The total Flow Envelope Intersections Metric for a metroplex is the sum of all volume intersections of distinct pairs of flow envelopes in the metroplex, shown as green volumes in Figure 29.

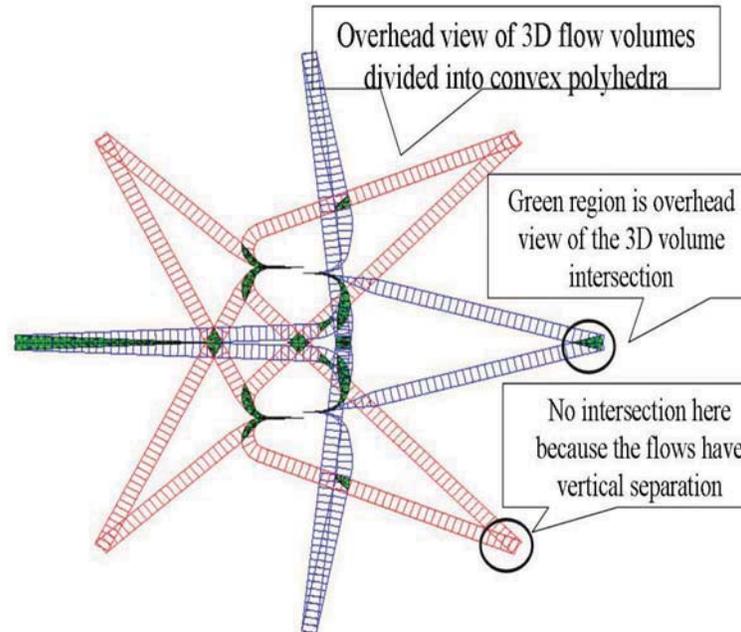


Figure 29: Plan view of 3D flow envelope intersection shown in green.

## Generic Metroplex Flow Envelopes

For the Generic Metroplex study aircraft flow envelopes are defined starting with the 3D paths for each combination of runway layout and fix location, and adding width and height dimensions to each path in accordance with defined horizontal and vertical parameters. In particular, take the parameters shown in

Table 1 as the maximum width and height, defining three flow shapes:

Table 1: Flow Shape Parameters

Flow Shape	1	2	3
Max Width, NM	3	2	2
Max Height, ft	1000	1000	200

Using these maximal values, the width and height at each point along the path is given as a function of distance from the runway (along the path) by linear interpolation between the values shown in Table .

Table 2: Flow Width and Height as Functions of Distance from the Runway

Distance from Runway	Flow Width	Flow Height
0 NM	100 ft	100 ft
5 NM	0.3 NM	Max height
> 10 NM	Max width	Max height

Each flow envelope is then divided into convex polyhedra having length 1 NM along the path centerline. Figure shows a plan view of the result for Geometry 3 using Flow Shape 2.

Intersections of the flow envelopes are colored in green. The number of polyhedra in each flow envelope will be determined by the length of the path from arrival or departure fix to runway.

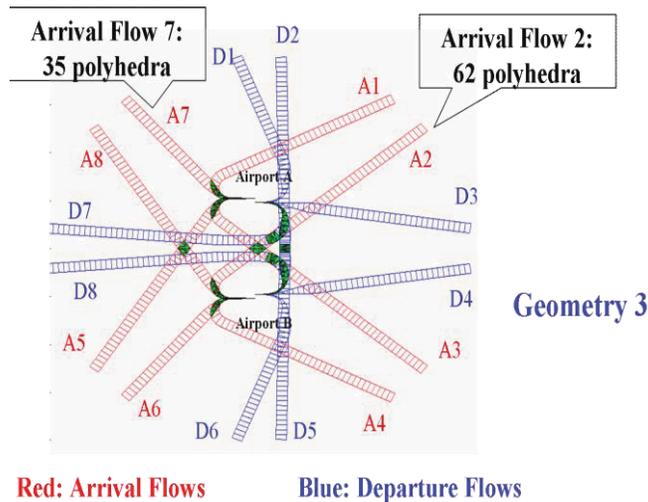


Figure 30: Overhead view of 3D aircraft flow envelopes showing polyhedra and labeling for arrival/departure flows and intersection volumes.

### Results: Generic Metroplex Flow Envelope Intersections

Airspace complexity as measured by Flow Envelope Intersection volume is compared for each of the seven Generic Metroplexes using each of the two Airspace Geometries.

#### Volume Intersections and Geometry Effects for Flow Shape 1

Table 3: IntersectFlows results for Generic Metroplexes, using Airspace Geometry 1 and Flow Shape 1

Generic Metroplex	Total Flow Envelope Volume, NM <sup>3</sup>	Flow Envelope Intersection Volume, NM <sup>3</sup>	Flow Intersection Percentage of Total Flow Volume
JFK/LGA	314.5	13.3	4.2
Dallas	314.4	19.3	6.1
Miami	323.2	11.0	3.4
Chicago	321.2	11.2	3.5
Houston	325.0	11.8	3.6
EWR/TEB	316.2	14.1	4.5
Half SFO/SJC	317.0	12.1	3.8

**Table 4: IntersectFlows Results for Generic Metroplexes, using Airspace Geometry 3 and Flow Shape 1**

<b>Generic Metroplex</b>	<b>Total Flow Envelope Volume, NM<sup>3</sup></b>	<b>Flow Envelope Intersection Volume, NM<sup>3</sup></b>	<b>Flow Intersection Percentage of Total Flow Volume</b>
<b>JFK/LGA</b>	311.7	11.1	3.6
<b>Dallas</b>	311.2	10.7	3.4
<b>Miami</b>	317.1	10.8	3.4
<b>Chicago</b>	314.6	9.7	3.1
<b>Houston</b>	320.7	10.0	3.1
<b>EWR/TEB</b>	312.8	13.2	4.2
<b>Half SFO/SJC</b>	313.3	11.8	3.8

**Table 5: Decrease in Intersection Volume using Airspace Geometry 3 vs. Airspace Geometry 1, using Flow Shape 1**

<b>Generic Metroplex</b>	<b>Intersection Volume Decrease Airspace 3 vs. 1 (percentage)</b>
<b>JFK/LGA</b>	16.5
<b>Dallas</b>	44.6
<b>Miami</b>	1.8
<b>Chicago</b>	13.4
<b>Houston</b>	15.3
<b>EWR/TEB</b>	6.4
<b>Half SFO/SJC</b>	2.5

## Volume Intersections and Geometry Effects for Flow Shape 2

Table 6: IntersectFlows results for Generic Metroplexes, using Airspace Geometry 1 and Flow Shape 2

Generic Metroplex	Total Flow Envelope Volume, NM <sup>3</sup>	Flow Envelope Intersection Volume, NM <sup>3</sup>	Flow Intersection Percentage of Total Flow Volume
<b>JFK/LGA</b>	210.6	7.9	3.8
<b>Dallas</b>	210.6	10.3	4.9
<b>Miami</b>	216.4	6.7	3.1
<b>Chicago</b>	215.2	6.8	3.2
<b>Houston</b>	217.7	6.8	3.1
<b>EWR/TEB</b>	211.8	8.4	4.0
<b>Half SFO/SJC</b>	212.3	7.3	3.4

Table 7: IntersectFlows Results for Generic Metroplexes, using Airspace Geometry 3 and Flow Shape 2

Generic Metroplex	Total Flow Envelope Volume, NM <sup>3</sup>	Flow Envelope Intersection Volume, NM <sup>3</sup>	Flow Intersection Percentage of Total Flow Volume
<b>JFK/LGA</b>	208.8	6.7	3.2
<b>Dallas</b>	208.5	6.6	3.2
<b>Miami</b>	212.4	6.8	3.2
<b>Chicago</b>	210.7	5.9	2.8
<b>Houston</b>	214.8	6.0	2.8
<b>EWR/TEB</b>	209.6	8.2	3.9
<b>Half SFO/SJC</b>	209.8	7.3	3.5

**Table 8: Decrease in Intersection Volume using Airspace Geometry 3 vs. Airspace Geometry 1, using Flow Shape 2**

<b>Generic Metroplex</b>	<b>Intersection Volume Decrease Airspace 3 vs. 1 (percentage)</b>
<b>JFK/LGA</b>	15.2
<b>Dallas</b>	35.9
<b>Miami</b>	-1.5
<b>Chicago</b>	13.2
<b>Houston</b>	11.8
<b>EWR/TEB</b>	2.4
<b>Half SFO/SJC</b>	0.0

**Volume Intersections and Geometry Effects for Flow Shape 3**

**Table 9: IntersectFlows results for Generic Metroplexes, using Airspace Geometry 1 and Flow Shape 3**

<b>Generic Metroplex</b>	<b>Total Flow Envelope Volume, NM<sup>3</sup></b>	<b>Flow Envelope Intersection Volume, NM<sup>3</sup></b>	<b>Flow Intersection Percentage of Total Flow Volume</b>
<b>JFK/LGA</b>	42.4	1.1	2.6
<b>Dallas</b>	42.4	1.3	3.1
<b>Miami</b>	43.6	1.1	2.5
<b>Chicago</b>	43.3	1.2	2.8
<b>Houston</b>	43.8	1.2	2.7
<b>EWR/TEB</b>	42.6	1.2	2.8
<b>Half SFO/SJC</b>	42.8	1.2	2.8

**Table 10: IntersectFlows Results for Generic Metroplexes, using Airspace Geometry 3 and Flow Shape 3**

<b>Generic Metroplex</b>	<b>Total Flow Envelope Volume, NM<sup>3</sup></b>	<b>Flow Envelope Intersection Volume, NM<sup>3</sup></b>	<b>Flow Intersection Percentage of Total Flow Volume</b>
<b>JFK/LGA</b>	42.0	1.2	2.9
<b>Dallas</b>	42.0	1.1	2.6
<b>Miami</b>	42.8	1.2	2.8
<b>Chicago</b>	42.4	1.1	2.6
<b>Houston</b>	43.2	1.1	2.5
<b>EWR/TEB</b>	42.2	1.2	2.8
<b>Half SFO/SJC</b>	42.3	1.2	2.8

**Table 11: Decrease in Intersection Volume using Airspace Geometry 3 vs. Airspace Geometry 1, using Flow Shape 3**

<b>Generic Metroplex</b>	<b>Intersection Volume Decrease Airspace 3 vs. 1 (percentage)</b>
<b>JFK/LGA</b>	-9.1
<b>Dallas</b>	15.4
<b>Miami</b>	-9.1
<b>Chicago</b>	8.3
<b>Houston</b>	8.3
<b>EWR/TEB</b>	0.0
<b>Half SFO/SJC</b>	0.0

**Table 12: Decrease in Intersection Volume using Airspace Geometry 3 vs. Airspace Geometry 1, all Flow Shapes**

<b>Generic Metroplex</b>	<b>Flow Shape 1 Intersection Volume Decrease for Airspace 3 vs. 1 (%)</b>	<b>Flow Shape 2 Intersection Volume Decrease for Airspace 3 vs. 1 (%)</b>	<b>Flow Shape 3 Intersection Volume Decrease for Airspace 3 vs. 1 (%)</b>
<b>JFK/LGA</b>	16.5	15.2	-9.1
<b>Dallas</b>	44.6	35.9	15.4
<b>Miami</b>	1.8	-1.5	-9.1
<b>Chicago</b>	13.4	13.2	8.3
<b>Houston</b>	15.3	11.8	8.3
<b>EWR/TEB</b>	6.4	2.4	0.0
<b>Half SFO/SJC</b>	2.5	0.0	0.0

# Overhead Views of Volume Intersections for all Generic Metroplexes using Flow Shape 1 and Fix Geometry 1

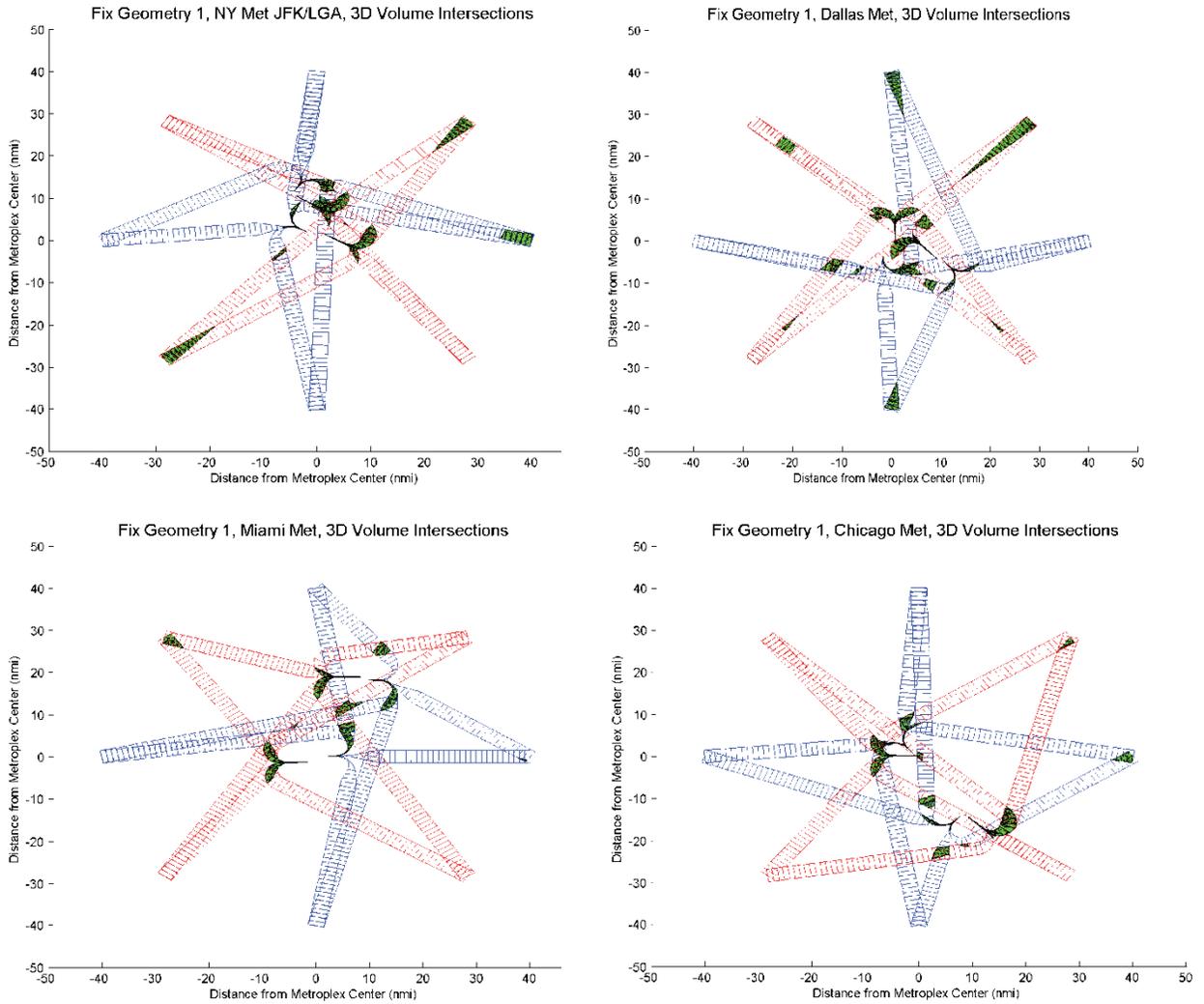
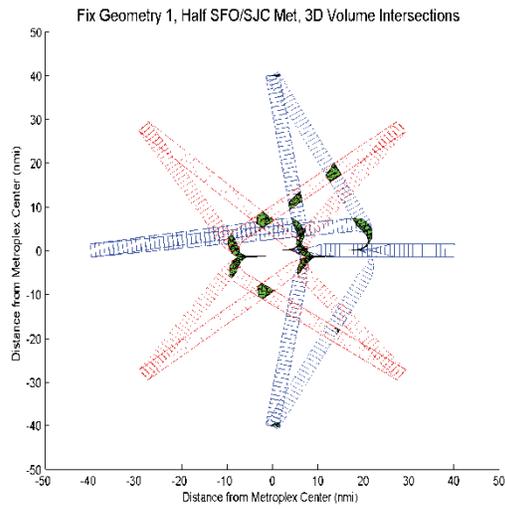
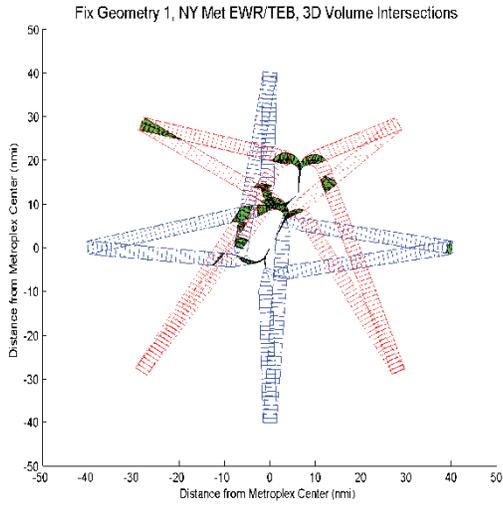
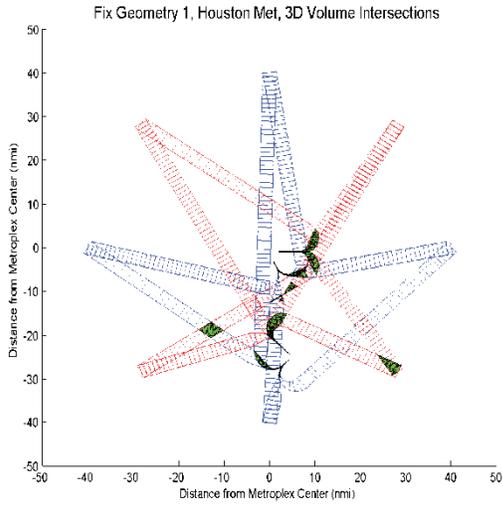


Figure 31: Overhead views of 3D Flow Envelope Intersections for Generic New York 1, Dallas, Miami and Chicago Metroplexes using Airspace Geometry 1.



**Figure 32: Overhead views of 3D Flow Envelope Intersections for Generic Houston, New York 2, and Half SFO/JFK Metroplexes using Airspace Geometry 1.**

# Side View of Volume Intersections for Dallas Metroplex using Flow Shape 1 and Fix Geometry 1

Fix Geometry 1, Dallas Met, 3D Volume Intersections

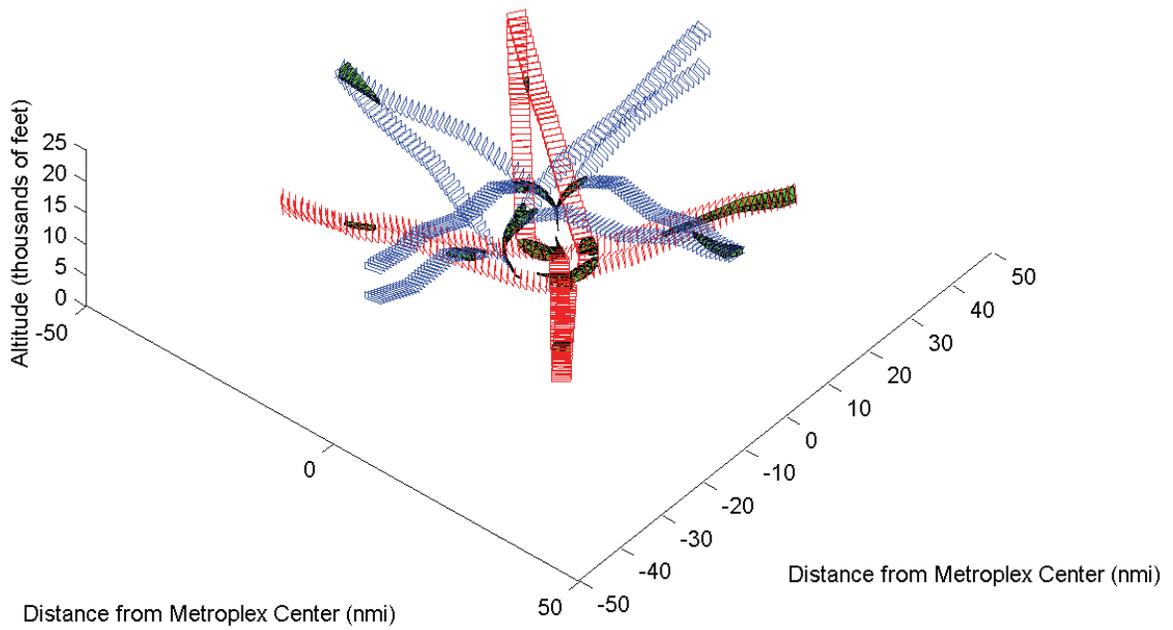


Figure 33: Side view of 3D Flow Envelope Intersections for Generic Dallas Metroplex using Airspace Geometry 1.

## Overhead Views of Volume Intersections for all Generic Metroplexes using Flow Shape 1 and Fix Geometry 3

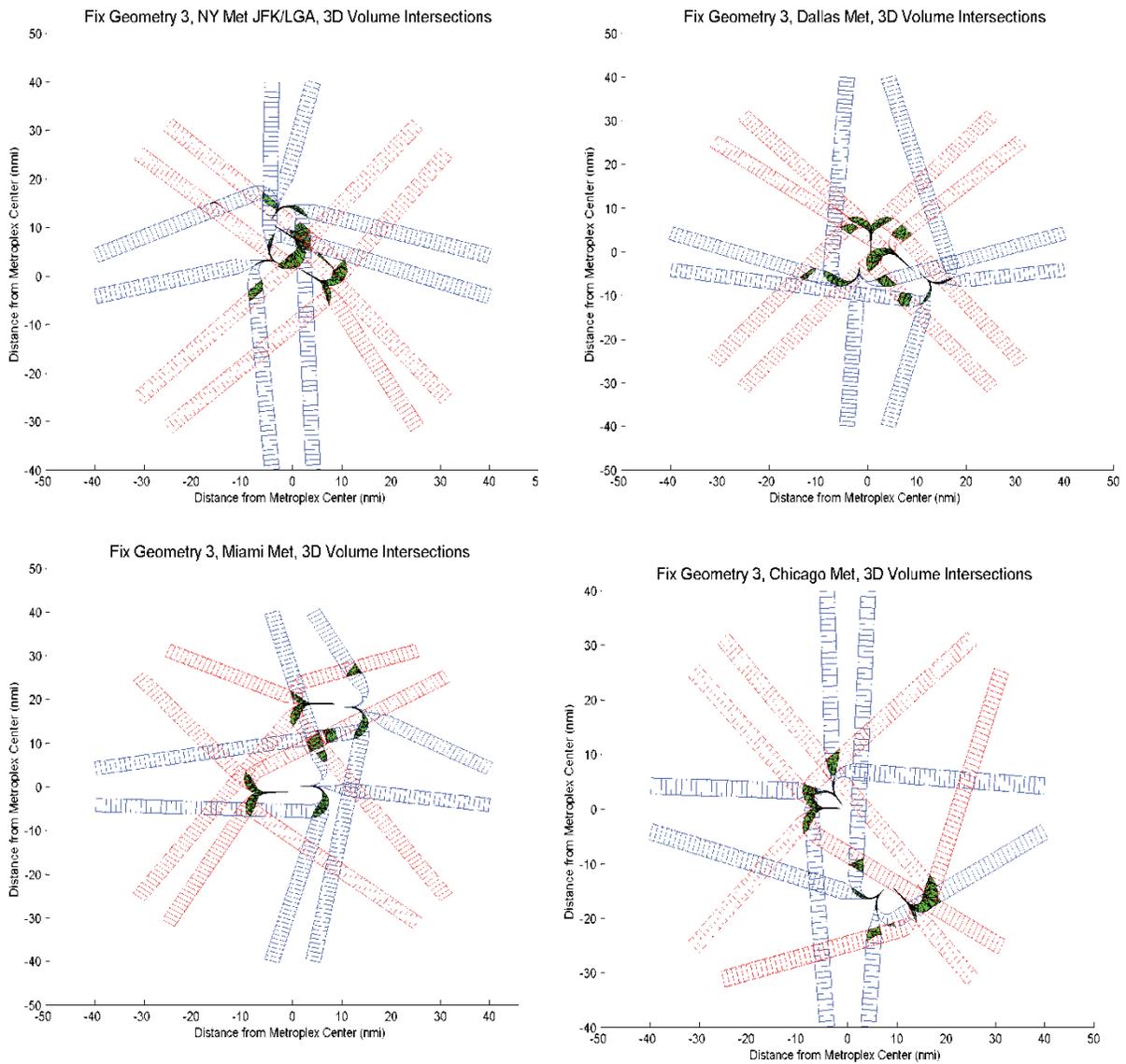
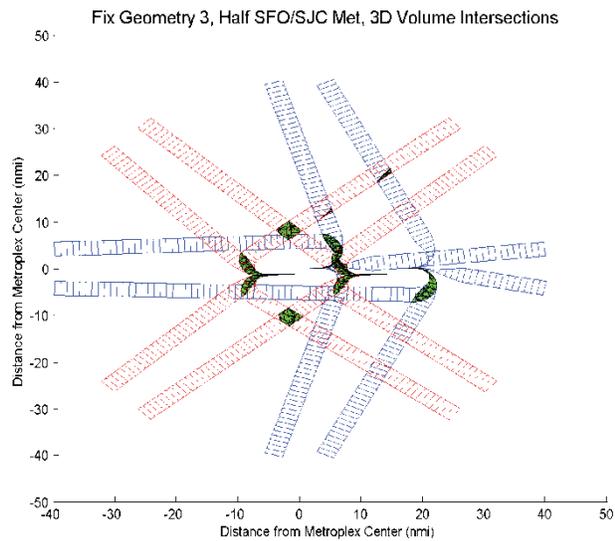
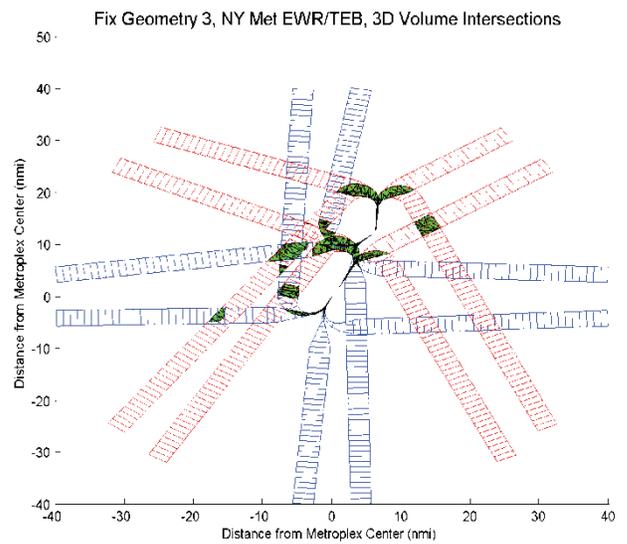
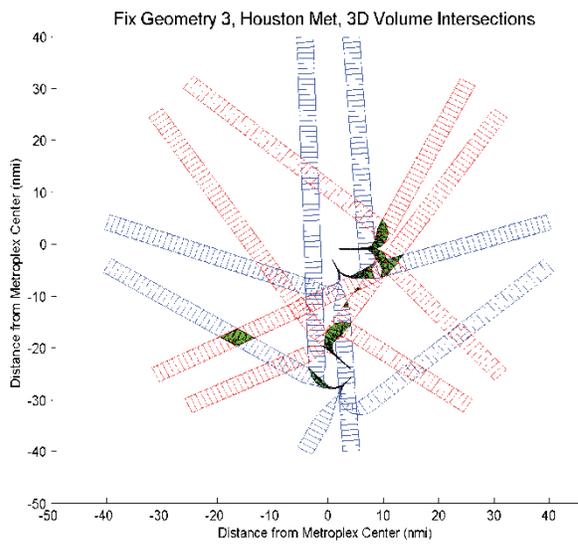


Figure 34: Overhead views of 3D Flow Envelope Intersections for Generic New York 1, Dallas, Miami and Chicago Metroplexes using Airspace Geometry 3.



**Figure 35: Overhead views of 3D Flow Envelope Intersections for Generic Houston, New York 2, and Half SFO/JFK Metroplexes using Airspace Geometry 3.**

## Volume Intersections for Deconflicted vs. Nondeconflicted Arrival/Departure Paths

**Table 13: IntersectFlows results for Deconflicted vs. Nondeconflicted (Baseline) Paths, using Airspace Geometry 1 and Flow Shape 1.**

<b>Generic Metroplex</b>	<b>Total Flow Envelope Volume, NM<sup>3</sup></b>	<b>Flow Envelope Intersection Volume, NM<sup>3</sup></b>	<b>Flow Intersection Percentage of Total Flow Volume</b>	<b>Percent Volume Intersection Decrease over Baseline</b>
<b>Miami Nondeconflicted</b>	311.7	9.1	2.9	0
<b>Miami Deconflicted</b>	311.7	4.8	1.5	47.3
<b>Chicago NonDeconflicted</b>	308.9	6.4	2.1	0
<b>Chicago Deconflicted</b>	308.9	6.1	2.0	4.7

**Table 14: IntersectFlows results for Deconflicted vs. Nondeconflicted Paths, using Airspace Geometry 3 and Flow Shape 1.**

<b>Generic Metroplex</b>	<b>Total Flow Envelope Volume, NM<sup>3</sup></b>	<b>Flow Envelope Intersection Volume, NM<sup>3</sup></b>	<b>Flow Intersection Percentage of Total Flow Volume</b>	<b>Percent Volume Intersection Decrease over Baseline</b>
<b>Miami Nondeconflicted</b>	304.7	5.8	1.9	0
<b>Miami Deconflicted</b>	304.6	3.2	1.1	44.8
<b>Chicago NonDeconflicted</b>	303.5	6.8	2.2	0
<b>Chicago Deconflicted</b>	303.5	4.8	1.6	29.4

## Overhead Views of Flow Volume Intersections for Deconflicted vs. Original Arrival/Departure Routes

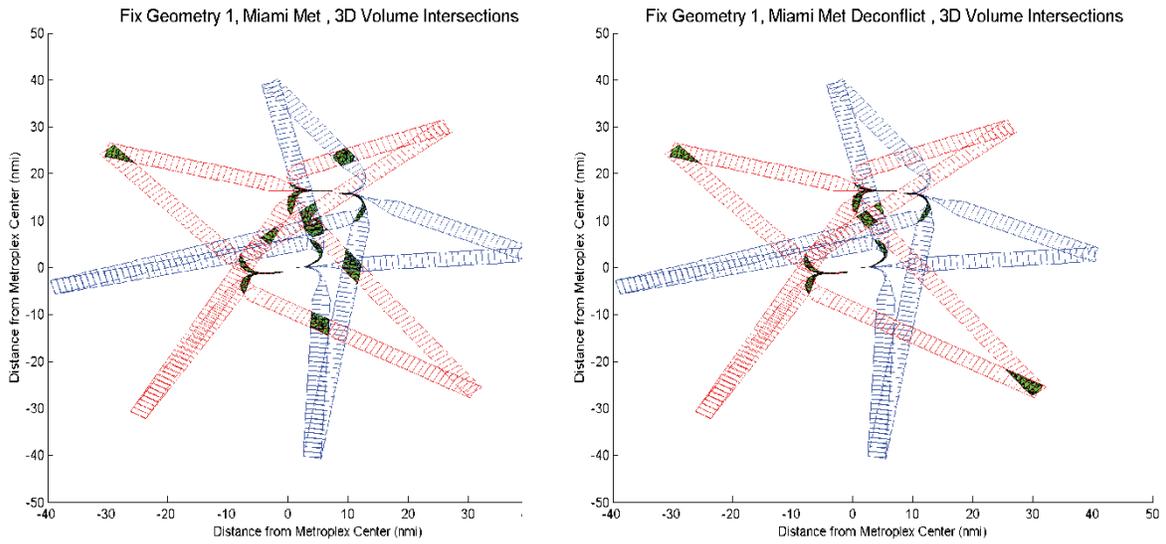


Figure 36: Overhead view of 3D flow volume intersections for Generic Miami with Fix Geometry 1, before and after spatial deconfliction of arrival/departure routes.

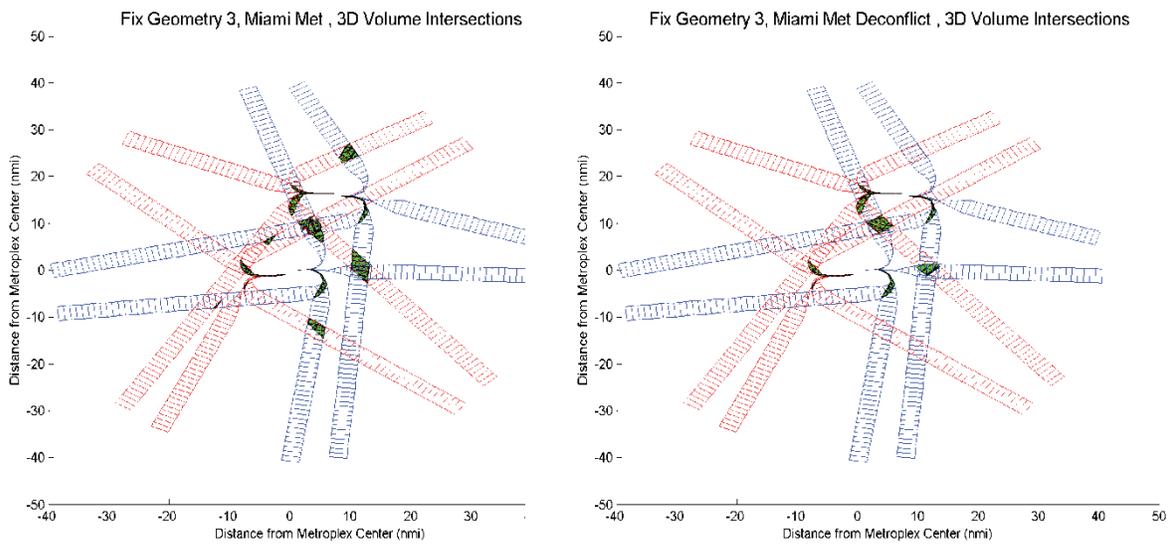
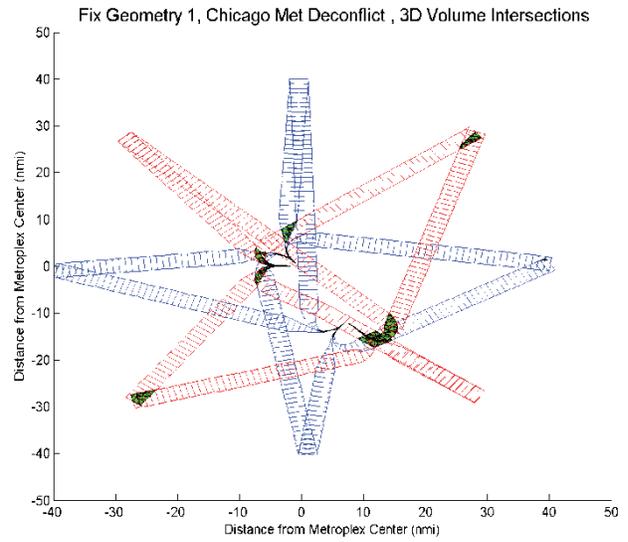
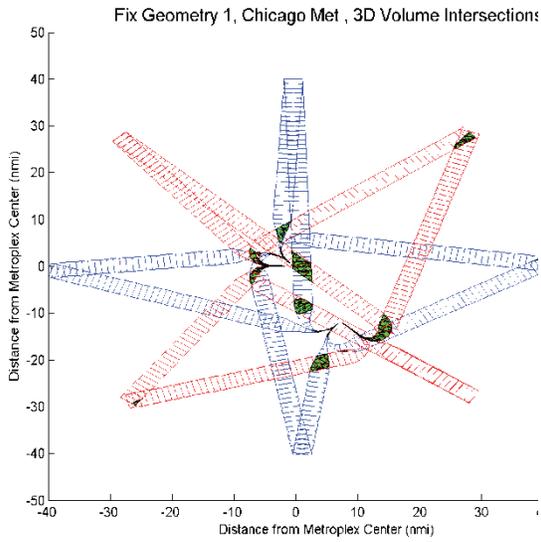
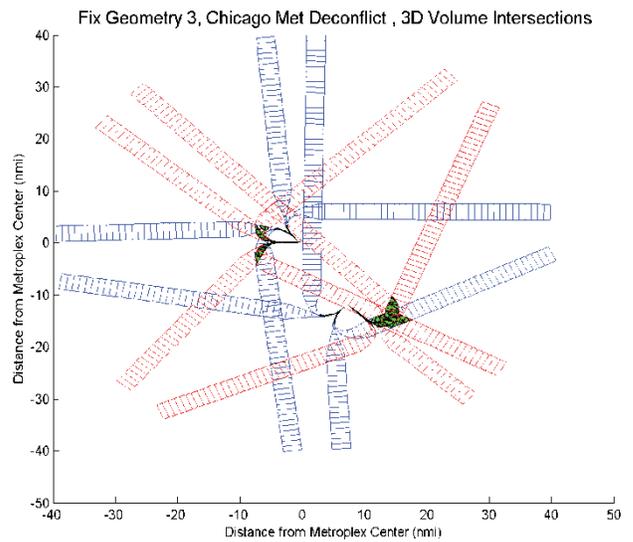
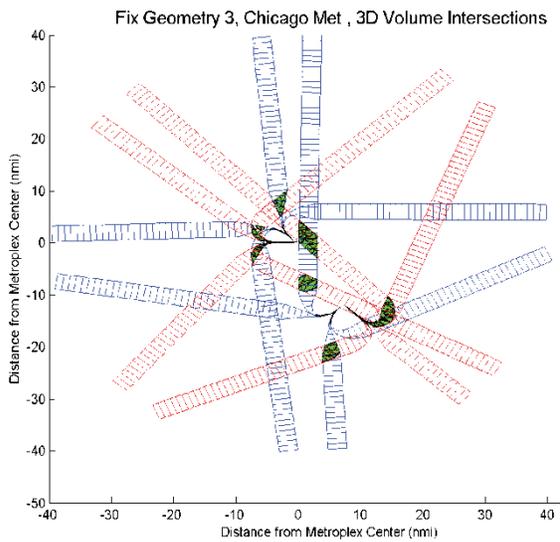


Figure 37: Overhead view of 3D flow volume intersections for Generic Miami with Fix Geometry 3, before and after spatial deconfliction of arrival/departure routes.



**Figure 38: Overhead view of 3D flow volume intersections for Generic Chicago with Fix Geometry 1, before and after spatial deconfliction of arrival/departure routes.**



**Figure 39: Overhead view of 3D flow volume intersections for Generic Chicago with Fix Geometry 3, before and after spatial deconfliction of arrival/departure routes.**

## Generic Metroplex Environmental Impact Analysis

---

In this section, the environmental impact for the seven Generic Metroplexes is studied, using a variety of alternative airspace designs. This analysis was conducted to evaluate the fuel burn and emissions impact of the two original airspace geometries, as well as the spatially deconflicted arrival/departure routes for each Airspace Geometry. The effects of scheduling and temporal control strategies were not considered for the sake of simplicity. The analysis utilized the NAS-wide Environmental Impact Model (NASEIM) to model fuel burn and emissions.

### NASEIM Fuel Burn Calculations

Fuel burn values in NASEIM are derived from Eurocontrol's Base of Aircraft Data (BADA) which contains fuel consumption rates for specific airframe and engine combinations at various altitudes and modes of flight (thrust settings). For portions of the flight below 3,000 ft above ground level (AGL), fuel burn is given by the Emissions and Dispersion Modeling System (EDMS). The basic formula for calculating fuel burn is given by:

$$F_{a,i} = ne_a \times ff_{a,i} \times tm_{a,i}$$

Where:

- $F_{a,i}$  = the fuel burned by aircraft  $a$ , while in mode  $i$
- $ne_a$  = the number of engines on aircraft  $a$
- $ff_{a,i}$  = the fuel flow rate of aircraft  $a$ , while in mode  $i$
- $tm_{a,i}$  = the time aircraft  $a$ , spends in mode  $i$

Flight modes are defined as climb, cruise, and descent, and are related to the engine thrust settings. The BADA and EDMS data tables specify fuel flow rates by altitude and flight mode. The tables specify fuel burn rates for low, nominal, and high aircraft weights; NASEIM assumes nominal aircraft weight. Fuel flow rates for intermediate altitudes are interpolated from the table values. Fuel burn is then calculated by multiplying the specified fuel flow rate by the time spent between each node in the flight trajectory. Summing the fuel burn for each trajectory segment gives the fuel burn for the entire flight.

### NASEIM Emissions Calculations

Emissions calculations in NASEIM utilize the value of fuel burned in each of several operational phases to estimate the mass of pollutants generated. For each of several pollutants (CO, HC, NO<sub>x</sub>, and SO<sub>x</sub>), the mass is given by:

$$M_{i,total} = \sum_m (F_m * EI_{i,m})$$

where  $F_m$  is the fuel burned in mode  $m$  (kg) and  $EI_{i,m}$  is the emission index for pollutant  $i$  in mode  $m$  (g/kg fuel).

Engine-specific International Civil Aviation Organization (ICAO)/EDMS taxi/idle fuel-flow values are used to derive the fuel burn during the taxi phase, and are combined with ICAO/EDMS taxi/idle emission factors to compute the pollutant totals emitted during surface movement.

The airborne aircraft trajectory is broken into several phases. Below 3000 feet AGL, engine-specific ICAO/EDMS fuel-flow rates and emissions indices are applied, with takeoff values used from takeoff to 1000 feet AGL, climb values between 1000 and 3000 feet on departure, and approach values between 3000 feet and touchdown. The mapping from aircraft type to engine type is made based on a review of the domestic commercial fleet and default engine assignments specified in the EDMS.

Above 3000 feet AGL, aircraft-specific BADA fuel-flow factors are used. Each distinct segment is classified as a climb, cruise, or descent segment, and the mean altitude of the segment is used to determine the corresponding BADA fuel-flow rate for that segment type.

## Generic Metroplex Inputs to NASEIM

The flight paths and traffic demand set for each Geometry used as inputs to NASEIM have been described in Section 0. Fuel burn and emissions calculations also require specification of aircraft types, and times at each point along each track. Speeds along each path were taken as linearly interpolated between 350 knots at the arrival or departure fix, and 150 knots at the runway threshold. The arrival or departure fix crossing times for each track were defined in the traffic demand set, and the times at each point along each track were computed from these points, using the interpolated speeds described above. A single aircraft type was used for this study (Boeing 752).

## Results for the Generic Metroplex Environmental Impact Study

Results are presented for each of the seven Generic Metroplexes using Runway Geometries 1 and 3. Fuel burn and emissions totals, as well as total distance traveled, are given for arrivals, departures, and all flights. The percentage change for each of these relative to the Baseline (Geometry 1) is also shown.

In addition, results are given for Miami Metroplex using spatially deconflicted arrival and departure paths as well as the original nondeconflicted paths, for Geometries 1 and 3. Here the percentage change is taken from the baseline of nondeconflicted paths.

**Table 15: Arrival Fuel Burn and Emissions for Generic Metroplex JFK/LGA, by Airspace Geometry**

	Arrival CO (kg)	Arrival HC (kg)	Arrival NOx (kg)	Arrival SOx (kg)	Arrival Fuel (kg)	Arrival Distance (km)
<b>Geometry 1</b>	1369.9	77.3	6863.7	702.5	702520.9	88459.7
<b>Geometry 3</b>	1361.6	76.8	6822.1	698.3	698272.5	87695.8

**Table 16: Departure Fuel Burn and Emissions for Generic Metroplex JFK/LGA, by Airspace Geometry**

	Departure CO (kg)	Departure HC (kg)	Departure NOx (kg)	Departure SOx (kg)	Departure Fuel (kg)	Departure Distance (km)
<b>Geometry 1</b>	2088.0	119.8	58444.5	2723.8	2723896.5	84416.8
<b>Geometry 3</b>	2062.5	118.3	58342.2	2710.6	2710621.9	83948.6

**Table 17: Total Fuel Burn and Emissions for Generic Metroplex JFK/LGA, by Airspace Geometry**

	Total CO (kg)	Total HC (kg)	Total NOx (kg)	Total SOx (kg)	Total Fuel (kg)	Total Distance (km)
<b>Geometry 1</b>	3458.0	197.1	65308.1	3426.4	3426360.4	172876.5
<b>Geometry 3</b>	3424.1	195.6	65164.3	3408.9	3408894.4	171644.4

**Table 18: Fuel Burn and Emissions for all Generic Metroplexes, Geometry 3 vs. Geometry 1 (Baseline), Arrivals**

Generic Metroplex	Arrival CO (%)	Arrival HC (%)	Arrival NOx (%)	Arrival SOx (%)	Arrival Fuel (%)	Arrival Distance (%)
JFK/LGA	-0.60	-0.60	-0.60	-0.60	-0.60	-0.86
Dallas	-1.58	-1.58	-1.58	-1.58	-1.58	-0.76
Miami	-0.39	-0.39	-0.39	-0.39	-0.39	-1.22
Chicago	-0.57	-0.57	-0.57	-0.57	-0.57	-1.46
Houston	-0.34	-0.34	-0.34	-0.34	-0.34	-0.63
EWR/TEB	-0.01	-0.01	-0.01	-0.01	-0.01	-0.02
Half SFO/SJC	-1.22	-1.22	-1.22	-1.22	-1.22	-1.24

**Table 19: Fuel Burn and Emissions for all Generic Metroplexes, Geometry 3 vs. Geometry 1 (Baseline), Departures**

Generic Metroplex	Departure CO (%)	Departure HC (%)	Departure NOx (%)	Departure SOx (%)	Departure Fuel (%)	Departure Distance (%)
JFK/LGA	-1.22	-1.20	-0.18	-0.49	-0.49	-0.55
Dallas	-0.64	-0.63	-0.06	-0.23	-0.23	-0.32
Miami	-1.44	-1.42	-0.57	-0.81	-0.81	-0.87
Chicago	-3.26	-3.22	-9.78	-1.62	-1.62	-1.79
Houston	-0.99	-0.98	-0.36	-0.53	-0.53	-0.70
EWR/TEB	-2.05	-2.01	-0.49	-0.93	-0.93	-1.10
Half SFO/SJC	-0.38	-0.37	-0.23	-0.25	-0.25	-0.29

**Table 20: Fuel Burn and Emissions for all Generic Metroplexes, Geometry 3 vs. Geometry 1 (Baseline), Totals**

Generic Metroplex	Total CO (%)	Total HC (%)	Total NOx (%)	Total SOx (%)	Total Fuel (%)	Total Distance (%)
JFK/LGA	-0.98	-0.96	-0.22	-0.51	-0.51	-0.71
Dallas	-1.00	-0.99	-0.22	-0.50	-0.50	-0.54
Miami	-1.03	-1.02	-0.55	-0.73	-0.73	-1.05
Chicago	-2.19	-2.18	-0.93	-1.40	-1.40	-1.62
Houston	-0.75	-0.74	-0.36	-0.49	-0.49	-0.66
EWR/TEB	-1.26	-1.24	-0.44	-0.75	-0.75	-0.65
Half SFO/SJC	-0.71	-0.70	-0.34	-0.45	-0.45	-0.78

## **Environmental Impact Results for Spatially Deconflicted vs. Original Arrival/Departure Routings**

The following tables show the increase in fuel burn and emissions for the arrival/departure routings which have been spatially deconflicted, vs. the original routings. It is important to notice that all numbers shown are in hundredths of a percent (‰). This is understandable since in most cases only four of the sixteen routes are changed by the deconfliction, and within each of these routes the changes are minor.

**Fuel Burn and Emissions Increase for Generic Miami Deconflicted vs. Nondeconflicted (Baseline), Arrivals**

<b>Fix Geometry</b>	<b>Arrivals CO (‰)</b>	<b>Arrivals HC (‰)</b>	<b>Arrivals NOx (‰)</b>	<b>Arrivals SOx (‰)</b>	<b>Arrivals Fuel (‰)</b>	<b>Arrivals Distance (‰)</b>
<b>Geometry 1</b>	1.11	1.11	1.11	1.11	1.11	0.47
<b>Geometry 3</b>	0.00	0.00	0.00	0.00	0.00	-0.11

**Fuel Burn and Emissions for Generic Miami Deconflicted vs. Nondeconflicted (Baseline), Departures**

<b>Fix Geometry</b>	<b>Departures CO (‰)</b>	<b>Departures HC (‰)</b>	<b>Departures NOx (‰)</b>	<b>Departures SOx (‰)</b>	<b>Departures Fuel (‰)</b>	<b>Departures Distance (‰)</b>
<b>Geometry 1</b>	-0.82	-0.81	-0.16	-0.33	-0.33	0.12
<b>Geometry 3</b>	0.09	0.09	0.22	0.20	0.20	-0.10

**Fuel Burn and Emissions Percent Change for Generic Miami Deconflicted from Nondeconflicted (Baseline), Totals**

<b>Fix Geometry</b>	<b>Total CO (‰)</b>	<b>Total HC (‰)</b>	<b>Total NOx (‰)</b>	<b>Total SOx, (‰)</b>	<b>Total Fuel (‰)</b>	<b>Total Distance (‰)</b>
<b>Geometry 1</b>	-0.06	-0.06	-0.02	-0.04	-0.04	0.03
<b>Geometry 3</b>	0.06	0.06	0.20	0.16	0.16	-0.10

## Conclusions

---

The airspace complexity analysis shows that although the total volume of arrival/departure flows may be fairly uniform for a given RNP level, there may quite a wide range of flow intersection volumes depending on the location and orientation of runways in a metroplex. For example, Generic Dallas using Flow Shape 1 and Fix Geometry 1 has almost 50% more intersection volume than any of the other metroplexes, while its total volume is similar to the others. This may be explained by the proximity and orientation of the Generic Dallas runways. This situation for Dallas is alleviated by using Fix Geometry 3, which separates the arrival and departure flows from each airport at the fixes. This is the clearest case in which Fix Geometry 3 is best; as the RNP levels decrease the benefits are less dramatic, and in the case of Flow Shape 3, two metroplexes actually do better with Fix Geometry 1. All of these results are using the original (nondeconflicted) arrival and departure paths.

For deconflicted arrival and departure paths, one might expect no flow volume intersections at all, since the deconfliction was done in order to spatially separate the flows. However, the criteria used to generate the deconflicted routes were different than those used for this analysis, so it is interesting to see the effect of the deconfliction on a more coarse RNP level (especially using Flow Shape 1). In all of the four cases studied (Generic Miami and Generic Chicago, with Fix Geometries 1 and 3), deconfliction achieved a significant reduction in flow intersection volume – near 50% in two cases.

The environmental metrics study shows a benefit across the board using Fix Geometry 3 for all Generic Metroplexes. This likely resulting simply because all of the arrival and departure paths are shorter using this fix geometry. Finally, the difference in environmental impact between the deconflicted and original arrival/departure routes is shown to be insignificant. This is understandable since the changes made to the routes for deconfliction were intentionally minor.

# Appendix C – Generic Metroplex Geometry Analysis



## Overview

As mentioned in the main body of this report, our research team studied two different metroplex boundary-fix configurations (see Figure 29) and three different runway-interaction geometries (see Figure 30). These in combination presented a total of six different generic metroplex geometries for our analysis. As seen from Figure 30, we tried to maintain similarity with three busy and complicated NAS metroplexes in order to capture the variations in scheduling benefits as applied to multiple different metroplex sites across the NAS. The main body of the report discusses the scheduling results for the MIA-FLL like runway interaction geometry covering both the shared fixes and segregated fixes boundary-fix geometries. This appendix presents the scheduling results for all other geometries (MIA-FLL results are also included here for the sake of completeness).

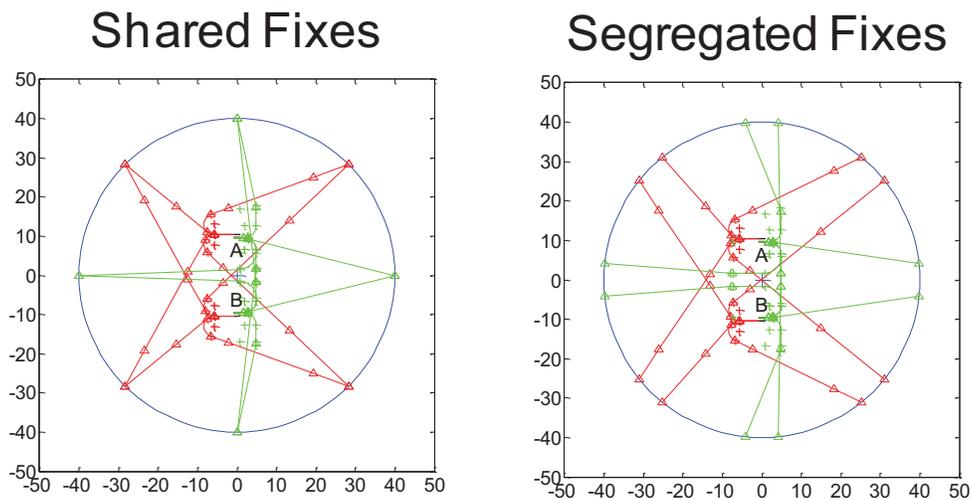


Figure 29: TRACON Boundary-fix Configurations Studied by the Georgia Tech Metroplex Research Team

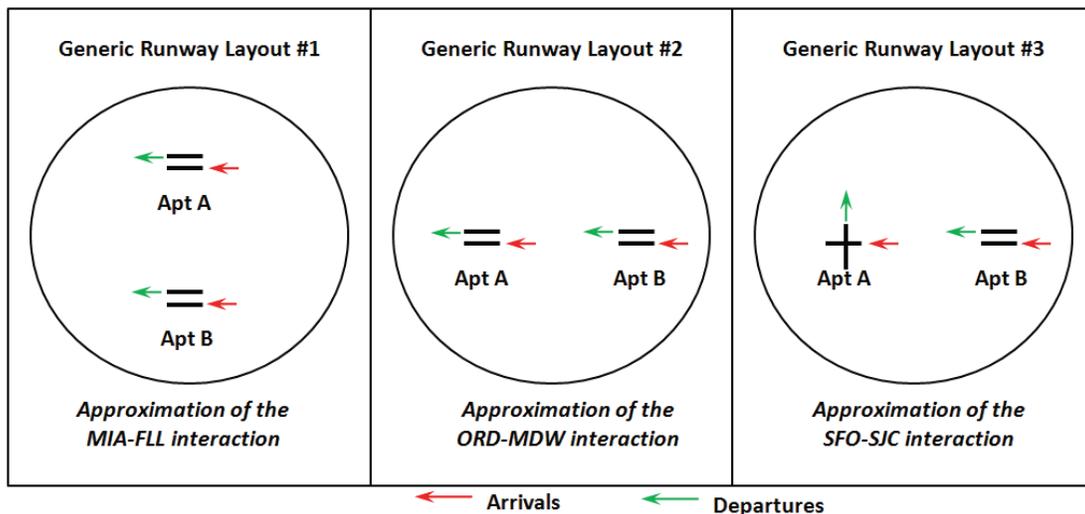


Figure 30: Metroplex Runway-Interaction Geometries Studied by the Georgia Tech Metroplex Research Team

## Analysis Methodology

To evaluate the two scheduling algorithms, a simple linked-list queuing simulation was developed that uses the exact same input files and configuration parameters so that they can be directly compared. This simulation was developed to remove any components or interactions in the simulation framework that were not directly considered in the scheduling algorithms. The simulation consisted of two queues, one at the arrival fix and one at the runway threshold. Each aircraft was injected into the simulation at the arrival fix at the scheduled time of arrival (STA). If the 5 NM separation was violated at any point, the trailing aircraft would be delayed until the required separation was achieved. Similarly, if the aircraft violated the separation defined in the wake vortex separation matrix, the trailing aircraft would be delayed. This delay was tabulated and the results of this delay are presented in this appendix.

## Results

The results are tabulated as follows: Each metroplex configuration was tabulated individually with three columns of results for all three traffic levels. The three simulation output values are the Enroute Delay, TRACON Delay, and Total Delay. The Enroute Delay is measured as the difference between the ETA to the arrival fix (which was the input value for the scheduling algorithms) and the time at which the aircraft could leave the arrival fix queue without violating any separation constraints. Implicitly, this value includes the difference between ETA and STA as well as any added delay that would be required to achieve the required 5 NM separation at the arrival fix. Similarly, the TRACON delay was computed by delaying each aircraft that is in queue for the runway long enough to achieve the required wake vortex separation requirements. Table 23 and Table 24 present the average results for the MIA-FLL metroplex configuration with both shared arrival fix airspace configurations and decoupled airspace configurations. This indicates that the MILP Multiplexer algorithm greatly outperforms the standard TMA algorithm when only the interactions considered within these algorithms are simulated. Comparing these two tables also would indicate that decoupling the airspace leads to some savings for both algorithms, but the difference between algorithms is much greater than the difference between airspace configurations. Table 29 and Table 30 present the cumulative results for the same simulations. Similarly, Table 25 and Table 26 present the average results for the SFO-SJC generic metroplex case while Table 31 and Table 32 tabulate the cumulative results. The average results for ORD-MDW are found in Table 27 and Table 28, while the cumulative results are contained in Table 33 and Table 34. These results also support our initial conclusions that while decoupling the airspace interactions will decrease delay, implementing optimal scheduling algorithms that consider all airspace interactions will provide a much greater impact in overall delay as traffic density increases. While the MILP showed little to no improvement over TMA for the low traffic density cases (even showing slightly worse performance in two cases), the MILP Multiplexer formulation shows greater than 10x improvement for all medium density cases and initial results for the high density cases show more than 20x reduction in total delay when compared to the high density TMA results.

**Table 23: MIA-FLL Average Delay for Shared Airspace [min]**

Density	Algorithm	Enroute Delay	TRACON Delay	Total Delay
Low	TMA	0.24	0.37	0.58
Low	MILP	0.17	0.01	0.18
Medium	TMA	11.99	0.58	12.57
Medium	MILP	0.77	0.01	0.78

<b>High</b>	TMA	31.47	0.63	32.10
<b>High</b>	MILP	1.34	0.02	1.36

**Table 24: MIA-FLL Average Delay for Decoupled Airspace [min]**

Density	Algorithm	Enroute Delay	TRACON Delay	Total Delay
<b>Low</b>	TMA	0.06	0.40	0.46
<b>Low</b>	MILP	0.15	0.00	0.15
<b>Medium</b>	TMA	6.82	0.34	7.16
<b>Medium</b>	MILP	0.59	0.010	0.60
<b>High</b>	TMA	18.61	0.49	19.09
<b>High</b>	MILP	1.10	0.02	1.12

**Table 25: SFO-SJC Average Delay for Shared Airspace [min]**

Density	Algorithm	Enroute Delay	TRACON Delay	Total Delay
<b>Low</b>	TMA	0.25	0.08	0.33
<b>Low</b>	MILP	0.17	0.01	0.18
<b>Medium</b>	TMA	11.92	0.11	12.02
<b>Medium</b>	MILP	0.81	0.01	0.82
<b>High</b>	TMA	30.03	0.12	30.15
<b>High</b>	MILP	1.41	0.02	1.43

**Table 26: SFO-SJC Average Delay for Decoupled Airspace [min]**

Density	Algorithm	Enroute Delay	TRACON Delay	Total Delay
<b>Low</b>	TMA	0.05	0.10	0.15
<b>Low</b>	MILP	0.15	0.00	0.15
<b>Medium</b>	TMA	7.06	0.08	7.14
<b>Medium</b>	MILP	0.56	0.01	0.57
<b>High</b>	TMA	18.49	0.08	18.57
<b>High</b>	MILP	1.07	0.02	1.09

**Table 27: ORD-MDW Average Delay for Shared Airspace [min]**

Density	Algorithm	Enroute Delay	TRACON Delay	Total Delay
<b>Low</b>	TMA	0.26	0.10	0.36
<b>Low</b>	MILP	0.20	0.00	0.21
<b>Medium</b>	TMA	12.00	0.10	12.11
<b>Medium</b>	MILP	0.84	0.01	0.86
<b>High</b>	TMA	29.93	0.11	30.04
<b>High</b>	MILP	x.xx	x.xx	x.xx

**Table 28: ORD-MDW Average Delay for Decoupled Airspace [min]**

Density	Algorithm	Enroute Delay	TRACON Delay	Total Delay
<b>Low</b>	TMA	0.06	0.09	0.15
<b>Low</b>	MILP	0.18	0.00	0.19
<b>Medium</b>	TMA	7.04	0.08	7.13
<b>Medium</b>	MILP	0.55	0.01	0.56
<b>High</b>	TMA	18.64	0.08	18.72
<b>High</b>	MILP	1.08	0.01	1.09

**Table 29: MIA-FLL Cumulative Delay for Shared Airspace [min]**

Density	Algorithm	Enroute Delay	TRACON Delay	Total Delay
<b>Low</b>	TMA	138.00	237.42	375.42
<b>Low</b>	MILP	110.08	3.84	113.92
<b>Medium</b>	TMA	11950.74	576.23	12526.97
<b>Medium</b>	MILP	765.39	10.41	775.80

<b>High</b>	TMA	40248.79	805.18	41053.97
<b>High</b>	MILP	1716.61	24.90	1741.52

**Table 30: MIA-FLL Cumulative Delay for Decoupled Airspace [min]**

Density	Algorithm	Enroute Delay	TRACON Delay	Total Delay
<b>Low</b>	TMA	36.23	260.12	296.35
<b>Low</b>	MILP	97.72	1.00	98.72
<b>Medium</b>	TMA	6803.02	342.39	7145.41
<b>Medium</b>	MILP	584.10	9.93	594.03
<b>High</b>	TMA	23814.77	624.92	24439.69
<b>High</b>	MILP	1410.70	20.53	1431.23

**Table 31: SFO-SJC Cumulative Delay for Shared Airspace [min]**

Density	Algorithm	Enroute Delay	TRACON Delay	Total Delay
<b>Low</b>	TMA	158.43	53.84	212.27
<b>Low</b>	MILP	110.87	2.70	113.57
<b>Medium</b>	TMA	11884.01	104.35	11988.36
<b>Medium</b>	MILP	805.55	7.11	812.66
<b>High</b>	TMA	38406.62	150.43	38557.05
<b>High</b>	MILP	1809.09	25.72	1834.81

**Table 32: SFO-SJC Cumulative Delay for Decoupled Airspace [min]**

Density	Algorithm	Enroute Delay	TRACON Delay	Total Delay
<b>Low</b>	TMA	33.87	61.68	95.55
<b>Low</b>	MILP	94.81	1.61	96.42
<b>Medium</b>	TMA	7046.84	77.35	7124.19
<b>Medium</b>	MILP	557.10	11.94	569.04
<b>High</b>	TMA	23669.25	105.72	23774.98
<b>High</b>	MILP	1372.18	25.51	1397.69

**Table 33: ORD-MDW Cumulative Delay for Shared Airspace [min]**

Density	Algorithm	Enroute Delay	TRACON Delay	Total Delay
<b>Low</b>	TMA	169.08	63.97	233.06
<b>Low</b>	MILP	130.75	2.57	133.32
<b>Medium</b>	TMA	11965.63	103.52	12069.15
<b>Medium</b>	MILP	841.19	11.77	852.96
<b>High</b>	TMA	38283.05	141.36	38424.41
<b>High</b>	MILP	x.xx	x.xx	x.xx

**Table 34: ORD-MDW Cumulative Delay for Decoupled Airspace [min]**

Density	Algorithm	Enroute Delay	TRACON Delay	Total Delay
<b>Low</b>	TMA	36.02	60.99	97.01
<b>Low</b>	MILP	118.24	1.32	119.55
<b>Medium</b>	TMA	7021.76	83.48	7105.23
<b>Medium</b>	MILP	548.73	8.06	556.79
<b>High</b>	TMA	23862.89	98.43	23961.32
<b>High</b>	MILP	1377.46	18.44	1395.91

**REPORT DOCUMENTATION PAGE**

*Form Approved  
OMB No. 0704-0188*

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.  
**PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

<b>1. REPORT DATE (DD-MM-YYYY)</b> 01-02-2012		<b>2. REPORT TYPE</b> Contractor Report		<b>3. DATES COVERED (From - To)</b> June 8, 2010 to June 11, 2011	
<b>4. TITLE AND SUBTITLE</b>  Investigation, Modeling, and Analysis of Integrated Metroplex Arrival and Departure Coordination Concepts				<b>5a. CONTRACT NUMBER</b> NNL10AA14C	
				<b>5b. GRANT NUMBER</b>	
				<b>5c. PROGRAM ELEMENT NUMBER</b>	
<b>6. AUTHOR(S)</b> Clarke, John-Paul B.; Brooks, James; McClain, Evan; Paladhi, Anwesha Roy; Li, Leihong; Schleicher, David; Saraf, Aditya; Timar, Sebastian; Crisp, Don; Bertino, Jason; Laroza, Ryan; Cross, Carolyn				<b>5d. PROJECT NUMBER</b>	
				<b>5e. TASK NUMBER</b>	
				<b>5f. WORK UNIT NUMBER</b> 305295.02.07.07.02	
<b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b> NASA Langley Research Center Hampton, Virginia 23681-2199			<b>8. PERFORMING ORGANIZATION REPORT NUMBER</b>  ATL-2011-1		
<b>9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)</b> National Aeronautics and Space Administration Washington, DC 20546-0001			<b>10. SPONSOR/MONITOR'S ACRONYM(S)</b>  NASA		
			<b>11. SPONSOR/MONITOR'S REPORT NUMBER(S)</b> NASA/CR-2012-217344		
<b>12. DISTRIBUTION/AVAILABILITY STATEMENT</b> Unclassified - Unlimited Subject Category 01 Availability: NASA CASI (443) 757-5802					
<b>13. SUPPLEMENTARY NOTES</b>  Langley Technical Monitor: Michael R. Sorokach					
<b>14. ABSTRACT</b> This work involves the development of a concept that enhances integrated metroplex arrival and departure coordination, determines the temporal (the use of time separation for aircraft sharing the same airspace resources) and spatial (the use of different routes or vertical profiles for aircraft streams at any given time) impact of metroplex traffic coordination within the National Airspace System (NAS), and quantifies the benefits of the most desirable metroplex traffic coordination concept. Researching and developing metroplex concepts is addressed in this work that broadly applies across the range of airspace and airport demand characteristics envisioned for NextGen metroplex operations. The objective of this work is to investigate, formulate, develop models, and analyze an operational concept that mitigates issues specific to the metroplex or that takes advantage of unique characteristics of metroplex airports to improve efficiencies. The concept is an innovative approach allowing the NAS to mitigate metroplex interdependencies between airports, optimize metroplex arrival and departure coordination among airports, maximize metroplex airport throughput, minimize delay due to airport runway configuration changes, increase resiliency to disruptions, and increase the tolerance of the system to degrade gracefully under adverse conditions such as weather, traffic management initiatives, and delays in general.					
<b>15. SUBJECT TERMS</b> Airspace Concept; Airspace Delay; Airspace Simulation; Arrival Departure Coordination; Metroplex Geometries; Metroplex Operations; Multiplexer; NextGen; Spatial Control; Temporal Control					
<b>16. SECURITY CLASSIFICATION OF:</b>			<b>17. LIMITATION OF ABSTRACT</b>	<b>18. NUMBER OF PAGES</b>	<b>19a. NAME OF RESPONSIBLE PERSON</b>
<b>a. REPORT</b>	<b>b. ABSTRACT</b>	<b>c. THIS PAGE</b>			STI Help Desk (email: help@sti.nasa.gov)
U	U	U	UU	190	<b>19b. TELEPHONE NUMBER (Include area code)</b> (443) 757-5802