Routing Flexible Traffic into Metroplex

Peng Wei[*] Jit-Tat Chen[*]
Dominick Andrisani[*]
Dengfeng Sun[*]

This research introduces the concepts of metroplex and flexible flight. The dependency metric between the airports in a metroplex is illustrated. Moreover, this paper focuses on metroplex routing algorithm. The authors establish a network model and develop the routing algorithm for flexible flights which fly from their original airports to destination metroplexes based on the model. The routing algorithm is performed under the sector congestion constraint. A metroplex consists of several airports around a metropolitan area instead of one single airport. As a flexible flight follows the metroplex routing instructions and approaches the decision boundary of the metroplex, a scheduler in the Multi-center Traffic Management Advisor (McTMA) will decide which runway of which airport the flight will land in. Together the routing and the scheduling make a complete flexible flight operation.

Nomenclature

d_l	Dependency metric of runway length
d_o	Dependency metric of runway orientation
d_p	Dependency metric of runway proximity
H	Hash table of waypoints for Jetway Network
c	Central point of all metroplex airports
ap_i	The ith airport in the metroplex
$w_{\alpha_{\max}}$	The largest great circle distance weight
w'_{α_e}	Normalized great circle distance weight on Edge e
γ	Penalty strengthen coefficient
w'_{β_e}	Sector congestion penalty weight

I. Introduction

With rapid growth of air traffic, the airports in a metropolitan area can not be considered as separated entities, but rather as subsystems of a larger, interdependent system. We call such a system of airports a *metroplex*. "Concept of Operations for the Next Generation Air Transportation System" [1] from The Joint Planning and Development Office (JPDO) defines a metroplex as a group of two or more adjacent airports whose arrival and departure operations are highly interdependent.

Airborne traffic congestion, airports in close proximity, and limited infrastructure resource lower the efficiency in those busy metroplex airspace. Separation rules and environmental problems also bring down efficiencies. Resource optimization of all the runways and airports and associated separation rules within the same metroplex will improve the overall throughput and also potentially reduce noise and emissions.

A metroplex phenomenon is an interaction between two or more airports in close geographically proximity and it has been observed and analysed by some researchers. Atkins and his Mosaic ATM colleagues [2] observed the metroplex phenomena in San Francisco bay area and showed that some metroplex phenomena will affect the total capacity of the metroplex airport system. In [2] the authors also built an initial framework

[*] School of Aeronautics and Astronautics, Purdue University.

to understand the nature of metroplexes by presenting a detailed metroplex definition based on a set of measurable dimensions.

The dependencies and impacts of the three major airports JFK, LGA and EWR in NYNJ metroplex were analysed by DeLaurentis and Ayyalasomayajula from Purdue University together with their collaboration partner George Mason University [3] sponsored by NASA. The dependency metric is developed to formulate policies and strategies for metroplex operations.

McClain and Clarke [4] developed the metric for metroplex clustering analysis and Clarke has been working on building a theoretical framework for quantifying the interactions between the airports in a metroplex. Donohue studied the airports in NYNJ metroplex early in 2008 [5, 6] and the two papers took the first look at the status of each airport in terms of the markets served, seat capacity, delays and other features. His recent work [7] with Schaar has applied metroplex concept to improve the domestic airline services.

For metroplex routing and scheduling, there are very few literature works existing. However, many weather avoidance reroute generating algorithms have already been developed and they can be borrowed to solve the routing problem with sector congestion constraint by treating congested sectors as severe weather areas. [8] adopted a grid network model to generate a reroute for a general aviation aircraft in free flight. Unfortunately, this algorithm can not be extended for commercial flights with Air Traffic Control.

In [9] the authors considered a mix of a grid network and a waypoint network when generating reroutes for traffic flows approaching the destination. It mainly focused on the arrival traffic weather avoidance routing and metrics associated with such routes. In [10] the authors developed the weather-specific Coded Departure Routes (CDRs) for pre-departure flights. The resulted CDRs were generated under current weather predictions. However, some of the CDRs may not be feasible. [11] studied ground traffic rerouting with a future congestion prediction along the current route. For ground traffic, any generated reroutes will be feasible because the network is built on existing roadways.

[12] proposed a dynamic rerouting algorithm which generates flexible and acceptable reroutes. The research in [12] invited the subject matter experts to help define the metrics of reroute acceptability and the network model was built based on acceptable flight routes from historic data.

The only metroplex routing research we can found is [13] which describes a mixed integer linear programming formulation and solves the routing and scheduling problems at the same time for the flights whose destinations are metroplexes. The authors also formed the problem to maintain its computational feasibility and analysed the resulted routes.

The aim of this work is to first introduce the concepts of metroplex and flexible flight and to illustrate the dependency between the airports in the same metroplex. Secondly, we present the Jetway Plus Network model and the metroplex routing algorithm which considers the sector congestion as the constraint. The routing is followed by a McTMA scheduler solving the runway assignment problem. To our best knowledge, this is the first work that studies the metroplex routing under the sector congestion constraint. In addition, the algorithm implementation is explicitly shown step by step and the simulation for one flexible flight is visualized by using FACET interfaces [14].

The rest of this paper is organized as follows. The second section introduces the concepts of metroplex and flexible flights. Also the dependency metric of the metroplex airports is illustrated in detail. The third section shows how the network model is established for running our metroplex routing algorithm. In the fourth section, the metroplex routing algorithm under the sector congestion constraint is presented and the runway scheduler following the routing algorithm is also simply introduced. The routing algorithm simulation is performed in Section V and Section VI concludes the paper.

II. Metroplex and Flexible Flight

A. The Concept of Metroplex

A metroplex is a region with several close airports which share traffic resources such as airspace, ground transportation. More rigorously, a metroplex consists of several close airports with consequential dependencies. Each metroplex subsists in a system-of-systems which includes the airports, the flights and ground traffic between them, the airline companies, Air Traffic Control (ATC) services etc. Fig. 1 illustrates the types and layers of networks in National Airspace System (NAS) and how they interact with the metroplex operations [3]. Studying a composite network combining the service networks of airline companies will provide insights into NAS-wide traffic between the various airports as nodes in this network (δ-level in Fig. 1).

Another level in this network leads to a scenario in which airports in a metroplex can be considered as "modules" which are connected to other metroplex modules and airports (γ-level in Fig. 1). The airports within each module are dependent on each other operationally and possibly economically. At the same time they are connected to the other parts of NAS by all means of airline services. The "external" interactions between a metroplex module and the other parts of the NAS are different from those dependencies "internal" to the metroplex module. In order to understand this difference, metrics are developed to characterize and quantify the dependencies between metroplex airports.

Studying these internal dependencies inside a module is critical for proper implementation of our flexible operation concept on metroplexes. Therefore the dependency metrics and related analyses have been developed and the major results are presented as the following.

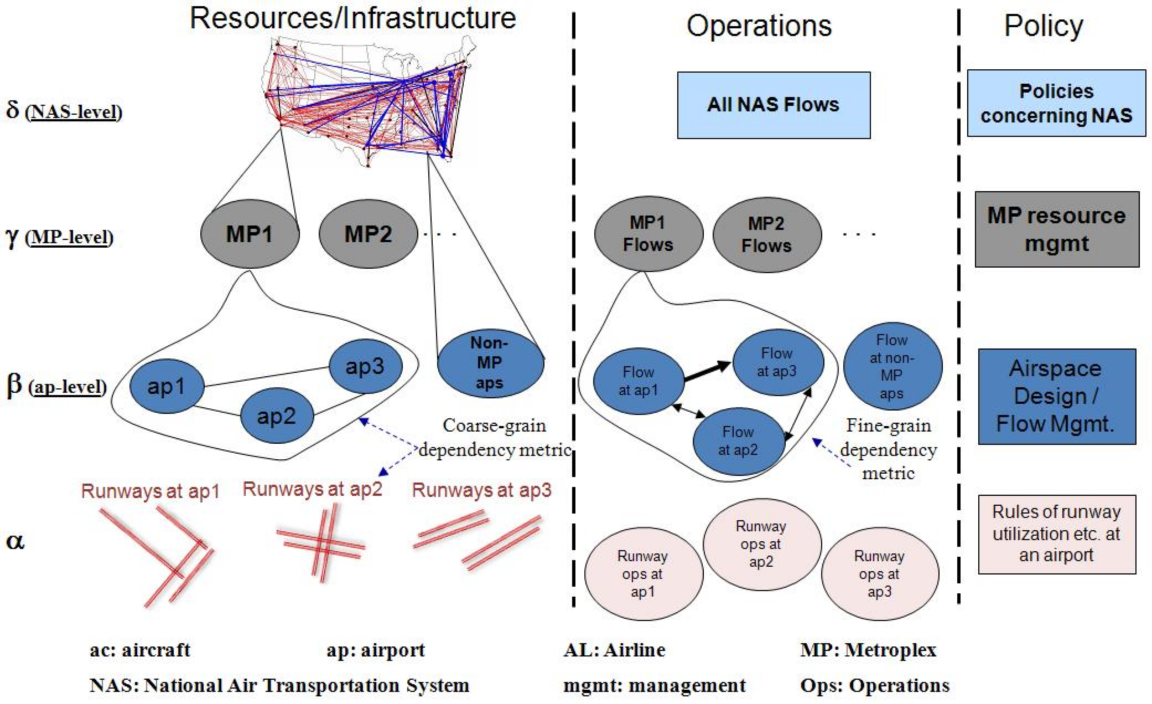


Figure 1: Interactions of hierarchical networks in NAS.

1. Characterization of Metroplex Dependencies

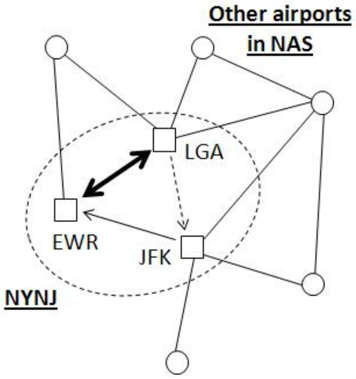


Figure 2: The major airports in New York and New Jersey.

The premise for quantifying dependencies is to observe how dependencies vary from one metroplex to another. Fig. 2 is a simplified view of NYNJ metroplex with its three major airports JFK, LGA and EWR.

The solid-black lines represent flights to and from NYNJ to the remaining parts of NAS. The arrows between the three airports are the "internal" interactions within this metroplex which bring dependencies on multiple dimensions (Table 1). For example, connecting passengers flying into LGA for their international flight from JFK require ground transportation, which is an operational dependency. From the perspective of the Port Authority of New York and New Jersey (PANYNJ), the ownership of all three NYNJ airports provides better economic synergies than the single ownership. However, many other metroplexes have multiple owners e.g., Northern California metroplex, which create different types of dependencies. Furthermore, policies and regulations at one airport will affect other airports in the same metroplex. For instance, all airports in a metroplex must agree to participate in the flexible arrivals policy and flexible flight operating. In summary, there are various interactions among many dimensions of dependencies as shown in Table 1.

Table 1: Dimensions of metroplex dependencies

Dimension	**Example Attributes**
Physical	Airport proximities, runway configurations, airspace geometry
Economic	Ownership of airports, landing fees, lease of terminals
Operational	Airport operations, airlines, capacities, passenger ground-link
Regulatory	Slot-controls, congestion-pricing, noise regulations, carbon credits

2. Dependency Metrics

In [3] the authors introduced two types of dependency analysis: Constructive and Observational. Constructive analysis deals with the physical resources of a metroplex airport (e.g., runway configurations) and their influence on operations at the metroplex. Observational analysis describes the study of metroplex operational data by observing how dependencies influence the metroplex operations. In this paper we introduce the development of a constructive coarse grain metric and show how it can be used to implement the flexible flight (flexible arrival) concept.

3. Constructive Coarse Grain Dependency Metrics

These metrics are based on simple coarse grain definitions developed to qualitatively analyse how runways influence traffic patterns and cause dependencies (Table 2). The analyse is done by quantifying the relative importance of each runway. The runway with the most traffic in a year is chosen as the reference runway (RR) and the metric is developed based on RR (shown in Table 2). The dependency metric of a runway is a composite of several runway features (e.g., length, orientation and proximity) and the traffic on each runway; a high value of the metric indicates larger dependencies. The metric details are not repeated here but are described explicitly in [3]. We extended these metrics to a metroplex by considering a metroplex as a "super airport" with its runways largely separated from the runways of other airports in the same metroplex. The reference runway in this case is the runway with the most traffic in the metroplex. Data for runway traffic is obtained from Aviation System Performance Metrics (ASPM) database for each airport.

4. Results of Dependency Characterization Study

In addition to NYNJ, Northern California (NoCal) metroplex is analysed to understand how dependencies differ from one metroplex to another. NoCal contains San Francisco International (SFO), Mineta San Jos International (SJC) and Oakland International (OAK) airports.

Results from computing coarse grain dependency metrics (from 2002 to 2007) are shown in Table 2. Data in Table 2 indicate that there was no significant difference between NYNJ and NoCal. The minor differences can be due to the difference in their total traffic or the definition of the metrics. This indicates that similar factors cause dependencies at these two metroplexes. It is found that the metric values vary with the choice

Table 2: Runway dependency metrics for NYNJ and NoCal

Year	**Metroplex Traffic**		**Reference Runway**		**RDM-M** ($d_l * d_o * d_p$)		**RDM-M** ($d_l + d_o + d_p$)	
	NYNJ	NoCal	NYNJ	NoCal	NYNJ	NoCal	NYNJ	NoCal
2007	1,233,473	697,489	JFK 31L	SFO 28R	0.359	0.139	1.927	1.459
2006	1,184,445	678,501	EWR 22R	SFO 28L	0.191	0.15	1.495	1.378
2005	1,141,250	666,660	EWR 22L	SFO 28R	0.177	0.138	1.421	1.48
2004	1,113,671	673,929	EWR 22R	SFO 28R	0.199	0.139	1.476	1.481
2003	1,016,652	619,400	EWR 22L	SFO 28R	0.197	0.1	1.442	1.252
2002	913,420	620,531	EWR 22L	SFO 28R	0.206	0.001	1.443	0.426

of reference runway (RR). For NYNJ, from 2002 to 2006, RR was located at EWR whereas it was at JFK in 2007. This was because of a significant increase in JFK traffic in 2007. Due to the RR location shift, runway proximity effect plays an critical role in causing a significant change in the dependency metric (Table 2). 2002 data is not available to estimate the traffic per runway at OAK and SJC, causing NoCals dependency metrics for this year to significantly differ from the other years.

Atlanta International (ATL) and Denver International (DEN) airports are analysed to understand how dependencies differ between metroplex and non-metroplex airports. In particular, ATL and DEN had operations comparable in volume to JFK and SFO (NYNJ and NoCal respectively), many airlines operate at each of them, and both have problems of delays and congestion. They are good candidates for this comparative study. In the case of non-metroplex airports, the dependency metrics changed considerably for ATL from 2005 to 2006 (Table 3). This is due to operation of runways from 2006. Decrease in dependency metrics shows that physical dependencies are less in 2006 and 2007, causing "decoupling" of runway operations at ATL. On the other hand, though a new runway was added to DEN in 2004, our metrics could not detect any significant change in DEN's dependencies.

Table 3: Runway dependency metrics for ATL and DEN

Year	**Airport Traffic**		**Reference Runway**		**RDM-A** ($d_l * d_o * d_p$)		**RDM-A** ($d_l + d_o + d_p$)	
	ATL	DEN	ATL	DEN	ATL	DEN	ATL	DEN
2007	962,390	594,001	ATL 27R	DEN 17R	0.024	0.051	1.032	0.985
2006	945,018	570,757	ATL 27R	DEN 8	0.024	0.057	1.034	1.023
2005	945,946	535,267	ATL 27R	DEN 8	0.049	0.059	1.315	1.19
2004	936,677	534,661	ATL 27R	DEN 17R	0.049	0.066	1.309	1.03
2003	874,647	474,688	ATL 27R	DEN 17R	0.049	0.061	1.314	0.983
2002	836,262	397,938	ATL 27R	DEN 17R	0.049	0.058	1.326	0.956

5. *Summary*

From the high dependency metric value in Table 2 and the low dependency metric value in Table 3. We can see there are more dependencies among airports in a metroplex than non-metroplex airports. The shared resources of the metroplex airports link them together as a system. Therefore the concept of the metroplex can be considered as a super destination airport for flexible flights.

B. Flexible Flights

A flexible flight is initially routed towards a metroplex instead of a pre-determined destination airport. Unlike the regular flight having a fixed destination airport, when the flight arrives the decision boundary of the destination metroplex, it will receive the information of which runway of which airport to land in.

The passengers taking a flexible flight accept arriving at one of several airports in a great metropolitan area without caring about the specific destination airport.

1. Benefits of the Flexible Flights

The metroplex and flexible flights concept can bring many benefits to both passengers and traffic controllers with existed airports and ground traffic infrastructures. Developing and analysing the concept of flexible operations at a metroplex and determining whether the alternate, flexible flight plan have significant potential operational and economic value. Some of the potential benefits are: a) Enhancing metroplex airport throughput without extensive infrastructure investment; b) Increasing robustness to disruptions; c) Providing maximal flexibility to passengers.

2. How to operate a flexible flight?

To operate a flexible fight from departure to landing includes two parts. The first part is to route the flexible flight towards its destination metroplex similarly like to route a regular flight to its destination airport. In this part we just need to treat a metroplex as a super airport. The second part is when the flexible flight arrives the decision boundary of the metroplex, it will be scheduled to some certain runway at one of the metroplex airports. However, until the runway scheduling decision is made, neither the operator nor the ATC system will know the exact destination airport. This paper mainly deals with the first part which is the metroplex routing algorithm while the second part scheduling will be simply introduced.

In practice the "decision boundary" consists of a series of metering fix points [15] for the ease of operation. In our algorithm, the "decision boundary" is considered as a polygon around the metroplex as shown in Fig. 3, where the blue polygon is the decision boundary of NYNJ metroplex.

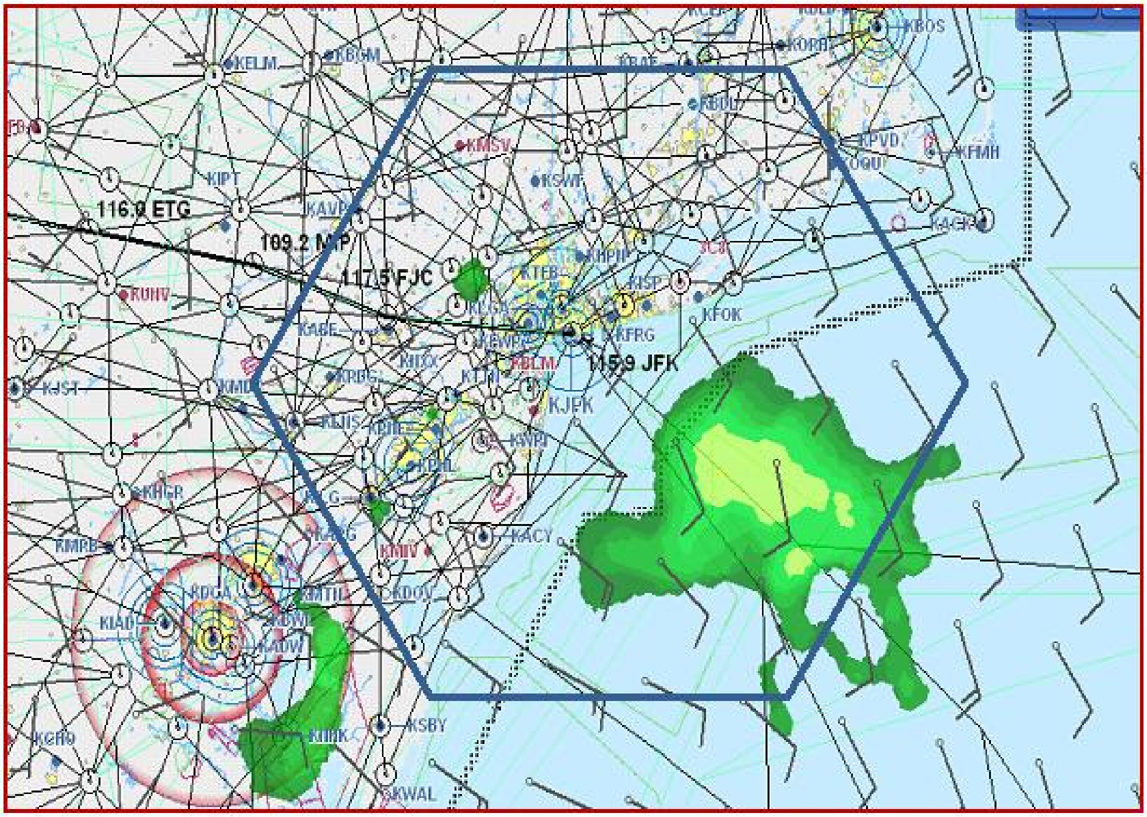


Figure 3: Decision boundary of NYNJ metroplex.

III. Jetway Network and Jetway Plus Network

The approach in this paper is based on a weighted shortest path search. A network model is required in order to design an efficient and effective metroplex routing algorithm. The waypoint-based network is exploited as the data structure to generate the optimal route for each flexible flight under sector congestion constraint. The initial network built from waypoints provided by FACET is named as Jetway Network. Another model called Jetway Plus Network is formulated by inserting additional waypoints and all airports information, which is used as the final data structure for running our algorithm.

A. Obtain the Waypoints and Jet Routes Information from FACET

To accurately capture the real flight paths that air traffic follow, a jet route-based waypoint network is built. Firstly, jet route and waypoint information are obtained from FACET. Using the *NavigationInterface* in FACET, the waypoint identifiers (index) and locations (longitude and latitude) are obtained for all waypoints. Also using the *NavigationInterface*, the sequence of waypoints that defines a jet route is obtained for all jet routes. To convert a jet route description in terms of waypoint locations to waypoint indices, the corresponding index is identified for each waypoint on a jet route using the waypoint information from FACET.

To form the network, we record the connection (edge) between two consecutive waypoints (vertices) for each jet route. Intersection points on the jet route system are automatically considered as waypoints when recording the connection along a jet route because an intersection point will be part of the waypoint sequence for every jet route sharing that intersection point. Figure 4 illustrates how edges are obtained from a jet route, along with intersection points. Additional edges were created to connect airports to the jet route system. Collectively, the edges and vertices define the graph. The edges are not directed because a jet route can support traffic in both directions (in different altitudes).

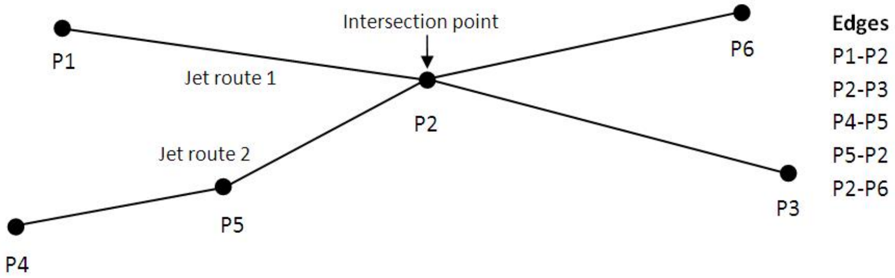


Figure 4: Obtaining graph edges from jet route, including intersection points.

B. Establish the Jetway Network

In order to take advantage of the Boost C++ Graph Library (BGL) [16], we transform the obtained waypoints and the jet routes data into the BGL format. The major contribution of the Boost Graph Library (BGL) is a generic interface that allows access to a graph's structure, but hides the details of the implementation. This is an "open" interface in the sense that any graph library that implements this interface will be interoperable with the BGL generic algorithms and with other algorithms that also use this interface. The BGL provides some general purpose graph classes that conform to this interface.

According to the requirement of C++ BGL, we have each waypoint provided by FACET in a C++ *struct* and put all these waypoint structs in an indexed C++ *map.* Each struct contains the name, latitude, longitude information of the waypoint. The waypoints serve as the nodes in a Jetway Network graph.

Each direct route between two nodes is considered as two directional edges in a Jetway Network graph and each of the two is recorded in a C++ struct which contains the starting point and the ending point. All the edges are also stored in an indexed C++ map. At the beginning of the network construction, the great circle distance of each edge is calculated and recorded as the non-negative weight in each edge struct.

By using *boost::add_vertex* and *boost::add_edge* functions from BGL, a Jetway Network graph is built based on the waypoints and direct routes as nodes and edges respectively. Additionally the weight of each edge is set into *boost::weight_map(pmWeigth)*. Finally the Dijkstra Algorithm *boost::dijkstra_shortest_paths* will take care of the optimizing job and output the distance and predecessor of each node after the input of origin airport is set. In other words, the Dijkstra Algorithm from BGL provides the results of all the shortest paths from the origin airport to every other airport except those are disconnected from the origin.

C. Hash Table

In order to insert new waypoints and dynamically maintain the network, the waypoints need to be indexed by a hash table. The similar hash table is built for direct route data. For simplicity, we only show how to build the hash table for waypoints in this section. The hash key in this paper is the rounded positive latitude-longitude pair of each waypoints, in other words, each waypoint's latitude and longitude are changed to their absolute values and then rounded into integers. Then the positive integers are stored in a hash key pair $<lat, lon>$. All of the existed waypoints in Jetway Network are processed to acquire their hash key pair. Initially the hash table H is set to be empty ($\emptyset$). The hash key pair $<lat, lon>$ is calculated according to the latitude and longitude of a waypoint WP_i and a search is started in the current hash table H based on $<lat, lon>$. If this hash key pair is not in H, a $<lat, lon>$ is created in H and this waypoint WP_i is inserted under the newly created hash key pair; if it is already in the hash table, compare WP_i to all the waypoints under this existed hash key pair, insert WP_i if no same waypoint is found. Repeatedly all the waypoints obtained from FACET have their hash indexes.

After we have all the waypoints in the hash table, it is very fast to insert new waypoints. In another aspect, building the hash pair index is in fact to divide the NAS into grids. Each grid is a square with four corners whose latitude and longitude are integers as shown in Fig. 5, where i and j are integers. When we have a new waypoint to be inserted, we first calculate its hash key pair and decide which grid it is in. As we use *floor* to obtain $<lat, lon>$ from the absolute latitude and longitude, if the waypoint exists, it should be in the grid with top-left corner coordinates (i, j)=$<lat, lon>$. Therefore only the existed waypoints $\mathrm{WP}_1, \mathrm{WP}_2, \mathrm{WP}_3$ in this grid need to be checked instead of checking all the waypoints existed in the entire Jetway Network. This is the mechanism of the hash table and it substantially enhances the speed of inserting new waypoints.

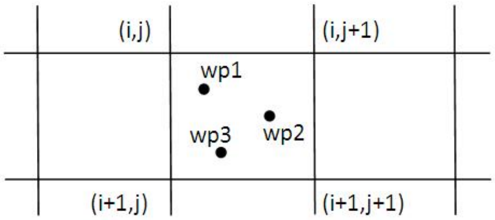
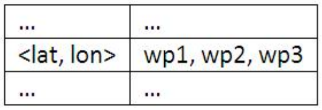


Figure 5: Hash table indexed grids enhance the search of waypoints.

The grids in hash table are stored with sorted hash key pair as shown in Fig. 6. The grids are saved in ascending order of the first integer of the hash key pair, and with the same first integer, the grids are recorded in ascending order of the second integer of the hash key pair. The waypoints in each grid are recorded under the hash key pair $<lat, lon>$ which is also actually the top-left coordinates (i, j) of the corresponding grid. If there is no waypoint in a grid, that grid will not be stored in the hash table.



...	...
<lat, lon>	wp1, wp2, wp3
...	...

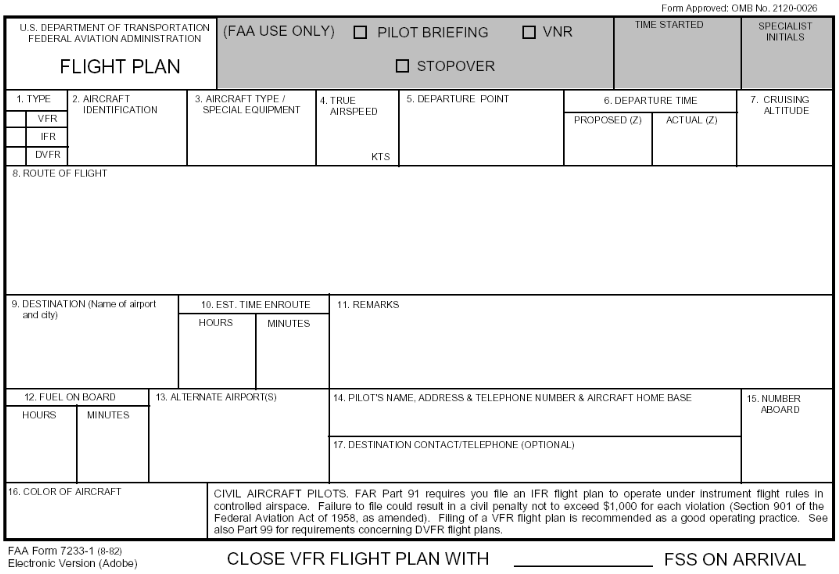
Figure 6: Hash table is stored with ordering.

D. Establish the Jetway Plus Network

After we have the Jetway Network, we can optimize the routes for input flight plans of flexible flights. Since the waypoints offered by FACET may not be complete, the new waypoints mentioned in the input flight plans need to be imported to the Jetway Network. At the same time, all the airports are imported by treating them as waypoints too. The additional edges are obtained from FACET by the historic flight track files. Together all the additional inserted elements with the Jetway Network formulate the Jetway Plus Network, which is the final data structure to apply our routing algorithm.

1. Input Flight Plans

Flight plans are documents filed by pilots or a Flight Dispatcher with the local Civil Aviation Authority (e.g. FAA in the USA) prior to departure. They generally include basic information such as departure and arrival points, the sequence of waypoints of the flight route, estimated time en route, alternate airports in case of bad weather, type of flight (whether instrument flight rules or visual flight rules), pilot's name and number of people on board. The standard FAA flight plan form is shown in Fig. 7.



Form Approved: OMB No. 2120-0026

U.S. DEPARTMENT OF TRANSPORTATION
FEDERAL AVIATION ADMINISTRATION

FLIGHT PLAN

(FAA USE ONLY) □ PILOT BRIEFING □ VNR
□ STOPOVER

TIME STARTED

SPECIALIST INITIALS

1. TYPE
□ VFR
□ IFR
□ DVFR

2. AIRCRAFT IDENTIFICATION

3. AIRCRAFT TYPE / SPECIAL EQUIPMENT

4. TRUE AIRSPEED
KTS

5. DEPARTURE POINT

6. DEPARTURE TIME
PROPOSED (Z) ACTUAL (Z)

7. CRUISING ALTITUDE

8. ROUTE OF FLIGHT

9. DESTINATION (Name of airport and city)

10. EST. TIME ENROUTE
HOURS MINUTES

11. REMARKS

12. FUEL ON BOARD
HOURS MINUTES

13. ALTERNATE AIRPORT(S)

14. PILOT'S NAME, ADDRESS & TELEPHONE NUMBER & AIRCRAFT HOME BASE

15. NUMBER ABOARD

17. DESTINATION CONTACT/TELEPHONE (OPTIONAL)

16. COLOR OF AIRCRAFT

CIVIL AIRCRAFT PILOTS. FAR Part 91 requires you file an IFR flight plan to operate under instrument flight rules in controlled airspace. Failure to file could result in a civil penalty not to exceed $1,000 for each violation (Section 901 of the Federal Aviation Act of 1958, as amended). Filing of a VFR flight plan is recommended as a good operating practice. See also Part 99 for requirements concerning DVFR flight plans.

FAA Form 7233-1 (8-82)
Electronic Version (Adobe)

CLOSE VFR FLIGHT PLAN WITH ______________ FSS ON ARRIVAL

Figure 7: Standard FAA flight plan form.

2. Obtain and insert new waypoints from the input flight plans

From the input flight plans which need to be optimized, firstly we obtain the sequence of waypoints of the flight route and insert the waypoints into the hashed Jetway Network. The related edges are also added in. Secondly for each flight we obtain its origin-destination pair because our routing algorithm needs them to generate the optimal reroute.

Note that the name of each waypoint FACET provides is a five-digit integer beginning with a 1 or 2 while the standard flight plan contains a sequence of the waypoints which are described by a three-letter combination. The FACET has a method called *getWaypointName* inside *NavigationInterface* to translate one expression to the other.

In detail, for each flight plan we first extract the sequence of the waypoints, then the FACET interface is used to translate these waypoints into five-digit integers. For each waypoint, we obtain its latitude-longitude pair and search it in the existed Jetway Network hash table. If this waypoint exists, continue to process the next waypoint of the current flight plan; if this waypoint is new, insert it to the hash table and create a new hash key pair by rounding the absolute latitude-longitude pair of this waypoint. After all of the flight plans are processed and all the airports are inserted, the Jetway Plus Network is formulated.

IV. Flexible Flight Routing Algorithms

A. Shortest Path Algorithm without Sector Congestion Constraint

The waypoint-based Jetway Plus Network obtained above is the routing data structure for the entire NAS. When the routing algorithm does not take the sector congestion into account, the weight of each edge is the great circle distance between two waypoints. The Dijkstra Algorithm [17] is used to find the shortest path from origin to destination. Since the flexible flights are considered, the destination airport of each flight is selected from the airports inside its destination metroplex. In this research, the *central airport* of a metroplex is chosen as the destination of a flexible flight.

1. Metroplex Central Airport

Since we treat the metroplex as a super airport, the geometric center of the metroplex should represent the whole metroplex when the routing algorithm is operated. Therefore, we calculate the central point c of the airports in the metroplex and set the closest airport to c as the "central" airport of the whole metroplex. The central airport will represent the destination metroplex when the flexible flights are routed to this metroplex. For example, the NYNJ metroplex has three airports, JFK, LGA, EWR. Firstly the geometric center is found by $c = (ap_1 + ap_2 + ap_3)/3$, where c and ap_i are both latitude-longitude pair $<lat, lon>$. Then the closest airport to the central point c is selected as the "central" airport of NYNJ. In this case, LGA is chosen. The three NYNJ airports are shown in Fig. 8.

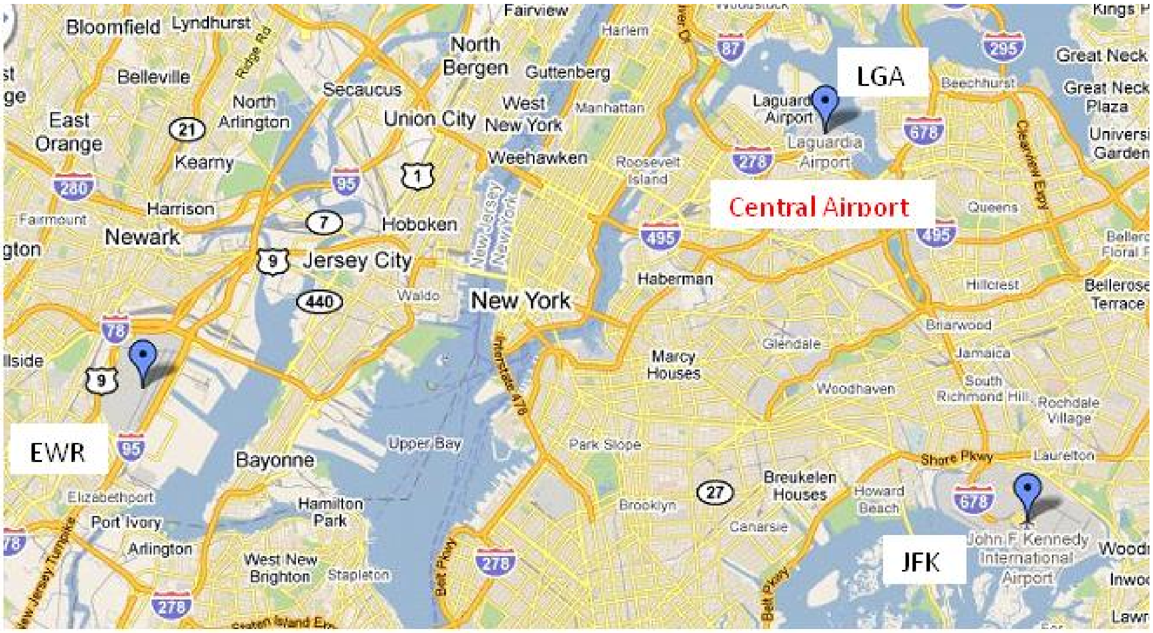


Figure 8: Three major airports in NYNJ metroplex.

2. McTMA Scheduler

In this research our algorithm generates the optimized route for a flexible flight while another team works on the runway scheduling based on McTMA. The airport/runway balancer is designed particularly for the metroplex destination to facilitate McTMA with flexible flights scheduling. The idea of their algorithm is to consider all the runways resource from all the airports in the same metroplex together. Since the regular destination-fixed flights already take most of the runway time slots, the airport/runway balancer checks the available time slots (holes) of all the runways for flexible flights. Once there is a slot available in any one or more runways of a metroplex airport, the approaching flexible flight will be scheduled to the closest runway. The airport/runway balancer can access McTMA through the interfaces shown in Fig. 9. Fig. 9 also shows the six steps of a complete metroplex runway scheduling.

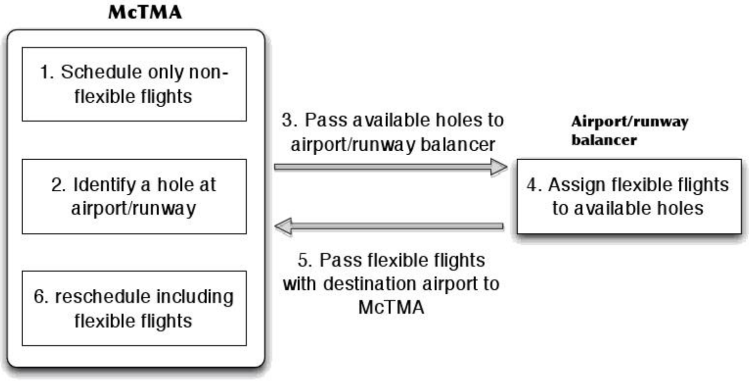

McTMA
1. Schedule only non-flexible flights
2. Identify a hole at airport/runway
6. reschedule including flexible flights
3. Pass available holes to airport/runway balancer
Airport/runway balancer
4. Assign flexible flights to available holes
5. Pass flexible flights with destination airport to McTMA

Figure 9: Interactions between McTMA and airport/runway balancer.

B. Shortest Path Algorithm with Sector Congestion Constraint

To introduce sector congestion constraint, the edges that intersect those congested sectors will be assigned with an extra huge weight. The congested sectors or the congested areas in a sector are depicted as polygons in this paper in order to easily check the intersections with the edges in Jetway Plus Network. Therefore the resulted shortest path represents the shortest weather conflict free route from origin to destination.

For advanced multiple-level sector congestion, the sector congestion impact is taken into account by assigning the edges with additional weights of different congestion statuses. More congested is a sector, the heavier weight it is on each intersecting edge. The congestion weight will be summed up with the great circle distance weight for each edge and then the shortest path algorithm is applied in this new weighted Jetway Plus Network. The detail of the weighting scheme will be discussed in the following.

1. Hourly Broadcasting Sector Congestion

Since the congestion's change is not very rapid in reality, the sector congestion status can be forecasted and published hourly by FAA or other professional agencies. The congestion status report imported into the Jetway Plus Network model is used to update the edge weights and the metroplex routing in next one hour is calculated based on these new weights.

The congestion status is not necessarily published in shape of sector. To be more precise, the congestion can be reported as a smaller and more accurate area within a sector, which makes it look like the weather forecast in Fig. 12.

2. Weighting Scheme

In order to be combined together as the new weight, the great circle distance weight and the sector congestion weight need to be normalized. To normalize each great circle distance weight w_{α_e} on each edge e, first find the largest great circle distance weight $w_{\alpha_{\max}}$, then the normalized great circle distance weight is $w'_{\alpha_e} = w_{\alpha_e}/w_{\alpha_{\max}}$. The impact of sector congestion on related edge will be added as a penalty weight w'_{β_e}, which is 1 if the edge intersects with any congestion area and 0 if no congestion area covers this edge.

The final weight w_e consists of the effects from w'_{α_e} and w'_{β_e}, and a coefficient γ is introduced to adjust the strengthen of the penalty weight in Eqn. 1. The larger γ places more emphasis on sector congestion avoidance in the algorithm.

$$w_e = w'_{\alpha_e} + \gamma \cdot w'_{\beta_e} \tag{1}$$

To be more accurate, the sector congestion can be described in different statuses. A multi-level rating of $0, 1, 2$ is used to describe "No congestion, Light congestion, Heavy congestion", whose number of levels can be adjusted when it is applied in practice. The only difference between multi-level rating and on-off rating is that w'_{β_e} will need to be normalized similarly as w'_{α_e} for multi-level rating.

One problem for multi-level rating is that sometimes an edge may intersect several areas with different congestion ratings as Fig. 10. In that case, the largest congestion rating on this edge will be picked as w_{β_e}. In the case of Fig. 10, the direct route will be assigned 2 (Heavy Congestion) as its penalty weight.

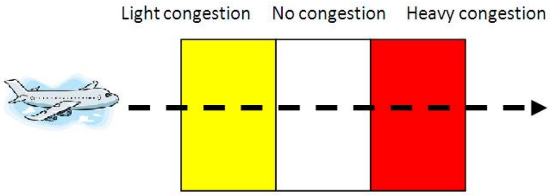


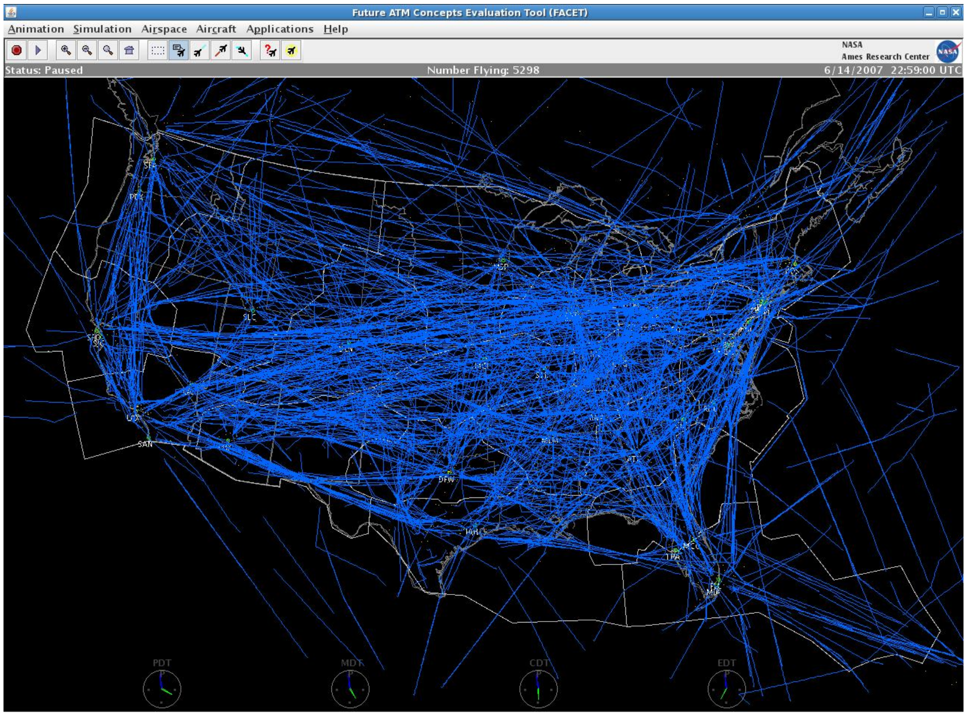
Figure 10: An edge intersects with three different rated congestion areas.

V. Simulation

A. Platform

FACET is a simulation and analysis tool designed by NASA to evaluate future ATM concepts and methods. FACET can be used through a graphical user interface (GUI) or by using the application programming interfaces (APIs) in the Java programming language. Some of FACET's functionalities used in this research include playing back recorded trajectories, simulation results and reading, writing and processing various NAS datasets such as sector capacities, aircraft performance data, jet route information, etc. The FACET APIs are also used extensively in this research because it allows for greater functionalities compared to the GUI.

The trajectory datasets provided by NASA contain trajectory information such as flight plan, longitude, latitude, altitude and speed, for every flight in the NAS. Using the FACET APIs, the flight plans are extracted from the dataset to be input to our program. In this research, FACET is used to compute the flight trajectories by given the optimized flight plans.



Status: Paused

Number Flying: 5298

6/14/2007 22:59:00 UTC

Figure 11: FACET simulated aircraft trajectories.

FACET contains an aircraft performance model database with 66 different aircraft types, along with an equivalent list that maps over 500 aircraft types to these 66 models. FACET also contains various datasets used to model the NAS, such as center/sector boundaries, sector capacities and jet route system/waypoints information. FACET has two routing options [14], the Direct Routing (DR) option and the Flight Plan Routing (FPR) option. The DR option computes a great circle route directly from origin to destination. The FPR option computes a route that connects waypoint to waypoint, from origin to destination. Segments connecting the waypoints are great circle routes. The default simulation time interval, which was used in this research, is 1 minute.

The FPR option would be the more suitable option for modelling current air traffic patterns since most of today's flights do not fly a direct route from origin to destination. The DR option would be more suitable for studying free flight operations. The FPR option is used in this research.

The FACET APIs along with other user written Java methods are used to simulate flight trajectories. This approach runs faster because the GUI is not invoked to display the trajectories as they are computed. Also, this approach allows the user to compute only the desired parameters instead of computing all parameters (the default GUI option), thus further reducing computing time. Using this approach, the complete trajectory (from origin to destination) for all flights within a 2-hour period (planning time horizon) can be computed in a few minutes on a standard desktop computer. Figure 11 shows the simulated trajectories of all aircraft ($n > 5000$) above 18000ft, between 2200 and 2300 hours on June 14th, 2007.

B. Optimized Flight Plan

To simulate a flight in FACET, a flight plan has to have more than just the route information (waypoint sequence). Other parameters such as commanded altitude, groundspeed and take-off time have to be specified as well. Therefore, to generate a flight plan for the optimized route, the original flight plan is modified by substituting in the optimized route information. All other parameters remain unchanged.

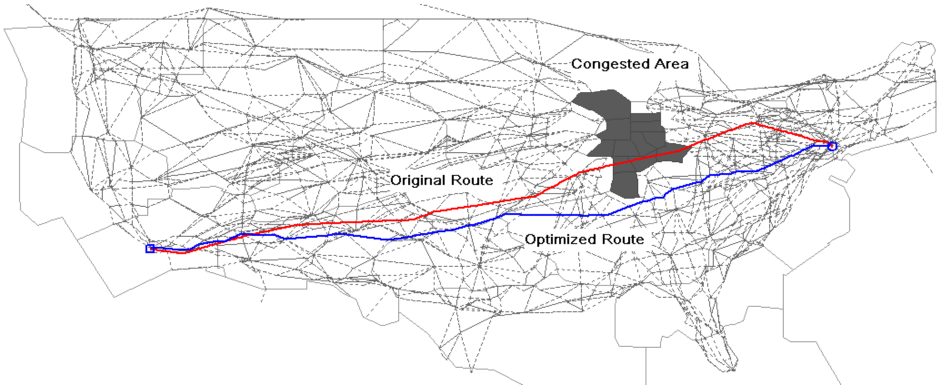


Figure 12: The original and optimized routes for a flexible flight.

For ease of observation, only one flexible flight is studied in Figure 12. The optimized flight plan is generated under the sector congestion constraints and is compared to its original flight plan . The congested sector is set to be the dark area in Figure 12. The origin airport is LAX and the destination metroplex is NYNJ whose central airport is LGA. The weighting scheme is on-off weighting and $gamma = 10$. The result shows that the optimized route successfully avoids the congested area in Figure 12.

VI. Conclusion

This paper first introduces the concept of metroplex and flexible flights and then the analysis shows that the airports within the same metroplex have dependency on each other, which makes the metroplex concept a potential solution for improving the throughput of the metroplex airports and increasing the flexibility for passengers. Moreover, the routing algorithm for flexible flights is developed under sector congestion constraint based on weighted shortest path algorithm. The routing algorithm is applied on a data structure called Jetway Plus Network which is built from waypoints and jetways. During the implementation process, the FACET from NASA Ames and its interfaces facilitate the data acquisition for establishing the Jetway

Plus Network and the algorithm simulation. The algorithm performance turns out to be efficient and effective for the metroplex routing and after the flexible flight arrives the decision boundary of the metroplex, the McTMA scheduler will assign a runway to it. The routing and scheduling together constitute a complete flexible flight operation.

Acknowledgements

The authors would like to thank NASA for funding under grant NRA NNL10AA15C. The valuable contributions from Prof. Daniel DeLaurentis, Prof. Steven Landry and their research teams are appreciated.

References

[1] JPDO, "Concept of Operations for the Next Generation Air Transportation System," June 13 2007, v2.0.

[2] Atkins, S. and Engelland, S., "Observation and Measurement of Metroplex Phenomena," *Digital Avionics System Conference*, October 2008.

[3] DeLaurentis, D. and Ayyalasomayajula, S., "Analysis of Dependencies and Impacts of Metroplex Operations," October 2010, Purdue University.

[4] McClain, E. and Clarke, J. P., "Traffic Volume Intersection Metric for Metroplex Clustering Analysis," *AIAA Guidance, Navigation, and Control Conference*, Aug 2009.

[5] Wang, L., Donohue, G., Hoffman, K., Sherry, L., and Oseguera-Lohr, R., "Analysis of Air Transportation for the New York Metroplex: Summer 2007," *Proceedings International Conference on Research in Air Transportation*, Fairfax, VA, Febuary 2008.

[6] Donohue, G., Hoffman, K., Wang, L., and DeLaurentis, D., "Metroplex Operations: Case Study of New York Metroplex Air Transportation System," *NASA Airspace Systems Program*, Austin, TX, March 2008.

[7] Schaar, D., Sherry, L., and Donohue, G. L., "Rethinking Airport Improvement: Analysis of Domestic Airline Service to U.S. Metroplex Airports," *CASTR Internal Report CASTR-01-2010*, March 2010.

[8] Bokadia, S. and Valasek, J., "Sever Weather Avoidance Using Informed Heuristic Search," *AIAA Guidance Navigation and Control Conference and Exhibit*, Montreal, CA, August 2001.

[9] Krozel, J., Penny, S., Prete, J., and Mitchell, J., "Automated Route Generation for Avoiding Deterministic Weather in Transition Airspace," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 1, January 2007, pp. 144–153.

[10] Krozel, J., Prete, J., Mitchell, J., Smith, P., and Andre, A., "Designing On-Demand Coded Departure Reroutes," *AIAA Guidance Navigation and Control Conference and Exhibit*, Keystone, CO, August 2006.

[11] Weyns, D., Holvoet, T., and Helleboogh, A., "Anticipatory Vehicle Routing using Delegate Multi-Agent System," *Proceedings of the 2007 IEEE Intelligent Transportation Systems Conference*, Seattle, WA, October 2007.

[12] Taylor, C. and Wanke, C., "Dynamic Generation of Operationally Acceptable Reroutes," *9th AIAA Aviation Technology, Integration, and Operations Conference (ATIO)*, Hilton Head, South Carolina, September 2009.

[13] Capozzi, B. and Atkins, S., "Dynamic Generation of Operationally Acceptable Reroutes," *9th AIAA Aviation Technology, Integration, and Operations Conference (ATIO)*, Hilton Head, South Carolina, September 2009.

[14] Bilimoria, K. and Sridhar, B., "FACET: Future ATM Concepts Evaluation Tool," *3rd USA/Europe Air Traffic Management R&D Seminar*, Napoli, Italy, 2000.

[15] Idris, H., Evans, A., and Evans, S., "Single-Year NAS-Wide Benefits Assessment of Multi-Center TMA," 2004, Titan Corporation Air Traffic Systems Division.

[16] *Boost Graph Library, The User Guide and Reference Manual*, Addison-Wesley Professional, 1st ed., Dec 20 2001.

[17] E.W.Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, Vol. 1, 1959, pp. 269–271.