Abstract—This paper discusses a comparison of several algorithm generated airspace boundary designs, known as sectorizations. Three algorithms are chosen that approach the airspace sectorization problem in different ways and produce radically different looking sectorizations due to the disparity in their methods. Simulations of air traffic using each of the sectorizations is completed and their resulting demand, capacity, complexity, and delay metrics are compared. Results identify strengths and weaknesses of each sectorization algorithm.

Keywords—airspace design; capacity management; dynamic airspace configuration

NOMENCLATURE

\begin{align*}
i & = \text{flight index} \\
s & = \text{sector index} \\
r & = \text{region comprised of a group of sectors} \\
t_e(s,i) & = \text{flight i egress time from sector } s \\
t_i(s,i) & = \text{flight i ingress time into sector } s \\
t_u(i) & = \text{flight i unconstrained gate arrival time} \\
t_c(i) & = \text{flight i constrained gate arrival time} \\
N_f(s) & = \text{the set of all flights within sector } s \\
N_f(r) & = \text{the set of all flights flying through region } r \\
N_s(r) & = \text{the set of all sectors in region } r \\
n_f(s) & = \text{total number of flights in set } N_f(s) \\
n_f(r) & = \text{total number of flights in set } N_f(r) \\
n_s(r) & = \text{total number of sectors in set } N_s(r) \\
c(s) & = \text{capacity (MAP value) for sector } s \\
\delta(s) & = \text{average time a flight spends in sector } s \\
m(s,k) & = \text{average flight count in sector } s \text{ for quarter-hour } k \\
m(s) & = \text{average flight count for mid 8 hours of sector } s \\
M(r) & = \text{average } m(s)\forall s \in N_s(r) \\
\sigma_m(r) & = \text{standard deviation of } m(s)\forall s \in N_s(r) \\
\Sigma_c(r) & = \text{sum of } c(s)\forall s \in N_s(r) \\
C(r) & = \text{average } c(s)\forall s \in N_s(r) \\
\sigma_c(r) & = \text{standard deviation of } c(s)\forall s \in r \\
\rho(s) & = \text{flight count/capacity ratio for sector } s \\
P(r) & = \text{average } \rho(s)\forall s \in N_s(r) \\
\sigma_P(r) & = \text{standard deviation of } \rho(s)\forall s \in N_s(r) \\
\epsilon(r) & = \text{average total delay for all flights in region } r \\
R_e(r) & = \text{recovered average total delay for region } r \\
R_s(r) & = \text{recovered average throughput for region } r
\end{align*}

I. INTRODUCTION

In recent Dynamic Airspace Configuration research, several algorithms for repartitioning the airspace into sectors have been developed. Each algorithm attempts to laterally partition a layer of airspace to minimize and/or balance controller workload, given a set of flight tracks. They approach the problem in different ways and produce radically different looking sectors.

Analyses have been conducted at a regional scale using different historical flight track data to assess the algorithm’s success. Because of their use of historical flight track data only, there have been no analyses done for expected future traffic levels.

This paper presents a comparison of sectorizations generated from different airspace partitioning algorithms that span the entire continental US airspace. The same flight track data was used to generate each sectorization. Thus, the results present a side-by-side comparison of the algorithms. In addition to current traffic levels, simulations were conducted and metrics were generated for 1.5 times today’s traffic levels.

This paper is organized as follows. Section II presents an overview of the sectorization algorithms. The experiment design is presented in section III. Section IV describes the metrics used to compare the sectorizations. Section V discusses results. Finally, concluding remarks are presented in section VI.
II. Sectorization Algorithms

This section presents an overview of the three algorithms used to produce a sectorization for this comparison.

A. Flight Clustering Algorithm

The Flight Clustering algorithm[1] groups flight track positions together to sectorize airspace. It is constrained to a maximum specified number of flight tracks to control the maximum workload that can be assigned to any sector. Sector boundaries are then formed around the groupings of flight route segments.

The clustering algorithm approach allows Dynamic Density[10] metrics to be implicitly manipulated. It attempts to partition airspace in such a manner that acceptable Dynamic Density levels are achieved, and the impact on user-preferred flight routes is minimized. Flight route segments are not only clustered according to distance from the cluster, but also according to heading, lateral and vertical speed, and future distance criterion. These criteria are selected and weighted to achieve control over the Dynamic Density of the resulting airspace partition. The number of sectors is determined by the chosen constraint, which is the maximum number of flight track instances per cluster.

B. Voronoi Genetic Algorithm

The Voronoi Genetic algorithm[2] uses the Voronoi Diagram to partition the airspace and a genetic algorithm to optimize the partitions.

The Voronoi Diagram decomposes a space into subdivisions around given generating points. All points within a region associated with a specific generating point, are closer to that generating point than any other generating point.

The genetic algorithm is a guided random search based on the principals of genetic inheritance and Darwinian evolution. Here, the genetic algorithm is used to find the set of Voronoi Diagram generating points that optimize given parameters. For this experiment, the genetic algorithm optimization goal is to maximize the gap between the capacity and peak aircraft count of each sector.

The number of resulting sectors is determined by the number of Voronoi Diagram generating points chosen to partition a region of airspace. Assuming a goal average capacity of 15 aircraft per sector, the desired number of sectors for a given region is estimated by dividing the peak traffic count for the region by 15.

C. Mixed Integer Programming Algorithm

The Mixed Integer Programming (MIP) algorithm[3] discretizes the airspace into hexagonal cells and clusters the cells according to workload and connectivity. The workload of a cell is the number of flight track counts within that cell. Connectivity from cell \( i \) to a neighboring cell \( j \) is the total number of flights that travel from cell \( i \) to the cell \( j \).

The algorithm works on the abstract quantity of workload flow. Flow enters each cell from at least one of its neighbors and exits into exactly one neighboring cell. The workload of each cell is added to the flow, which is finally absorbed by a sink cell. A sector consists of all cells whose flows converge to one sink. The algorithm attempts to minimize the flow between cells in different sectors. Thus, cell clusters will tend to be oriented along the dominant traffic flows. The number of resulting sectors is determined \( a \ priori \).

Improvements to the MIP algorithm were made in [4] by making connectivity between cells symmetric. Flows that are biased to go in the same direction allowed more convoluted flow paths to arise leading to rough sector edges containing unwieldy spurs. By redefining the connectivity between two neighboring cells \( i \) and \( j \) to include both flights traveling from cell \( i \) to cell \( j \) and from cell \( j \) to cell \( i \), flows become bidirectional and give the optimization more options to find a feasible solution with a lower objective function.

III. Experiment Design

This section discusses how the experiment was designed. The goal was to generate metrics for a side-by-side comparison of the current day sectorization and several algorithm generated sectorizations for the continental US airspace. Therefore, each sectorization is generated according to the same guidelines using the same flight track data.

Rather than using historical data to evaluate the sectorizations, flight track data is generated through simulation to remove the effects of current sector constraints and traffic management initiatives. Simulating the flight track data also allows the sectorizations to be evaluated for projected future traffic levels that do not exist in historical data. Simulated flight tracks can also be generated subject to constraints imposed by the new sectorization rather than the current sectorization.

Simulations were completed using the Airspace Concept Evaluation System (ACES) [5]. ACES models gate-to-gate flight operations on airport surfaces and in terminal and en-route airspaces. These include gate pushback and arrival, taxi, runway takeoff and landing, local approach and departure, climb, decent, transition, and cruise. Air traffic control and traffic flow management models control flights during these operations to ensure that airport and airspace capacity constraints are not violated.

Fig. 1 shows the process used to generate sectors and constraints and simulate traffic for each algorithm. Each of the stages in this process are discussed in the following subsections.

A. Unconstrained Simulation

The first stage of the experiment process is to simulate unconstrained flight tracks. The unconstrained simulation flies each flight according to its proposed flight plan contained in the flight schedule without any capacity constraints.

The flight schedule consisted of all the flight plans departing within 24 hours from a single high-traffic, good-weather day starting at 4/21/2005 8AM GMT. The last flight plan submitted before departure for each flight was used. The uncon-
stratified simulation produced unconstrained flight tracks for every minute of each flight. These unconstrained flight tracks are used to generate new sectorizations with new sector capacities.

B. Generating Sectorizations

The second stage of the experiment process generates new sectorizations using the unconstrained flight tracks produced by the first stage. Due to the infinite number of ways each sectorization algorithm could generate new airspace partitions, the algorithms are made to follow guidelines that make the resulting sectorizations more comparable.

The algorithms all produce lateral airspace divisions for a given set of flight track data. However, the current airspace sectorization is far more complicated than a single layer of sectors spanning the nation. National airspace is first divided into regions called centers and then subdivided into sectors. Sectors are also stratified into low, high, and super-high altitudes. The altitude ranges of sectors within the same stratum are not necessarily consistent.

Future airspace operational concepts do not necessarily preclude the airspace from being redesigned irrespective of center boundaries. However, in order to make the sectorization comparable at a more regional level, all sectorization algorithms were constrained to redesigning the airspace within the current center boundaries shown in Fig. 2.

C. Estimating Sector Capacities

After new sectorizations were generated, their sector capacities were estimated. Several methods for estimating sector capacity have been proposed in the literature [6, 7]. However, the most straightforward capacity estimation method, and the one used for this experiment, is the method used by the FAA to determine Monitor Alert Parameter (MAP) values [8]. The MAP formula is a function of the average flight time of aircraft in the sector including the highest flight level was classified as super-high altitude. Some sectors covered the entire altitude range and were classified as both high and super-high altitude sectors.

D. Constrained Simulation

The final stage of the experiment process is to simulate constrained flight tracks for each sectorization. Constrains are applied to ACES simulations as airport and sector capacities. Air traffic control and traffic flow management models control flights to ensure that capacity constraints are not violated by delaying flights along their filed flight plan. For the purposes of this experiment, only capacity constraints on sectors within redesigned airspace are applied. Airport capacities and capacities for airspace outside the scope of this experiment are left unconstrained. The sector capacity constrained simulations result in a unique set of constrained flight track data for each sectorization simulated. These flight tracks and simulated delays are used to generate metrics with which to compare the sectorizations.

Figure 1: Experiment process.

Figure 2: Continental United States airspace centers.
IV. Metrics

This section discusses the metrics used to compare the sectorizations. All of the metrics can be applied to a group of sectors. The basic metrics are number of sectors, demand, throughput, capacity, complexity, and delay. The demand/capacity and throughput/capacity ratios are also of interest since ideally capacity should be placed where demanded.

A. Number of Sectors

There were no restrictions made on the number of sectors, \( n_s(r) \), generated for a given region \( r \) in the new sectorizations. Everything else being equal, a lower \( n_s \) is desirable to make more efficient use of controller resources. There is an inherent tradeoff between \( n_s \) and capacity. As \( n_s(r) \) increases, the sum capacity in region \( r \) would be expected to increase as well. However, increasing \( n_s \) reduces the average sector size which may make average \( \delta(s) \) and \( c(s) \) for each sector lower. At some point, increasing the number of sectors will reduce average sector capacity so much that the sum capacity of the sectors does not increase.

B. Demand and Throughput

Demand and throughput metrics are both based on average instantaneous sector flight counts within quarter-hour intervals. Demand and throughput are computed from unconstrained and constrained flight track data, respectively. Demand and throughput metrics are computed for a region of airspace, \( r \), by averaging the average instantaneous flight counts per quarter-hour over the mid 8 hours of the day when traffic is highest. A region can comprise any group of sectors, such as a single sector, center, altitude layer, or NAS-wide. Let \( m(r, k) \) be the average flight count in \( r \) for quarter-hour, \( k \). The average sector flight count in \( r \) is given by

\[
m(r) = \frac{1}{32} \sum_{k=0}^{k_0+32} m(r, k).
\]

Let the subscripts \( d \) and \( t \) denote demand and throughput. Let \( M_d(r) \) and \( M_t(r) \) be the average \( m_d(s) \) and \( m_t(s) \) for all \( s \) in \( r \). Let \( \sigma_{m_d}(r) \) and \( \sigma_{m_t}(r) \) be the standard deviations for \( m_d(s) \) and \( m_t(s) \) for all \( s \) in \( r \).

Throughput is desired to be as close as possible to demand. The ratio of throughput to demand should be as close to 1 as possible. Recovered throughput is a metric designed to evaluate how much a new sectorization increases the throughput/demand ratio from the Current Day sectorization. Let \( R_t(r) \) be the recovered average throughput in region \( r \) given by

\[
R_t(r) = 1 - \frac{(1 - m_t(r))}{(1 - m_d(r))}
\]

where the subscript 0 denotes the Current Day sectorization.

Assuming that maximum quarter-hourly flight count is a major component of controller workload, it is desirable for \( \sigma_{m_d}(r) \) and \( \sigma_{m_t}(r) \) to be low in order to balance the workload.

C. Capacity

The capacity of each sector is defined according to (1) and (2). The \( c(s) \)'s are used as constant constraints for the sector capacity constrained ACES simulations. Capacity sum, average, and standard deviation for regions or sectors are computed as well. Let \( \Sigma_c(r) \), \( C(r) \), and \( \sigma_c(r) \) be the sum, average, and standard deviation of \( c(s) \) for all \( s \) in \( r \).

Higher \( \Sigma_c(r) \) and \( C(r) \) are desirable to enable increased throughput. Intuitively, a lower \( \sigma_c(r) \) should be desirable. However, capacities themselves are designed to keep workload within acceptable limits. Therefore, there may be a tradeoff between balancing workload metrics and capacity. It is more desirable to balance workload metrics than capacity.

D. Capacity Ratios

A set of metrics that is perhaps more relevant than demand and throughput, or capacity, are the ratios of demand and throughput to capacity. Ideally, capacity should be placed where the demand is in order to maximize overall throughput. The demand/capacity ratio is a measure of how well the sectors accommodate the traffic and the throughput ratio is a measure of workload levels.

Let \( \rho_d(s) \) and \( \rho_t(s) \) be the average maximum quarter-hourly demand/capacity and throughput/capacity ratios for \( s \) over the mid 8 hours of the day given by

\[
\rho_d(s) = \frac{m_d(s)}{c(s)} \quad \text{and} \quad \rho_t(s) = \frac{m_t(s)}{c(s)}
\]

Average and standard deviations of capacity ratios for regions of sectors are also computed. Let \( P_d(r) \) and \( P_t(r) \) be the average \( \rho_d(s) \) and \( \rho_t(s) \) for all \( s \) in \( r \). Let \( \sigma_{\rho_d}(r) \) and \( \sigma_{\rho_t}(r) \) be the standard deviations of \( \rho_d(s) \) and \( \rho_t(s) \) for all \( s \) in \( r \).

E. Complexity

There has been a lot of research done to develop complexity metrics that measure controller workload [9, 10, 11, 12, 13, 14]. These efforts concentrate on identifying and validating up to 52 quantifiable complexity variables based on the factors that contribute to workload. References [12], [13], and [14] present the most simplified subset of complexity metrics referred to as Simplified Dynamic Density (SDD) metrics. SDD metrics comprise of just seven components that can all be derived from 1 minute resolution historical track data. Because simulated track data from ACES also has a 1 minute resolution, SDD metrics were chosen to represent complexity in this experiment.

The seven components \( x_1 \) through \( x_7 \) of SDD metrics are occupancy counts, proximities, altitude transitions, sector boundary crossings, aircraft per sector volume, heading variance, and cruise speed variance. These are calculated per quarter-hour and are combined with a weighted sum. Component weights were taken from [14].

1) Occupancy Count \((x_1)\) and Aircraft per Sector Volume \((x_5)\): The occupancy count component for SDD metrics is the
average instantaneous sector flight count for the given quarter-hour. Therefore, \( x_1(s, k) = m(r, k) \) where \( N_s(r) = \{s\} \). Aircraft per sector volume is simply \( x_1 \) divided by the sector volume in cubic kilometers.

2) **Proximity** \((x_2)\): Proximity events of different severity levels are calculated for all aircraft pairs within 10 nmi at the same time. Proximity severity levels in [12], [13], and [14] were designed to account for historical data location uncertainty, time-stamps being recorded at different times, and 1 minute granularity. The proximity severity computations for simulation data have been simplified with respect to time, because the 1 minute time-stamps for each flight are produced for the exact same time. Table 1 shows the criteria for calculating proximity between a pair of aircraft within the same timestamp.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>&lt;1000 ft</td>
<td>&lt;5 nmi</td>
</tr>
<tr>
<td>2</td>
<td>&lt;1000 ft</td>
<td>5 to 7.5 nmi</td>
</tr>
<tr>
<td>3</td>
<td>&lt;1000 ft</td>
<td>7.5 to 10 nmi</td>
</tr>
<tr>
<td>4</td>
<td>≥1000 ft</td>
<td>&lt;5 nmi</td>
</tr>
</tbody>
</table>

The combined proximity component, \( x_2(s, k) \), is defined as follows.

\[
x_2(s, k) = \frac{1}{4} (4p_1 + 2p_2 + p_3 + p_4)
\]

where \( p_1, p_2, p_3, \) and \( p_4 \) indicate the number of proximities counted in \( s \) during \( k \) for each corresponding severity level.

3) **Altitude Transitions** \((x_3)\): Altitude transitions are counted for tracks that climb or descend more than 500 feet within a minute. \( x_3(s, k) \) is the sum of all tracks within \( s \) during \( k \) that are not within 500 feet of their last track.

4) **Sector Boundary Crossings** \((x_4)\): Every time a flight crosses a sector boundary, a boundary crossing is counted for both the outbound and inbound sector. \( x_4(s, k) \) is the combined number of flights that enter or exit \( s \) within \( k \).

5) **Heading Variance** \((x_5)\) and **Speed Variance** \((x_7)\): Heading and Speed Variances, \( x_5(s, k) \) and \( x_7(s, k) \), are calculated for the set tracks in sector \( s \) within quarter-hour \( k \). Variances are calculated for heading in degrees and for groundspeed in knots.

6) **Combined SDD Components**: The seven SDD components described above were combined in a weighted sum as follows.

\[
x = [x_1, x_2, x_3, x_4, x_5, x_6, x_7]
\]

\[
w = [2.2, -4, -3, 5, 30000, .0005, .0005]
\]

\[
\chi = x \cdot w
\]

The average complexity for a given sector, \( \chi(s) \), is calculated similar to \( m(s) \) as follows.

\[
\chi(s) = \frac{1}{32} \sum_{k=k_0}^{k_0+32} \chi(s, k)
\]

where \( k_0 \) is the first quarter hour in the 8 hour for which \( s \) has the highest average traffic counts.

Average and standard deviations of complexity for regions of sectors are also computed. Let \( X(r) \) be the average \( \chi(s) \) for all \( s \) in \( r \). Let \( \sigma_X(r) \) be the standard deviation of \( \chi(s) \) for all \( s \) in \( r \).

F. **Delay**

ACES collects the time for various events of each simulated flight. The difference between event times for unconstrained and constrained simulations is delay. Traffic flow management models may apply delay multiple times to the same flight event in order to meet multiple constraints. It is very difficult to attribute a flight event delay to a single constraint such as a particular sector’s capacity. Therefore, sector delays are computed as average total delay for all flights flying through the sector. Delays by region are computed similarly.

The total delay for a single flight is the difference between its constrained and unconstrained gate arrival time. Let \( \epsilon(r) \) be the average total delay for all flights in region \( r \) given by

\[
\epsilon(r) = \frac{1}{n_f(r)} \sum_{i} (t_c(i) - t_u(i)) \forall i \in N_f(r)
\]

where \( t_c(i) \) and \( t_u(i) \) are the constrained and unconstrained gate arrival times for flight \( i \).

V. **Simulation Results**

This section presents and discusses the metric comparison results between sectorizations. Each of the three algorithms used flight tracks generated from an unconstrained simulation of a current day traffic schedule (1X) to create unique sectorizations subject to the guidelines discussed in section IV. Fig. 3 shows the resulting sectorizations for high altitude Fort Worth Center. The same set of unconstrained 1X flight tracks were used to estimate sector capacities for each of these sectorizations and the modified current day sectorization. These sector capacities were then used to simulate constrained 1X flight tracks through each sectorization.

In order to test the sectorizations with more futuristic flight traffic levels, a flight demand generation tool called AvDemand[15] was used to homogeneously grow the 1X flight traffic schedule to 1.5X. Another unconstrained simulation was used to generate unconstrained 1.5X flight tracks for this traffic schedule. No new sectorizations or capacity estimations were generated using the unconstrained 1.5X flight tracks. Instead, the 1X generated sectorizations and capacity estimations were used to simulate constrained 1.5X flight tracks.

The resulting 16 sets of flight track data were used to calculate the metrics defined in section IV. They consisted of 4
No design restriction for number of sectors for each region was placed on the sectorization algorithms. Therefore, some sectorizations resulted in very different numbers of sectors from the current day sectorization.

Fig. 4 shows each sectorization’s total number of sectors for the National Airspace System (NAS), \( n_s(NAS) \), at each altitude stratum. The NAS includes all 20 centers. For the algorithm generated sectorizations, \( n_s(NASall) \) is the sum of \( n_s(NAShigh) \) and \( n_s(NASsuper) \). For the current day sectorization, there were many sectors classified as both high and super-high altitude due to the inconsistent altitude structure. The MIP algorithm was given a goal number of sectors per center to be similar to the average number of sectors at any one altitude for that center. The MIP number of sectors is the same for both high and super-high altitude strata. The number of sectors for the Flight Clustering and Voronoi Genetic sectorizations were influenced by other algorithm parameters as described in sections A. and B.. Both resulted in fewer super-high sectors than high sectors. All the algorithm generated sectorizations have more total sectors than the current day sectorization. The Flight Clustering produced more than twice as many total sectors. If number-of-flight-tracks-per-cluster parameter were increased, the resulting number of sectors would be decreased. A sensitivity analysis has not yet been completed to determine the optimal parameter setting.

The relative numbers of sectors per center for each sectorization follow similar patterns as the NAS. There are several exceptions for high altitude centers where the Voronoi Genetic algorithm produces close to or more than the number of sectors as the Flight Clustering Algorithm. These centers are ZDC, ZID, ZNY, and ZOB, all notoriously busy centers.

B. Capacity

Like number of sectors, capacity is something that is unique to each sectorization and not the simulation that was run. Although 1X unconstrained track data was used to estimate the capacities of each sectorization, the same set of capacities was used in both the 1X and 1.5X constrained simulations.

In general, average capacities, \( C(r) \), are lower for the algorithm generated sectorizations than the current day sectorization. However, because algorithm sectorization resulted in more sectors per region, the sum capacities, \( \Sigma_c(NASall) \), are higher than for the Current Day sectorization, especially in the case of the Flight Clustering sectorization. Fig. 5 shows \( \Sigma_c(NASall) \) for each sectorization. The Flight Clustering and Voronoi Genetic algorithms show general increases in \( \sigma_c(r) \) over Current Day, whereas the MIP algorithm shows general decreases in \( \sigma_c(r) \) over Current Day. Consistency in capacity between sec-
tors is not as important as constancy in capacity ratios or complexity.

C. Demand and Throughput

The Demand and Throughput metrics are important metrics for assessing how well the sectorization accommodates traffic demand. These metrics compare track data from unconstrained and constrained simulations within the same 1X or 1.5X traffic level. Fig. 6 and 7 show the NAS-wide average Throughput/Demand percentages for each altitude stratum for 1X and 1.5X traffic respectively. Higher percentages indicate that the traffic is less constrained and throughput is being allowed to reach demand levels.

As seen in Fig. 6 for 1X traffic, the algorithm generated sectorization increases average NAS throughput/demand percentages, but there is very little room to improve because the Current Day sectorization 1X throughput/demand percentage is already above 100%. In some cases, the percentage is slightly above 100%. This is possible when minor traffic schedule shifting ripples through the system to create a situation where more aircraft occupy an airspace at the same time than was originally demanded.

The 1.5X traffic does a better job of straining the system to more effectively evaluate the algorithm generated sectorization improvements over the Current Day sectorization. Fig. 7 clearly shows that all three algorithm generated sectorizations produce higher throughput/demand percentages than the Current Day sectorization. Flight Clustering and MIP sectorizations produce larger improvements at the high altitude stratum and the Voronoi Genetic algorithm produces the largest improvement at the super-high altitude stratum.

The standard deviations of demand/capacity and throughput/capacity ratios between sectors in a region are more important than the ratios themselves. The standard deviations, especially for $\rho_d$ as opposed to $\rho_t$, indicate weather capacity is distributed appropriately to accommodate demand. A smaller $\sigma_{\rho_d}(r)$ means that the sectorization placed more capacity where demand needed it, and less capacity where it wasn’t needed. Because, each sectorization was design for a single center at a single stratum at a time, it makes the most sense to evaluate $\sigma_{\rho_d}(r)$ at the center regional level.

Fig. 9 shows the average $\sigma_{\rho_d}(r)$ across all centers in each altitude stratum for each sectorization. All three algorithm generated sectorizations show a reduced average $\sigma_{\rho_d}(r)$ over the Current Day sectorization. The MIP algorithm does an especially good job of distributing capacity appropriately to match demand. It has half the average $\sigma_{\rho_d}(r)$ for high altitude and a quarter the average $\sigma_{\rho_d}(r)$ for super-high altitude.

The 1X average $\sigma_{\rho_t}(r)$s are almost identical to the 1X average $\sigma_{\rho_d}(r)$s. This is not surprising, considering how little throughput deviated from demand for all of the 1X simulations. Both the average $\sigma_{\rho_d}(r)$s and $\sigma_{\rho_t}(r)$s for 1.5X show the exact same trend between sectorizations and altitude stratum as in Fig. 9. The the average $\sigma_{\rho_d}(r)$s increased by 70% and the average $\sigma_{\rho_t}(r)$s increased by 46% between 1X and 1.5X traffic.

As in Fig. 5 for 1X traffic, the algorithm generated sectorization increases average NAS throughput/demand percentages, but there is very little room to improve because the Current Day sectorization 1X throughput/demand percentage is already above 99%. In some cases, the percentage is slightly above 100%. This is possible when minor traffic schedule shifting ripples through the system to create a situation where more aircraft occupy an airspace at the same time than was originally demanded.

The 1.5X traffic does a better job of straining the system to more effectively evaluate the algorithm generated sectorization improvements over the Current Day sectorization.
Figure 8: Percent recovered throughput by region, $R_t(r)$, for 1.5X traffic.

Figure 9: Average 1X standard deviation of Demand/Capacity for all center regions within each altitude stratum.

Figure 10: Average Complexity in the National Airspace System, $X(NAS)$, within each altitude stratum.

E. Complexity

Complexity is computed to serve as a more realistic measure of controller workload than just occupancy count. All of the sectorization algorithms utilize metrics that relate to complexity components other than occupancy count. Although occupancy count is still a driving factor in the design.

Fig. 10 shows the average NAS-wide complexity within each altitude stratum for 1X and 1.5X, and unconstrained and constrained simulations. Because occupancy count is a heavily weighted component of complexity, 1.5x has larger $X(NAS)$s than 1X, and $X_t(NAS)$ is lower than $X_d(NAS)$ for every sectorization because occupancy counts are controlled to be below capacity constraints. $X_t(NAS)$ is only slightly lower than $X_d(NAS)$ for 1X because the throughput did not deviate from demand much at 1X.

The Flight Clustering sectorization reduces complexity the most due to its high number of sectors and consequently low number of flights per sector. Other complexity components increase more for Flight Clustering due to the reduced average size of more sectors but reduced occupancy count makes up for increases in other components. The Voronoi Genetic sectorization has the most similar $X(NAS)$ between altitude strata because number of sectors designed for each center and altitude stratum were based on the occupancy count. MIP is the only algorithm generated sectorization with higher $X(NAS)$ than Current Day for the high-altitude stratum. MIP also had the least number of sectors at this stratum.

Fig. 11 shows the percentage each algorithm generated sectorization reduces $X_d(r)$ from the Current Day sectorization at 1X broken out by center regions. The Flight Clustering sectorization is the only one that reduces $X_d(r)$ for every center. Many of the centers for which the Voronoi Genetic or MIP sectorization increase complexity are the same centers for which throughput decreased in Fig. 8. These were the centers with the lowest number of sectors.

Fig. 12 shows the average $\sigma_\chi(r)$ for all centers within each altitude stratum for 1X and 1.5X, and unconstrained and constrained simulations. Average $\sigma_\chi(r)$ follows the same trend as $X(NAS)$ between 1X and 1.5X and between unconstrained and constrained simulations. Super-High altitude average $\sigma_\chi(r)$s are consistently lower than for High altitude.

Average $\sigma_\chi(r)$s are significantly lower for the algorithm generated sectorizations than the Current Day sectorization. The Flight Clustering and Voronoi Genetic sectorizations have very similar average $\sigma_\chi(r)$. The average $\sigma_\chi(r)$ for MIP is much lower in each case. This means that the MIP algorithm did the best job of distributing workload, as measured by SDD complexity, evenly between sectors.
Figure 11: Percent that $X_d(r)$ is reduced from the Current Day to each algorithm generated sectorization by region for 1X traffic.

Figure 12: Average standard deviation of Complexity for all centers, $\sigma_\chi(r)$, within each altitude stratum.

F. Delay

Delay is the ultimate measure of cost to the airspace system customers. Table II. shows the average total delay simulated for all flights, $\epsilon_{(NAS)}$, at 1X and 1.5X traffic for each sectorization. All of the algorithm generated sectorizations significantly reduce delay of current day at 1X traffic and by similar amounts. While delays are still reduced at 1.5X, the amount varies more by sectorization.

The Voronoi Genetic algorithm shows the most robust partitioning with respect to delay for increasing traffic demand. Because delay is incurred by modifying traffic to meet capacity constraints, the method for estimating capacity directly affects delay. Recall that the capacity estimation for this experiment is a linear increase with the average time a flight takes to fly through the sector. The Voronoi Genetic algorithm explicitly incorporated this capacity estimation into it’s optimization function with a goal of increasing the residual capacity once the maximum occupancy count is subtracted.

<table>
<thead>
<tr>
<th>Sectorization</th>
<th>$\epsilon_{(NAS)}$ for 1X</th>
<th>$\epsilon_{(NAS)}$ for 1.5X</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current Day</td>
<td>3.99 min</td>
<td>45.02 min</td>
</tr>
<tr>
<td>Flight Clustering</td>
<td>0.39 min</td>
<td>18.68 min</td>
</tr>
<tr>
<td>Voronoi Genetic</td>
<td>0.47 min</td>
<td>16.50 min</td>
</tr>
<tr>
<td>MIP</td>
<td>0.55 min</td>
<td>30.11 min</td>
</tr>
</tbody>
</table>

Fig. 13 and 14 show the percent recovered delay, $R_\epsilon(r)$, for each center region. There were more instances of negative $R_\epsilon(r)$, indicating increased delay over the Current Day sectorization, for 1X than for 1.5X traffic. This is because at many centers, the 1X traffic did not stress the system enough to incur significant delay, especially Chicago Center (ZOB). ZOB is a very busy center for which the airspace was more recently designed than other centers to accommodate it’s high demand. Only the MIP sectorization shows negative $R_\epsilon(r)$ at 1.5X for the same three centers with significant negative $R_\epsilon(r)$ at 1X.

VI. Conclusions

This experiment is the first US nation-wide, side-by-side comparison of different algorithm generated sectorizations. Simulating flight traffic through each of these sectorization for both 1X and 1.5X traffic schedules improved the comparison by stressing the system enough to evaluate the algorithms strengths and weaknesses.

Each algorithm shows it’s strengths and weaknesses through the different metrics. The Flight Clustering sectorization significantly increased throughput, while reducing complexity and delay, but only at the cost of doubling the number of sectors that exist in the current day system. The Voronoi Genetic sectorization had a more comparable number of sectors to todays system, while increasing throughput and reducing delay similar to the Flight Clustering algorithms. The Voronoi Genetic sectorization had a more modest reduced complexity over the Current Day sectorization with respect to Flight Clustering. The MIP sectorization also had comparable numbers of sectors to Current Day with similar increases in throughput to Flight Clustering and Voronoi Genetic. However, the MIP algorithm’s greatest strength was in balancing capacity and complexity. The MIP algorithm’s weakness was in reducing complexity and delay at high traffic levels.

Two major realizations from this experiment were that the number of sectors designed for each region and the altitudes at which airspace is stratified are not trivial airspace design factors. These design factors need to be better incorporated into the algorithms. The number of sectors has been well integrated in to the Flight Clustering design but sensitivity analyses still need to be performed to evaluate how the rest of the metrics are effected as the number-of-flight-tracks-clustered parameter is increased and the number of sectors decreases.

In order to test the effect of intelligent altitude stratification, unique altitudes were chosen to split each center into high and super-high stratum that would split traffic equally between the two. A Voronoi Genetic sectorization of the resulting regions produced a 95% recovered delay at 1X vs. the 88% recovered
delay from the Voronoi Genetic sectorizations stratified at flight level 350 across all centers. This technique does not incorporate altitude stratification into the algorithm but it does show the advantage of using some intelligence in stratifying the airspace over using an arbitrary altitude.

ACKNOWLEDGMENT

Many thanks to Charlene Cayabyab for her above-and-beyond excellent support on this research in the ACES lab at NASA Ames Research Center.

REFERENCES


AUTHOR BIOGRAPHY

Shannon J. Zelinski received a M.S. degree in Electrical Engineering Robotics and Controls from the University of California at Berkeley in 2003. She then
joined the Aviation Systems Division at NASA Ames Research Center, where she gained expertise in air traffic management, simulation validation, and future demand generation, working with the Airspace Concept Evaluation System. Ms. Zelinski currently leads Dynamic Airspace Configuration research at NASA as an Associate Principal Investigator for the NextGen-Airspace project.