

Three Dimensional Sector Design with Optimal Number of Sectors

Min Xue*

University of California at Santa Cruz, Moffett Field, CA 94035

In the national airspace system, sectors get overloaded due to high traffic demand and inefficient airspace designs. Overloads can be eliminated in some cases by redesigning sector boundaries. This paper extends the Voronoi-based sector design method by automatically selecting the number of sectors, allowing three-dimensional partitions, and enforcing traffic pattern conformance. The method was used to design sectors at Fort-Worth and Indianapolis centers for current traffic scenarios. Results show that new designs can eliminate overloaded sectors, although not in all cases, reduce the number of necessary sectors, and conform to major traffic patterns. Overall, the new methodology produces enhanced and efficient sector designs.

Nomenclature

M	average sector dwell time, <i>minute</i>
P_I	intersection proximity, 1
P_D	dominant flow proximity, 1
N_s	aircraft count of short dwell, 1
N_a	average aircraft count in a sector, 1
V_p	variance of sector peak aircraft count, %
T	dwell time of a flight, <i>minute</i>

I. Introduction

In the national airspace system, high traffic demand and inefficient airspace designs cause overloaded sectors. Because some overloads can be eliminated by redesigning sector boundaries, the congested sectors impose unnecessary delays and traffic rerouting. These problems become worse when traffic patterns and demands fluctuate. An automatic algorithm is needed to design airspace boundaries such that overloads can be reduced. Then, to accommodate the fluctuating traffic patterns and demand, the automatic algorithm can be applied as needed to address the current airspace constraints.

Considerable research in automatic sector design methods has been conducted.¹⁻⁹ However, these methods are still facing some challenges. For instance, the GA-based algorithms¹⁻³ struggle with handling a large volume of real traffic. The cell-based algorithms⁴⁻⁷ produce jagged boundaries or unacceptable sector shapes. The sector boundaries generated by the computational geometry algorithm⁸ so far cannot conform to major flows. Also, as shown by Zelinski,¹⁰ the flight clustering based algorithm⁹ produces too many sectors. In previous work,¹¹ an approach that combines Voronoi diagrams, a genetic algorithm, and an iterative deepening algorithm was proposed. The method guarantees that the sector boundaries are convex and accepts nonlinear objective functions other than aircraft count. Designs produced by the method also conform to the dominant flows. However, the number of sectors was predefined and fixed, and only two-dimensional partitions were allowed. These restrictions constrained the Voronoi-based method's ability to achieve efficient sector designs.

The current work extends the Voronoi-based method by automatically selecting the number of sectors and allowing three dimensional partitions. Automatic selection of the number of sectors enhances the method's

*Research Scientist, University Affiliated Research Center. Mail Stop 210-8. AIAA member

ability to create sector boundaries that conform with traffic patterns. The new method includes three costs: primary, secondary, and design cost. The primary and secondary costs were empirically designed for traffic pattern conformance. The design cost was set up to judge if a sector is overloaded or not. The method was applied to sector designs at Fort-Worth and Indianapolis centers with different traffic scenarios. The new designs have self-defined numbers of sectors and conform to traffic patterns. Compared with the previous method, the designs generated by the enhanced method can eliminate or reduce overloads without changing the traffic.

In this paper, a preliminary study of Voronoi and the iterative deepening algorithms is described in Section II. Section III presents the new design strategy. Section IV describes cost functions. Section V presents the results and comparisons. Conclusions are discussed in Section VI.

II. Preliminary Study

In order to enhance the previous Voronoi-based method, studies must be conducted to examine the strength and weakness of included algorithms. In the following sections, the limitations of Voronoi diagrams are investigated and the way the iterative deepening algorithm compensates for these limitations is studied. Finally, the benefits of vertical partitions are discussed.

A. Voronoi Diagrams and Iterative Deepening Algorithm

Given a set of generating points, Voronoi diagrams can partition space into proximal regions such that any point in the region is associated with the closest generating point.¹² One of the properties of Voronoi cells is that they are always convex.¹² There are a great variety of Voronoi diagrams¹² depending on the way of defining generators (points, lines, areas, or 3D objects), the way of defining distance (Euclidian distance, Hausdorff distance, etc), and the way of weighting distance to generators. The Voronoi adopted in this work is the ordinary Voronoi diagrams that apply point generators, Euclidian distance, and uniform weights on distances. The property of partitioning space based on generating points reduces the number of parameters for optimization. An optimizer only needs to adjust the locations of the generating points to find a good sector design.

However, using the ordinary Voronoi diagrams alone might over-restrict the design space, especially when the number of partitions becomes high. In experiments, when a center was divided into a large number of sectors in one step, the resulting performance cost was not improved much, even after many generations of GA. It was found that, as the number of partitions increases, the Voronoi design space could be more limited. This implies that the number of partitions should not be too high when applying Voronoi diagrams directly at a single step.

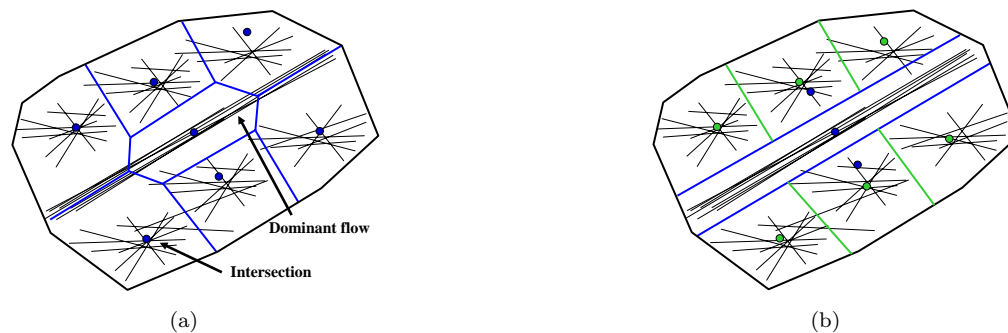


Figure 1. A sample of airspace partition. (a) Voronoi diagrams only. (b) Voronoi diagrams and the iterative deepening algorithm

In order to cope with this restriction, the iterative deepening algorithm is used to break up a partition with a large number of sectors into multiple steps. To illustrate the principle, a theoretical case where the traffic is not real will be used. Suppose there exists a dominant traffic flow surrounded by several intersection areas, as shown in Fig. 1(a). Assuming seven partitions is the best choice for this structure, if Voronoi diagrams are applied directly with seven partitions, the dominant flow cannot be contained in

a single sector. In fact, as shown by the blue lines in Fig. 1(a) that were generated by Voronoi directly, Voronoi segregates the flow into several parts. However, if the iterative deepening algorithm is exploited, this airspace can first be separated into three parts, as shown by blue lines in Fig. 1(b). Then two of the parts can be further partitioned in accordance with the traffic pattern. The resulting partition, shown by the blue and green lines in Fig. 1(b), is more desirable because it does not break the main flow. Thus, the iterative deepening algorithm can compensate for the limitations of Voronoi diagrams.

In this work, the number of divisions at a single step is selected to be less than six. Although, five or six divisions may still be high, however, they offer flexibility. For example, Voronoi diagrams can easily form a good partition for a typical hub-like traffic pattern.

B. Vertical Partitions

If only horizontal partitions are allowed, sectorization is constrained. Fig. 2(a) and 2(b) show an example of how vertical segregation can help create a good partition. Suppose, as in the scenario shown in the figure, there are two groups of aircraft flying at two different levels. (For simplicity, instead of symbols of aircraft, only time-elapsd trajectories are shown in this figure.) Assuming the capacity is approximated based on average dwell time and the demand is based on peak aircraft count (e.g. using the criterion of “ $PeakAircraftCount > 5/3 \times AverageDwellTime$ ”^a), this airspace is overloaded because the capacity estimation is 25 and peak count is 30. Further division may be needed. If the airspace is segregated with the green dash lines in the figures, assuming aircraft are evenly distributed at the peak time, both sector average dwell time and peak aircraft count will be halved to 12.5 and 15, respectively. Thus the resulting two sectors are still overloaded. Whereas if the airspace is vertically stratified using the blue dash line, only the peak count is halved and sector dwell is kept the same. The resulting two sectors would not be overloaded anymore. Therefore, in this case a vertical partition is desired, both for maintaining good controllers’ situation awareness and meeting the design requirement. These costs are discussed further in subsection E.

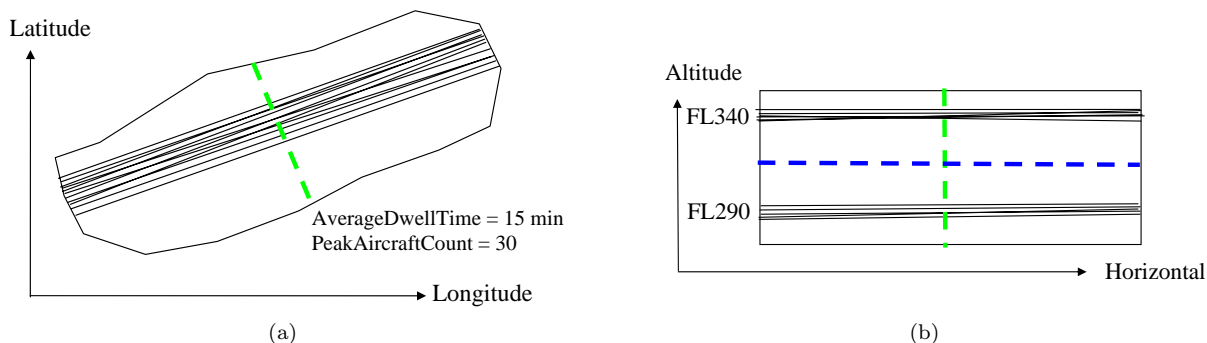


Figure 2. A sample of vertical partition (a) top view (b) side view

It is preferable not to use too many vertical partitions because entering a sector vertically causes more workload than horizontal boundary crossings. In this work, a threshold (e.g. 1,500ft) was set up to limit the smallest vertical range for a single sector. Additionally, in terms of when to stratify vertically and when to stop, three types of rules are tested in this work: conservative, moderate, and aggressive. In the “conservative” regime, stratification is applied only when, right after the application, the design requirement can be satisfied. In the “moderate” and “aggressive” regimes, even though the immediate consequences of vertical partitions do not satisfy the design cost, stratifications will be performed if horizontal partitions cannot produce a design that is closer to the design requirement. The difference between “moderate” and “aggressive” is that, in “moderate” regime, although the requirements cannot be satisfied after stratification, the resulting divisions will not be further divided, whereas in “aggressive” cases, the resulting divisions are still open for further partitioning.

^aThe five thirds is calculated from the table in FAA Order 7210.3 using the curve-fitting method. This formula has been used as a simple capacity estimation in many airspace redesign studies

III. Approach

Based on the above discussions, a new tri-cost strategy that uses three costs - a primary cost, a secondary cost, and a design cost - is established. Optimization with the primary cost is repeated for different numbers of sectors to create a set of eligible designs. The secondary cost is used to judge the preference among designs. The design cost is utilized as a criterion to determine the ultimate number of sectors. When the design cost reaches a defined threshold, the partitioning process stops.

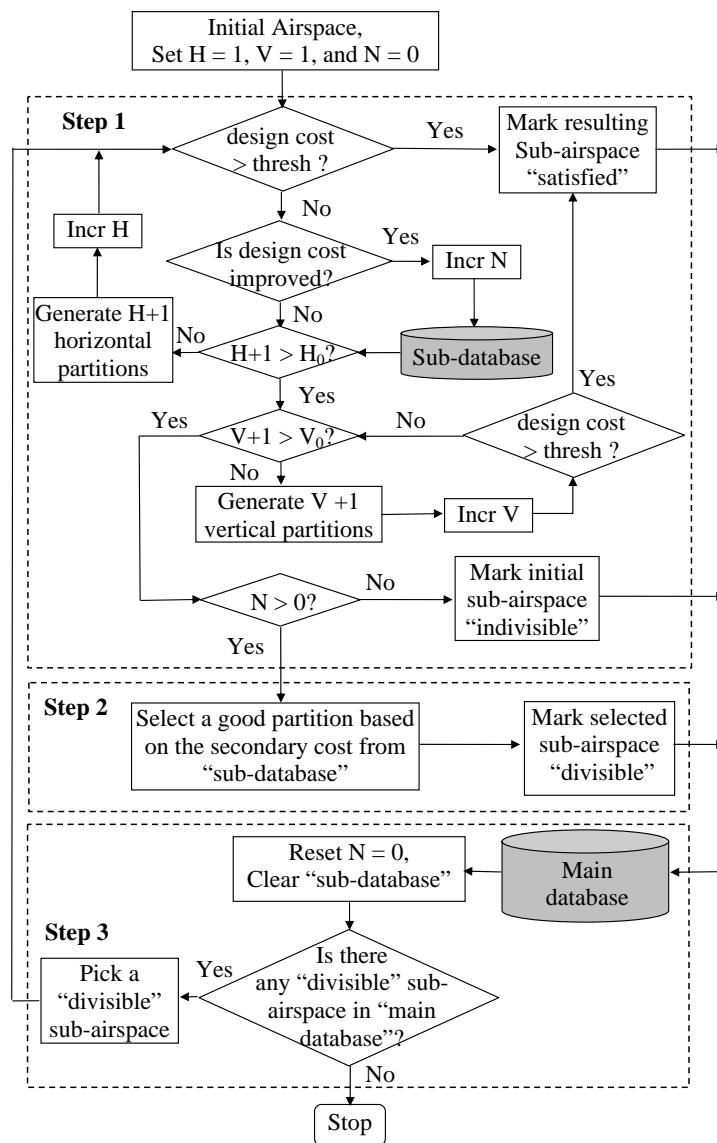


Figure 3. Procedure of the tri-cost strategy

A flow chart is shown in Fig. 3 to illustrate the detailed procedure. In this chart, H represents the number of horizontal partitions, and V denotes the number of vertical partitions. H_0 is the limit of the number of horizontal partitions, and V_0 is the limit of the number of vertical partitions. In this work, H_0 and V_0 are set to be six and three, respectively. N is a temporary count of sectors stored in the sub-database. A "divisible" airspace can gain capacity by further division. Whereas an "indivisible" airspace implies that dividing the airspace will reduce capacity, thus it is good not to be divided. A "satisfied" airspace is an airspace that meets design cost or requirement. The "conservative" regime is shown in the flow chart. The following are detailed descriptions:

Step 1 The algorithm first checks if partitioning is needed for a given airspace according to the design requirement. If no, the airspace is marked with “satisfied” and sent to the main database, and the process goes to Step 3. If yes, optimal sectorization is performed based on the primary cost function. Different numbers of partitions including vertical stratifications are generated. In this work, the exploration range is set at between two and six partitions for horizontal segregation and between two and three for vertical division. If any resulting partitions meet the design requirement, they are marked with “satisfied”. In this step, if any partition improves the design cost, it will be stored in “sub-database” for Step 2. If no partition that provides better design cost than upper-iteration sub-airspace, the initial sub-airspace is marked “indivisible”.

Step 2 A secondary cost is utilized to judge the partitions in the “sub-database” to see which partition among them may lead to a good design. The winner’s partitions are marked with “divisible” and sent to the main database. Meanwhile, the “sub-database” gets cleared, and N is reset to zero.

Step 3 All sub-airspaces stored in the main database are checked. If there is any “divisible” sub-airspace or sub-sectors, the algorithm will pick one and go to Step one. If all sub-airspace are either “satisfied” or “indivisible”, they will be output as final sectors and the entire process will be terminated.

IV. Objectives

In this work, efforts were dedicated to developing new costs on the basis of previous work. The primary and secondary costs were identified based on numerous experiments. Before describing these costs, the next few sections present key metrics used in costs, including the sector average dwell time, the intersection proximity, the dominant flow proximity, the aircraft count of short dwell flights, and the variance of peak aircraft count. Straightforward metrics such as the sector boundary crossings are skipped. Later sections describe how the costs are constructed.

A. Average Sector Dwell Time (M)

As discussed in previous work, the average sector dwell time M_j is defined as Eqn. 1, where T_{ij} is the dwell time for flight i at Sector j , and n_j is the total number of aircraft in Sector j .

$$M_j = \frac{\sum_i T_{ij}}{n_j} \quad (1)$$

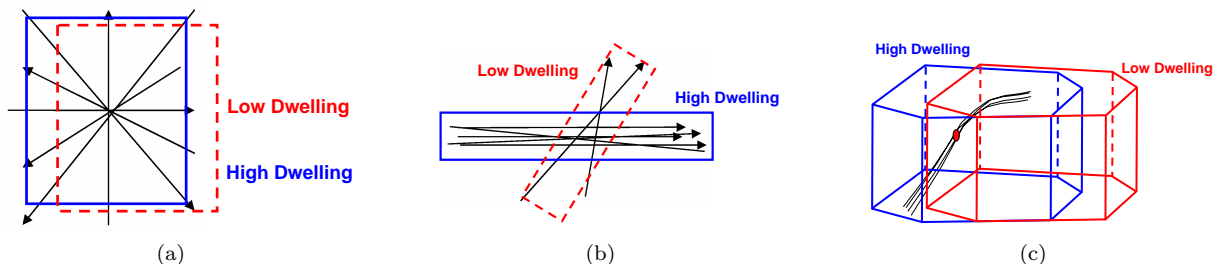


Figure 4. Function of dwell Time. (a) intersection area. (b) 2D dominant flow. (c) 3D dominant flow.

Experiments show the dwell time is a critical metric for a good sector design. As demonstrated in Fig. 4(a), maximizing dwell time will lead to the sector boundary with the blue edges rather than the red one, since the blue one has higher dwell time than the red one. Similarly, in a region which includes a dominant flow, as in Fig. 4(b), the cost of dwell time metric will prefer the blue design instead of the red one. A similar result is obtained when handling 3D traffic. As displayed in Fig. 4(c), the entire major ascent or descent flow will be included in the blue boundaries in order to maintain a higher dwell time than would be obtained with the red boundaries.

B. Intersection Proximity (P_I)

However, using merely the dwell time is not enough. Adjustment should be made by incorporating an intersection-related cost.

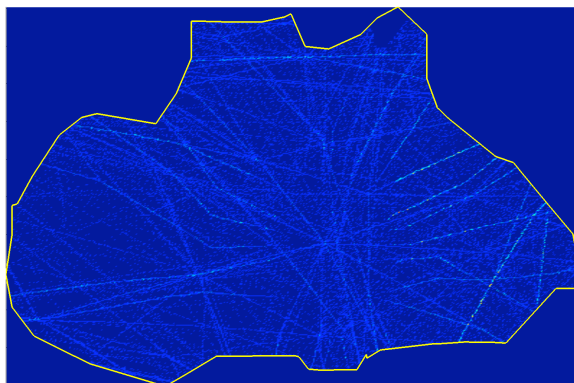


Figure 5. Time elapsed traffic of Fort-Worth (ZFW) center (FL240-999)

To set up such a cost, intersection areas should be identified first. Fig. 5 shows the time-elapsd traffic above FL240 of Fort-Worth (ZFW) center for an entire day. Each pixel corresponds to a grid point with a resolution of 0.01 degree in both latitude and longitude. In order to find the intersection grid points, a window is first defined for each grid point, and the orientation of all traffic entering or leaving the window is computed, as shown in Fig. 6(a). The slope of a trajectory as it crosses the horizontal axis is used to represent the orientation. As a result, a trajectory going in and out of the window will have only one orientation. A histogram of orientations at each pixel is calculated based on the surrounding window. The range of orientations is then divided into 18 bins with 10 degrees for each bin. These bins act as accumulators to count the orientations in a given range. A sample histogram at the pixel of longitude -97.1°E and latitude 32.9°N is displayed in Fig. 6(b).

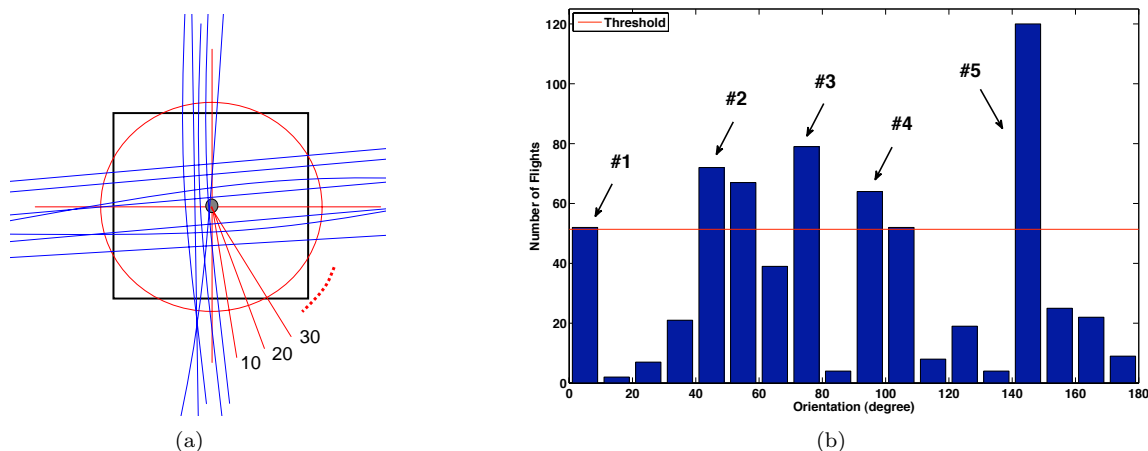


Figure 6. Computation of intersection index. (a) computation of flow orientations. (b) histogram of flow orientations.

Next a threshold is set to be the average pixel value of the entire center. For example, the threshold for the ZFW center is 51.37 and is shown as a red horizontal line in Fig. 6(b). The bars that are higher than the threshold are kept and the others are discarded. Consecutive bars among those that remain are treated as one. At the end of this process, the number of bars that remain is counted as the “intersection index” for that pixel. Any pixel that has “intersection index” greater than or equal to two will be treated as an intersection point. The remaining bars for the sample point are labeled No.1 to 5 in Fig. 6(b), thus the “intersection index” is five, which roughly corresponds to the five distinct flow directions in Fig. 7(a). The

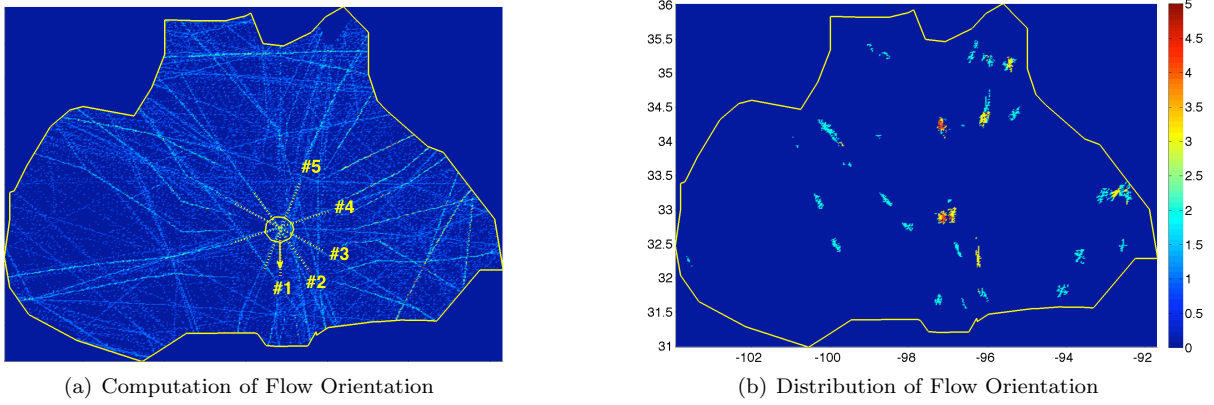


Figure 7. Intersection index map of ZFW center (FL240-999)

center of the circle in Fig. 7(a) is the sample point and the five major flows are marked correspondingly.

An image composed of intersection indexes can be created, as presented in Fig. 7(b). Pixel values in this figure are set equal to corresponding intersection indices. Dark blue means no intersection while a “hot” color denotes a major intersection. To construct a scalar cost, a Gaussian-like penalty (Eqn. 2) is defined according to the distances between sector boundaries and these intersection points. d_i is the minimum distance between pixel i and any boundary. In order to simplify the computation, only the penalty from the closest boundary is taken into account. Γ is a distance threshold that is defined to trigger the penalty. I_i is the index value of pixel i , and c_1 and σ_1 are defined constants.

$$P_I = \sum_i (d_i < \Gamma) \cdot I_i \cdot e^{-\frac{d_i \cdot c_1^2}{2 \cdot \sigma_1^2}} \quad (2)$$

C. Dominant Flow Proximity (P_D)

A boundary may lie on a major flow when pursuing long dwell time. Although the number of sector boundary crossing may not be significant, it definitely lowers the situation awareness of controllers because the boundary is too close to the trajectories. This should be avoided in sector designs, and thus a penalty is needed to prevent this. The penalty should push the boundaries away from the flows or at least let the boundaries cross the flows with near-perpendicular angles.

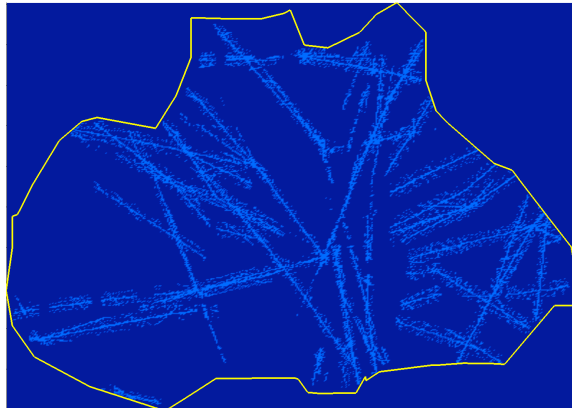


Figure 8. Dominant flow in ZFW center (FL240 and above)

Although there exist other ways to identify dominant flows, the method used here is the same as the one for identifying intersection areas, but with the index threshold changed from two to one. Any pixel with a

nonzero index will be recognized as a dominant flow point. The identified dominant flows in ZFW center are shown in Fig. 8. Note that the dominant flows match well with the time elapsed traffic in Fig. 5. Similarly, a Gaussian-like penalty function is shown in Eqn. 3, where c_1 and σ_2 are defined constants, and others have the same meanings as in the intersection penalty.

$$P_D = \frac{\sum_i e^{-\frac{d_i \cdot c_1^2}{2 \cdot \sigma_2^2}}}{\sum_i I_i > 0} \quad (3)$$

D. Aircraft Count of Short Dwell (N_s)

Because the primary costs try to increase the “average” of sector flight dwell time, some undesired extremely short dwell flights may remain. In experiments, according to observations, most cases can be ameliorated by a minor adjustment. Thus, a cost based on the aircraft count of short dwell is built in. In this work, the flights that have a sector dwell time less than three minutes are counted and used for setting the penalty.

E. The Variance of Sector Peak Aircraft Count (V_p)

If there is no constraint for balancing of sector peak aircraft counts, the rule of “long dwell for high peak” may lead to a partition with tiny pieces containing low traffic areas. The process would be like pie-biting, which is not preferred. Thus the variances of the peak aircraft counts (V_p) for the partitions at the same level are applied to penalize such extreme cases. High costs will cancel out the benefits of long dwell if the partitions are extremely-unbalanced.

F. Primary Cost

A scalar primary cost is computed based on the fundamental metrics discussed above as well as the results of numerous experiments. The optimization is to maximize the primary cost described as:

$$F_p = \frac{\min_j \left\{ \frac{c_0 \cdot M_j}{(N_p)_j} \right\}}{a_1 \cdot a_2 \cdot a_3 \cdot a_4} \quad (4)$$

where:

$$a_1 = e^{P_I} + c_2 \cdot P_D \quad (5)$$

$$a_2 = c_3 + \frac{N_c}{N \cdot (N_p - 1)} \quad (6)$$

$$a_3 = c_4 + \frac{N_s}{N} \quad (7)$$

$$a_4 = \begin{cases} c_5 + V_p, & \text{for } N_a > \Theta, \text{ or } N_a \leq \Theta \text{ and } V_p < N_a \cdot c_6, \\ \infty, & \text{for } N_a \leq \Theta \text{ and } V_p \geq N_a \cdot c_6. \end{cases} \quad (8)$$

The basic principle is to allocate long dwell time for regions with high peak aircraft, which is shown as the numerator on the right side of Eqn. 4. M_j is proportional to the dwell time in sector j , and $(N_p)_j$ is the peak aircraft count in sector j . The remaining “ a ”s are costs for tuning designs. The traffic pattern related penalty is composed of the proximities to intersections and major flows. This is shown in Eqn. 5, where the P_I is the penalty for the intersection proximity and P_D is for major flows. All “ c ”s are constants defined based on experimental observations. The second adjustment is a function of sector boundary crossings as in Eqn. 6, where N_c is the count of sector boundary crossings, N is total number of aircraft, and N_p is the number of partitions. Then the adjustment for extremely short dwell time is established in Eqn. 7. N_s is the count of flights that have a sector dwell time less than three minutes. The final adjustment shown in Eqn. 8 is related to the variance of sector peak counts, where V_p denotes the variance of peak aircraft counts, N_a is the average peak aircraft count over sectors, and Θ is a threshold defined to strengthen the variance constraint when partitions approach the end of the process or sectors become small.

G. Secondary Cost

At each level of the iterative deepening algorithm, a secondary cost is needed to judge which partition among the set of partitions may be good and would lead to a good final partition. The purpose of the secondary cost is to make sure the division follows the nature of the traffic pattern. In this work, based on the observations of experiments, the crossing angles between trajectories and sector boundaries are identified as the key component. The crossing angles for an ideal partitioning should be close to a right angle, which means that in the ideal case the flows either don't cross the sector boundaries or cross them with near-perpendicular angles. The number of crossings should also matter. Therefore the sum of the shallow crossing angles is used as a secondary cost as shown in Eqn. 9. Here θ_k is the crossing angle between a trajectory and a sector boundary for the k th crossing. Of course, one flight can have multiple crossings. Only shallow-angle crossings are counted and Φ is a define threshold for a shallow angle.

$$F_s = \sum_k [(\theta_k - 90^\circ) - \Phi > 0] \cdot [(\theta_k - 90^\circ) - \Phi] \quad (9)$$

H. Design Cost

Design cost serves as a stop criterion. In following experiments, two design costs are used. The first one is the cost for increasing capacity shown in Eqn. 10. The average sector dwell time is used to estimate sector capacity, and the sector peak aircraft count is applied as an estimation of complexity. Then, the residual capacity based on these two estimations are maximized to increase the capacity. Additionally, the high bound of the calculated capacity is limited to be 18, and the lower bound of the demand (the peak aircraft count) is limited to be five. Another design cost requires the peak count of each sector not to exceed a threshold as in Eqn. 11.

$$Model\ I : \begin{cases} Capacity \geq Demand, \\ Capacity = \frac{5}{3} \cdot average_sector_dwell_time, \\ Demand = sector_peak_aircraft_count. \end{cases} \quad (10)$$

$$Model\ II : sector_peak_aircraft_count \leq a\ threshold \quad (11)$$

V. Results

In this work, simulated traffic data were used for sector designs. The traffic was generated from the flight schedule for August 20, 2005, and simulated using the Airspace Concept Evaluation System (ACES)¹³ based on filed flight plans without capacity constraints in both airports and en-route airspaces. Detailed description can be found in Zelinski's work.¹⁰ Both design requirements mentioned in the previous section were tested as stop criteria. The results were obtained on a Mac-Pro computer with a dual-core CPU at 2.8MHz and with the GA code multi-threaded.

In the experiments, unless otherwise specified, the design cost of Model I in Eqn 10 was applied and the "conservative" rule was used for vertical stratifications. To demonstrate the benefits, several cases are studied and compared. The results for high altitude Fort-Worth(ZFW) center are presented. Then the results for combined airspace with FL240 and above are presented and compared. In the comparisons, different design costs are also applied. Finally, design results for Indianapolis(ZID) center, which is typically heavy-loaded and intersection-dominated, are compared to show the adaptability of designs to the flow patterns.

A. Sector Design Comparison for High Altitude Fort-Worth Center

In order to demonstrate the benefit of the new tri-cost strategy over the original method, several cases were studied for high altitude using the simulated unconstrained traffic data between FL240 and FL350 for Fort Worth center (ZFW). First, the results from the original method are presented. Then, the results from the tri-cost method with and without 3D are discussed, respectively. Finally, comparisons are made.

1. Case I: Predefined Number of Sectors

In this case, the original Voronoi-based method from previous work was applied. The design cost of increasing capacity was directly used as the optimization cost. The overall number of partitions and the number of divisions at each level were fixed and defined before the experiments. For example, given a 15-sector design, the first level of partition segregated the airspace into three sectors, and the second level divided each of those into five.

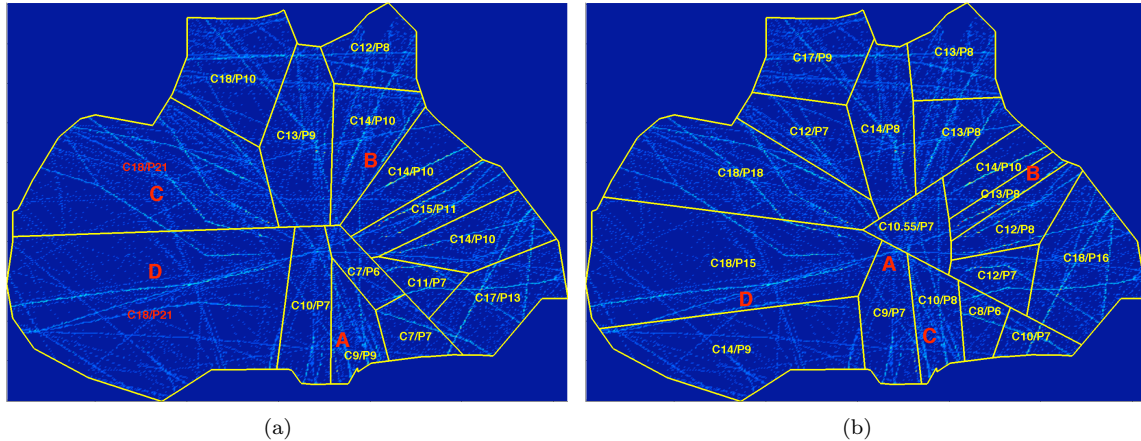


Figure 9. Designs for ZFW center using original method (FL240-350) (a) a 15-sector design. (b) a 18-sector design.

Fig. 9(a) shows a 15-sector design, whereas Fig. 9(b) shows an 18-sector design. In the figures, the “C” means the calculated capacities, and the “P” means the peak aircraft counts of the sectors. If a sector is overloaded, which means the “P” is higher than the “C”, these symbols and values are red, otherwise they are yellow. Because the number of partitions at each level is blindly defined and fixed in advance, parts of the airspace have excessive sectors while others need more sectors. As shown in Fig. 9(a), sector C and D are still overloaded in this 15-sector design. Although an 18-sector design eliminates the overloaded sectors, it introduces unnecessary sectors in region B as shown in Fig. 9(b). Because the design cost is used directly in the optimization, there exist several prohibitive boundaries in both designs. For instance, in Fig. 9(a), sector boundaries are too close to major flows in the A and B regions. And in Fig. 9(b), sector boundaries cross the intersection area in region A and the major flows in regions B, C, and D.

2. Case II: Tri-cost Strategy Without Vertical Stratifications

In this case the new tri-cost method was applied, but vertical stratification was not allowed. The design results are presented in Fig. 10(a). At the first level, a four-sector partition (shown as green dash lines) was picked by the secondary cost. It appears that the secondary cost performs well, because the four-sector partition follows the hub-like traffic pattern and provides a good beginning for subsequent partitions. In the final design, the regions surrounding the center are well sectorized. Sectors such as A and B conform to the major flows. The only undesired part is that the center area containing major flows has been divided into five sectors, C, D, E, F, and G. In order to meet the design requirement, this kind of partitioning sacrifices controllers’ situation awareness because of the proximity between boundaries and flows. Counting these five sectors, the number of sectors required becomes 16.

3. Case III: Tri-cost Strategy With Vertical Stratifications

The previous case demonstrates that when a region contains only one integral traffic pattern, further horizontal divisions may damage the integrity of the traffic pattern and lower controllers’ situation awareness. In this situation, appropriate vertical partitions may resolve the dilemma of either keeping the traffic pattern or meeting design requirements.

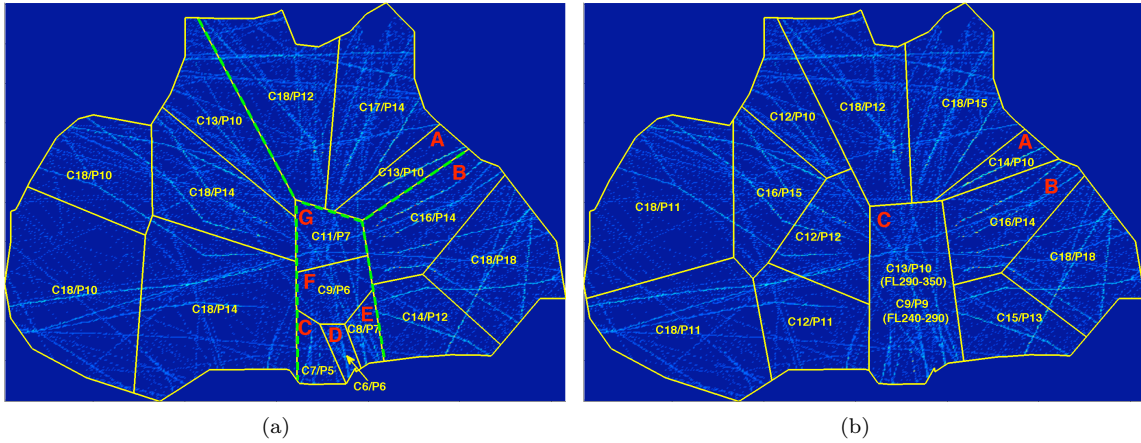


Figure 10. Designs for ZFW center using new method (FL240-350) (a) horizontal partition only. (b) three-dimensional partition.

Fig. 10(b) shows the design produced by the complete version of the tri-cost strategy, which allows “conservative” 3D sectorization. The total number of sectors in this design is 14. Similar to the previous case, a four-sector partition was picked at the first level from the explorations of different partitions. In the final design, the surrounding sectors, especially A and B, follow the traffic patterns well. The center region C is vertically partitioned into two layers. One is from FL240 to FL290 and another from FL290 to FL350. Both layers satisfy the design requirement. With the vertical division, the pattern in region C has been well preserved and the total number of sectors is also lowered. In fact, the number of boundary crossings between the upper and lower layers is 244, which is lower than the 254 crossings among five sectors in the previous case.

4. Comparisons

With the original Voronoi-based method, Case I could not allocate the sectors efficiently. Unnecessary sectors increased the total number of partitions and could damage controllers’ situation awareness. Without the pattern-related cost, sector boundaries were allowed to cross the intersection areas and encroach major flows. Case II utilized the tri-cost method without vertical partitions. Although some problems in Case I were resolved, it created a trade-off situation between reaching the design requirements and keeping integrated traffic patterns within the sectors. Case III resolved these issues using the complete tri-cost method. Appropriate vertical partitions were applied to remove unnecessary partitions and both the integrity of traffic patterns and design requirements were satisfied.

Table 1. Comparison of Designs for ZFW from FL240 to FL350

	Case IA	Case IB	Case II	Case III	Current
Number of sectors	15	18	16	14	19
# of overloaded sectors (Model I)	2	0	0	0	1
Boundary crossings	2,698	2,822	2,368	2,471	2,851
Variance of peak counts	97%	94%	69%	47%	69%
Dominant flow proximity cost	2.89	3.70	2.39	2.14	2.96
Intersection proximity cost	253.7	271.7	4.55	0.97	44.6

In Table 1 several metrics are presented for these cases in order to show the differences quantitatively. The last column shows the measurements for current ZFW sectors. Although the altitude ranges of current ZFW sectors are FL240 and above, to be comparable, the same range of traffic data (FL240 to FL350) was used to calculate those metrics. The “Dominant Flow Proximity” and “Intersection Proximity” are

normalized cost values. A low value means that boundaries are away from flows and intersections. The lower values in the table are preferred and shown in bold. The comparisons show that Case III has the most preferred values and the new tri-cost method is a substantial improvement over the original method.

B. Sector Design for Combined High and Super-high Fort-Worth Center

In the above studies, the sectorization was applied to the airspace between FL240 and FL350, with FL350 chosen as an artificial split between high and superhigh airspace. The purpose of the next set of experiments was to design for combined high and super-high airspace (FL240 and above) without such artifacts. The combined airspace in ZFW center had 4,210 flights on Apr.21 compared with 3,493 flights in the high sectors alone. Three experiments will be presented in the following sections. Increasing capacity was used as design costs in the first two, denoted as “CAP”, and the peak aircraft count was used in the third one, denoted as “PEAK”.

1. CAP2D: Capacity Increasing Design Cost without 3D Partition

In the first experiment, only horizontal partitions were allowed with the tri-cost strategy. The results are shown in Fig. 11(a). In this design most of the sectors capture the flow patterns. However, without vertical partitions, the algorithms have been pushed to the limit. Even though there are 22 sectors, the design still has two saturated sectors, A and B. The sizes of many sectors are small as well, especially sectors C and D.

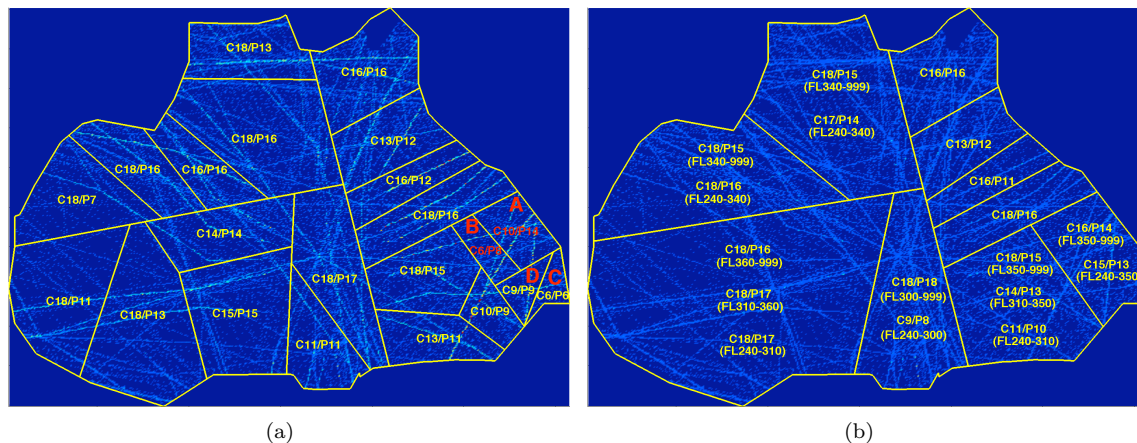


Figure 11. Designs to increase capacity for ZFW center (FL240 and above) (a) CAP2D: horizontal partition only. (b) CAP3D: three-dimensional partition.

2. CAP3D: Capacity Increasing Design Cost with 3D Partition

The next experiment allowed 3D partitions, and the results are shown in Fig. 11(b). With 3D partitions allowed, the integrity of the hub has been preserved. The 3D design has no overloaded sector, and the sector count is only 18. This shows that three dimensional partitions not only produce preferred sector shapes but also dramatically enhance the capability of meeting design requirements. The comparison further testifies that appropriate 3D partitions have a significant advantage over 2D partitions.

3. PEAK3D: Peak Count Capping Design Cost with Stratification

The third case explored a different requirement to show the adaptation of the method. The second design requirement in Section IV.H, which requires the sector peak count not exceed 15, was utilized. Fig. 12 presents the final design. It shows that major traffic patterns have been captured by pursuing the peak aircraft count. The total number of sectors 17 is similar to previous ones.

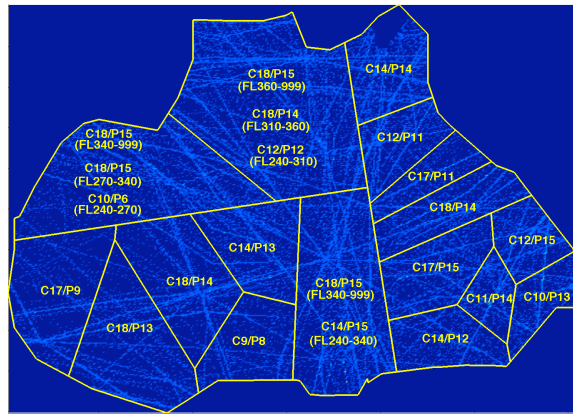


Figure 12. PEAK3D: sector design to cap peak counts with 3D partition for ZFW center (FL240 and Above)

4. Comparison

Comparisons of the above designs are shown in Table 2. “Requirement 1” denotes the first design cost of increasing airspace capacity, while “Requirement 2” refers to the second cost of capping peak aircraft counts. The optimal values for metrics are in bold. Both “CAP3D” and “PEAK3D” achieved their respective design goals. In the table, all optimal values occurred in “CAP3D” and “PEAK3D”. “PEAK3D” has the lowest intersection proximity while “CAP3D” has the rest of the optimal values.

Table 2. Comparison of Designs for ZFW center FL240 and above

	CAP2D	CAP3D	PEAK3D	Current
Number of sectors	22	18	22	19
# of overloaded sectors (Model I)	2	0	4	6
# of overloaded sectors (Model II)	6	7	0	8
Boundary crossings	6,583	4,889	6,160	5,579
Variance of peak counts	60.3%	43.8%	53.5%	78.5%
Dominant flow proximity cost	3.55	2.96	3.30	3.16
Intersection proximity cost	74.1	73.8	60.8	288.5

C. Sector Design for Heavy-loaded Airspace

To explore how the method behaves when sectorizing a heavily-loaded airspace, the new method was applied to ZID center. On the same day, ZID had 6,401 flights based on simulated data. That is 50% more than the traffic volume of ZFW and even more in terms of traffic density because ZID is smaller than ZFW. The other reason for picking the ZID center is that ZID has traffic patterns that are dominated by intersections, which is different from the hub-like patterns in ZFW. Because of the high traffic volume, the “moderate” or “aggressive” stratification was also used for vertical partitions. The first two designs used the requirement of increasing capacity. The third design applied the requirement of capping peak count.

1. CAP3DMOD: Capacity Increasing Design Cost with “Moderate” Stratification

The first “CAP3DMOD” uses the “moderate” rule for vertical partitions, and the result is shown in Fig. 13. The result again demonstrates that when the new method is used, the intersection areas and major patterns are well preserved. Due to the high volume of traffic, even though 3D partitions were used, 11 out of 49 sectors are still overloaded. Regions A and B are stratified because the resulting sectors lower the severity of saturation, although the resulting sectors are still overloaded. The process stopped there because no improvement could be achieved with further divisions of the airspace. Region C is relatively better than A

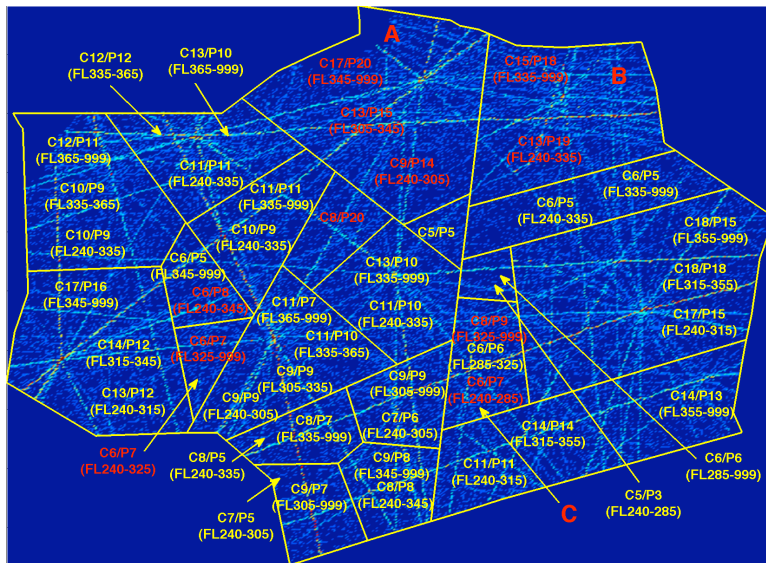


Figure 13. CAP3DMOD: Sector design to increase capacity for ZID center (“moderate” stratification)

and B because one of the vertical stratifications is not overloaded, and the overloaded sectors exceed capacity by only one flight.

2. CAP3DAGR: Capacity Increasing Design Cost with “Aggressive” Stratification

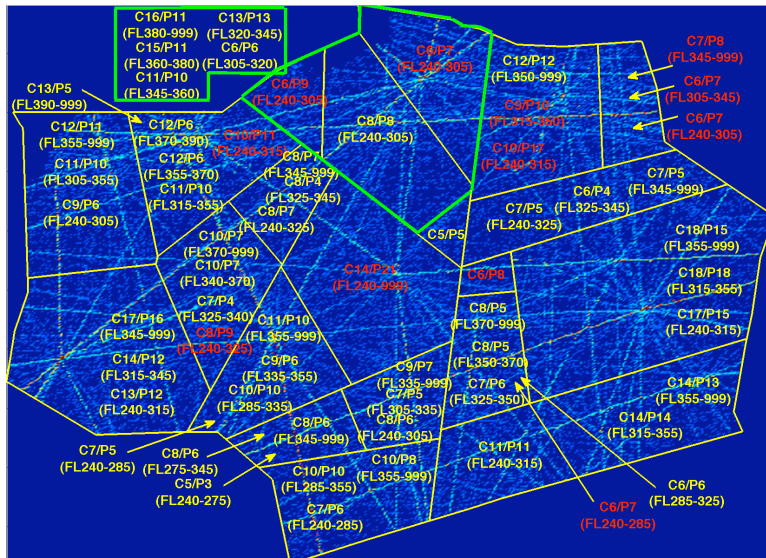


Figure 14. CAP3DAGR: Sector design to increase capacity for ZID center (“aggressive” stratification)

To examine the function of different rules in vertical stratification, the “aggressive” stratification was applied. The result in Fig. 14 shows that the airspace volume of overloaded sectors was less, but the number of sectors increased. 13 out of 62 sectors are saturated. By carefully examining the region surrounded by the green polygon and comparing it to Region A in the “moderate” case, one can see that the “aggressive” rule stratified the region into six layers. Five of them are not overloaded anymore, while the bottom layer from FL240 to FL305 is laterally segregated into three sectors, and one becomes unsaturated. So with a different rule of vertical partition, the exact same region has been divided into eight sectors with two overloaded, instead of three saturated sectors in the previous experiment. Thus, an aggressive use of vertical partitions

may increase the overall capacity and may cause fewer delays, but more stratifications can happen. Overall, these two experiments imply that if the calculated capacity is used as the criterion, then there exists a limit of capacity that a design can provide with a given rule of vertical partition, and further partitions will not increase the capacity anymore.

3. PEAK3DCON: Peak Count Capping Design Cost with “Conservative” Stratification

The third design applied the requirement of capping peak count. The results, shown in Fig. 15, capture the intersection areas and major flows as well. Note that in order to meet the requirement, only 36 sectors were needed and all sectors except one have 15 or fewer aircraft at any one time. The exception is due to the constraint on sector size that was imposed in the design. Because many aircraft were lumped together in that small area, the peak aircraft count could only be lowered by generating small sectors. Theoretically, any peak count requirement could be achieved by chopping the sector into smaller and smaller pieces.

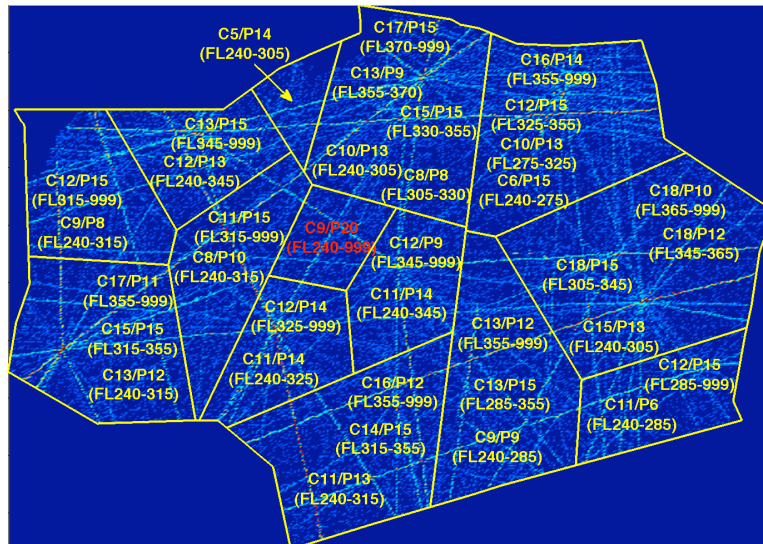


Figure 15. PEAK3DCON: Sector design to cap peak count for ZID center (FL240 and above)

4. Comparison

Table 3. Comparison of Designs for ZID center FL240 and above

	CAP3DMOD	CAP3DAGR	PEAK3DCON	Current
Number of sectors	49	62	36	28
# of overloaded sectors (Model I)	11	13	19	20
# of overloaded sectors (Model II)	6	4	1	15
Boundary crossings	15,241	16,775	13,539	12,341
Variance of peak counts	97.1%	142%	55.5%	66.9%
Dominant flow proximity cost	5.0	5.2	4.1	3.9
Intersection proximity cost	37.9	107.1	31.1	391.0

From the above experiments, it can be seen that it is easier to fulfill the requirement of capping peak count compared to increasing sector dwell time. Table 3 presents a comparison of these two designs with current sectors. The bold values are the lowest compared with others. Obviously, the current system has a high number of violations on both requirements. It is noted that current sectors have the lowest boundary crossings and dominant flow proximity cost. The reason is that the current sectors are much fewer in number,

which causes unfair comparisons in these two metrics. As an extreme example, if the center contains only one sector, these two values would be zero. Taking this fact into account, the new designs perform much better on these metrics. The “CAP3DMOD” and “PEAK3DCON” have the fewest number of violations on the capacity and peak aircraft count limit, respectively. Both follow the traffic pattern well, based on the pattern-related metrics.

VI. Conclusions

This paper presents an enhanced Voronoi-based sector design method, which automatically selects the number of sectors and allows 3D partitions. New costs enforcing traffic pattern conformance have been designed based on numerous experiments. It was found that Voronoi diagrams have the advantage of simplifying the graph partitioning problem but they have limitations when the number of partitions becomes high. Using the iterative deepening algorithm was found to compensate for the limitations caused by Voronoi diagrams, instead of simply speeding up the computation. It is also demonstrated that vertical partitions can enhance sector designs.

The experiments show that the new traffic pattern related costs have the capability of avoiding intersection areas, and enhance the ability of aligning with major flows. In the experiments these new costs lead to designs that capture trajectory patterns, lower sector hands-off, and meet the user’s requirements. Meanwhile, the results also show that the secondary cost can pick out the preferred partitions among the options with different numbers of partitions. It was also found that vertical partitions can be used to preserve integrated traffic patterns while increasing the ability to achieve the user’s design requirements. Since new method enlarges design space by using iterative deepening and vertical partition, it results in less necessary sectors while meeting the capacity constraints. Overall, the new methodology, along with the new costs, substantially improves the original one and makes the sector designs more acceptable.

References

- ¹Delahaye, D., Alliot, J., Schoenauer, M., and Farges, J., “Genetic Algorithms for Automatic Regroupement of Air Traffic Control Sectors,” *Proceedings of the International Conference on Evolutionary Programming*, 1995.
- ²Delahaye, D., Schoenauer, M., and Alliot, J., “Airspace Sectoring by Evolutionary Computation,” *Proceedings of IEEE International Congress on Evolutionary Computation*, 1998.
- ³Trandac, H. and Duong, V., “Optimized Sectorization of Airspace with Constraints,” *5th Erocontrol/FAA ATM R&D Seminar*, Budapest, Hungary, June 23-27 2003.
- ⁴Yousefi, A. and Donohue, G., “Temporal and Spatial Distribution of Airspace Complexity for New Methodologies in Airpace Design,” *4th AIAA Aviation Technology, Integration, and Operations conference (ATIO)*, Chicago, IL, September 2004.
- ⁵Klein, A., “An Efficient Method for Airspace Analysis and Partitioning Based on Equalized Traffic Mass,” *6th USA/Europe Seminar on Air Traffic Management Research and Development*, Baltimore, MD, June 27-30 2005.
- ⁶Drew, M., “Analysis of an Optimal Sector Design,” *27th Digital Avionics Systems Conference*, October 26-30 2008.
- ⁷Martinez, S., Chatterji, G., Sun, D., and Bayen, A., “A Weighted-Graph Approach for Dynamic Airspace Configuration,” *Proceedings of AIAA Guidance, Navigation, and Control Conference*, Hilton Head, South Carolina, August 20-23 2007.
- ⁸Basu, A., Mitchell, J., and Sabhnani, G., “Geometric Algorithms for Optimal Airspace Design and Air Traffic Controller Workload Balancing,” *Proceedings of the 9th Workshop on Algorithm Engineering and Experiments(ALENEX)*, San Francisco, California, January 19 2008.
- ⁹Brinton, C. and Pledgie, S., “Airspace Partitioning using Flight Clustering and Computational Geometry,” *27th Digital Avionics Systems Conference*, October 26-30 2008.
- ¹⁰Zelinski, S., “A Comparison of Algorithm Generated Sectorizations,” *8th USA/Europe Air Traffic Management Research and Development Seminar*, Napa, CA, June 29 - July 2 2009.
- ¹¹Xue, M., “Airspace Sector Redesign Based on Voronoi Diagrams,” *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Honolulu, Hawaii, August 18-21 2008.
- ¹²Okabe, A., Boots, B., Sugihara, K., and Chiu, S. N., *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams (2nd edition)*, John Wiley & Sons, Chichester, UK, 1999.
- ¹³Meyn, L., Windhorst, R., Roth, K., Van Drei, D., Kubat, G., Manikonda, V., Roney, S., Hunter, G., Huang, A., and Couluris, G., “Build 4 of the Airspace Concept Evaluation System,” *AIAA Modeling and Simulation Technologies Conference an Exhibit*, Keyston, CO, 2006.