# A Weighted-Graph Approach for Dynamic Airspace Configuration

Stephane A. Martinez[1]
*Ecole Nationale de l'Aviation Civile, Toulouse, 31400, France*

Gano B. Chatterji[2]
*University of California Santa Cruz, Moffett Field, CA 94035-1000, USA*

Dengfeng Sun[3]
*University of California, Berkeley, CA 94720-1710, USA*

Alexandre M. Bayen[4]
*University of California, Berkeley, CA 94720-1710, USA*

**A method for partitioning airspace into smaller regions based on a peak traffic-counts metric is described. The three setup steps consist of 1) creating a network flow graph, 2) creating an occupancy grid composed of grid cells of specified size for discretizing the airspace and 3) assigning the grid cells to the nodes of the network flow graph. Both the occupancy grid and the grid cell assignment to nodes are computationally realized using matrices. During the run phase of the method, the network flow graph is partitioned into its two sub-graphs and these two sub-graphs and then partitioned into their two sub-graphs, and so on till a termination criterion is met. Weights of the sub-graphs are computed by summing the number of aircraft in each grid cell associated with the nodes of the sub-graphs at each time instant. This process is accomplished by using the occupancy and assignment matrices created during the setup step. The final weight is obtained as the maximum count over a time period. Spectral bisection is then used to split the sub-graph with the maximum weight into its two sub-graphs. Recursive application of the spectral bisection method and weight computation results in the final set of sub-graphs. The grid cells associated with each sub-graph then represent the geometry of the associated sector. Results of sectorization of the airspace over the continental United States are provided to demonstrate the merits and the limitations of the method. The weighted-graph technique created larger sectors in regions of light-traffic and smaller sectors in regions of heavy-traffic. Peak traffic-counts in the sectors were found to be within the range of the Monitor Alert Parameters specified in the Enhanced Traffic Management System.**

## I.  Introduction

This article addresses the problem of partitioning the airspace into smaller regions - sectors such that there are few aircraft in each region at any given time. The maximum-number-of-aircraft limit is primarily governed by the air traffic controller's ability to monitor and provide separation, communication and flow-control services to the aircraft in the sector. Motivation for dynamic airspace configuration (DAC) is to meet airspace user's needs by tailoring the availability and capacity of the airspace, and communicating its status to users. Specific goals of DAC include: 1) allocating and de-allocating airspace for military and special uses, 2) creating high-density airspace

---

1 MS. Student, Computer Science and Air Traffic Engineering; Visiting Graduate Student, Systems Engineering, University of California, Berkeley.
2 Principal Scientist and Task Manager, MS 210-8, Associate Fellow AIAA.
3 Ph.D. Student, Systems Engineering, Department of Civil and Environmental Engineering, Student Member AIAA.
4 Assistant Professor, Systems Engineering, Department of Civil and Environmental Engineering, Member AIAA.

corridors, 3) changing airspace boundaries for balancing airspace complexity, and 4) providing airspace access based upon aircraft performance capabilities.[1]

Past research on goal 3, which is the topic of this paper, offer solutions that can be broadly classified as: a) region-based and b) graph-based. References 2 through 5 present good examples of region-based approaches and Refs. 6 and 7 describe graph-based techniques. These references are discussed in Section II. Region-based techniques, for example in Ref. 2, have the advantage that smaller geometric units such as hexagonal cells can be combined to form sectors, which are larger geometric units. The main limitation of region-based techniques is that they ignore the underlying flow structure. The strength of graph-based techniques, such as the ones described in Refs. 6 and 7, is that they utilize the flow structure. After partitioning the flow graph into its sub-graphs, these techniques assign a geometric attribute to the sub-graphs for defining sectors. For example, when the nodes of the sub-graphs are associated with polygons, the sub-graphs represent sectors containing the polygons associated with the nodes. The method for partitioning airspace described in this paper is a combination of the region-based and graph-based methods. The static portion of the technique consists of creating a network flow graph, discretizing the airspace into cells and associating the cells with the nodes of the graph. The recursive part of the technique consists of weight computation and splitting of the graph into its sub-graphs. The sub-graphs of the previous step are graphs for the next step. All of them are sub-graphs of the original network flow graph. Weight of a sub-graph is obtained as the maximum of the total number of aircraft in all of the cells associated with the nodes of the sub-graph over a period of time, where the number of aircraft in individual cells are added together to obtain the total number of aircraft at each time. The sub-graph with the maximum weight is partitioned using the spectral bisection method. Considerable simplicity and computational advantage is achieved by splitting the graph without considering the weights in the splitting process. Weights are only used to select a sub-graph for further partitioning.

The rest of the paper is organized as follows. Section III provides an outline of the secorization technique. Steps needed for setting up the algorithm: network flow graph creation, occupancy grid creation and assignment of grid cells to nodes of the flow graph are described in Section IV. Section V describes the weight computation and flow graph partitioning steps needed during the run phase of the algorithm. Sectorization results for the continental United States obtained using the algorithm are discussed in Section VI. Finally, the paper is summarized and concluded in Section VII.

## II.   Background

Reference 2 described a technique in which the airspace is discretized using hexagonal grid-cells. Workload values are computed using a simulation and assigned to these grid-cells. A linear-programming-based clustering is then used to group the grid-cells such that the resulting sectors have similar workload values. Reference 3 proposes aircraft count as a metric and describes an efficient algorithm for locating aircraft in hexagonal grid-cells. A fixed number of seed locations is chosen, and a region growing algorithm is then used for merging grid-cells starting from the seed locations. The region growing process is continued with the objective of creating regions having equal traffic-counts. Reference 4 discusses a genetic programming algorithm. The method consists of creating sets of x-y locations using crossover and mutation operations starting from an initial population of x-y locations. Each set of points represents a possible sectorization in terms of the Voronoi polygons associated with their x-y locations. Sets that represent good sectorization are favored for creating new sets using crossover and mutation operations. Methods for cutting regions into sub-regions using binary space partitions and pie-cuts are described in Ref. 5. Salient features of this approach include performing one-dimensional sectorizaton along lines at different orientations using a greedy algorithm for determining cuts that create equally weighted regions. Examples of graph-based approaches are given in Refs. 6 and 7.  The method described in Ref. 6 uses aircraft trajectories to build a flow network with nodes, located where trajectories intersect, and edges connecting the nodes. The edges are assigned a weight for representing a workload metric. Genetic programming is then used for partitioning the flow network (graph) into sub-graphs such that their weights, obtained by summing the weights of the edges, are nearly equal. The technique in Ref. 7 first partitions the weighted graph recursively into sub-graphs using the heuristic technique described in Ref. 8 and then locally re-constructs the sub-graphs to equalize their weights using a constraint programming technique. Sectors are obtained by grouping the Voronoi diagrams associated with the nodes of the sub-graphs.

## III.   Dynamic Airspace Configuration Algorithm

Airspace configuration is generally understood to be a partitioning of the airspace into smaller regions in which air traffic controllers are able to ensure that aircraft maintain separation. These smaller regions known as sectors are defined as three-dimensional geometric objects and located with a geographic reference. The purpose of sectorization, however, is to create a geometric framework for distributed control such that separation and flow

American Institute of Aeronautics and Astronautics

control services can be provided to the aircraft within the confines of each sector. This construct of distributing control by partitioning the airspace into sectors is dependent on whether a human controller or automation creates separation assurance and flow control solutions. In a human-centered air traffic management, sectorization is guided by the way traffic is organized to assist human controller's cognitive situational awareness, which leads to identification of future violations of separation minimums. Traffic organization often places departure, arrival and overflight traffic in separate streams to eliminate conflicts between them. For example, this organization eliminates conflict between climbing and descending aircraft. Aircraft with diverse performance characteristics such as jets and turboprops are often kept in separate flows to reduce the complexity of conflict resolution.

The extent to which traffic organization is needed in an automation-centered system, where automation creates the solution, compared to a human-centered system is not obvious. Although automation depends on flight plans and surveillance derived aircraft position data in exactly the same manner as the human-centered system, it has the advantage in its ability to exhaustively examine the combinations of parameters and conditions that could lead to undesirable situations in the future. For example, an automated conflict detection algorithm can determine the possibility of conflict between two aircraft based on the assumptions 1) both aircraft fly along their flight plan, 2) one flies along the flight plan while the other continues along its present trajectory, and 3) both continue to fly along their present trajectory.[9] Similarly, a conflict resolution algorithm can determine the combinations of altitude, speed and heading maneuvers that would resolve conflicts between several aircraft.[10] If robust conflict detection and resolution automation is accomplished and if airport capacity constraints are met during flight operations, the need for sectorization, especially high-altitude sectorization, is not apparent. Sectorization of airspace is therefore viewed within the context of human-centered system in this paper.

Given that 69% of air transport and air taxi operations in the United States are conducted between the 35 major airports (with one in Hawaii) included in the Federal Aviation Administration's (FAA) Operational Evolution Plan (OEP), the flight paths between the 34 OEP airports that are located within the continental United States establish the primary traffic pattern in the airspace over the continental United States. The underlying topological structure, pathways or routes, of a flow pattern can be viewed in terms of a graph. The flow, measured for example as number of aircraft, along a link of the graph can then be assigned as the weight of that link or as the weight of the nodes of that link. Thus, the flow pattern can be mathematically posed as a weighted graph, where the weights change with time. With this representation, the objective of partitioning the airspace to control the air traffic flow can be achieved by partitioning the weighted graph into sub-graphs to represent the desired sectorization. The algorithm for sectorization that is described below is based upon this central idea.

The algorithm consists of the steps portrayed in Fig. 1. These steps can be categorized as setup and recursive partitioning steps. The flow graph creation, occupancy grid creation and grid cell assignment steps are needed for setting up the airspace sectorization problem. These functions are indicated by the blocks labeled: *Network Flow Graph*, *Occupancy Grid* and *Grid Cell Assignment* in the flow chart shown in Fig. 1. Weight computation and graph partitioning steps, indicated by the *Weight Computation* and *Flow Graph Partitioning* blocks in Fig. 1, are done recursively to partition the initial graph into smaller graphs until the desired number of such graphs or the desired weights associated with these graphs are obtained. The smaller graphs (sub-graphs of the initial graph) are then labeled to represent sectors in the final step for obtaining the desired airspace configuration.

The two setup steps for creating the flow graph and populating the occupancy grid are based on the flight plan and aircraft position data. Both the flight plan and the aircraft position data needed were obtained from the Aircraft Situation Display to Industry (ASDI) data, which is provided by the FAA's Enhanced Traffic Management System (ETMS). Details of flow graph creation and the
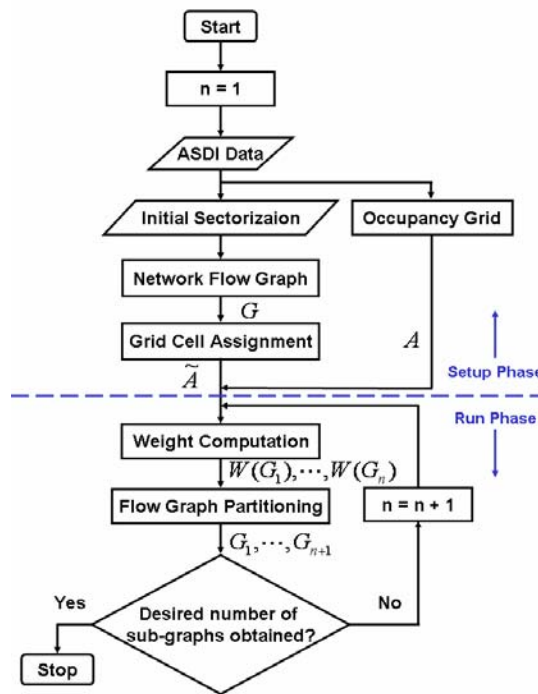


**Figure 1. Steps of the dynamic sectorization algorithm.**

procedure for populating the occupancy grid are described in Section IV. The third setup step of grid cell assignment to each node of the flow graph is also discussed in Section IV.

The next step of creating a weighted graph is based on the three setup steps. The flow graph provides the nodes that need to be assigned a weight. The grid cell assignment technique assigns cells in the neighborhood of nodes to those nodes of the flow graph, and the filled occupancy grid provides the weight associated with each cell. Adding the weights associated with each cell assigned to the node determines the weight of the node. This process is outlined in Section V. Section V also describes the use of spectral bisection method for recursively partitioning the graphs into their sub-graphs till the termination criteria of the algorithm is met. Finally, examples are presented in Section VI to illustrate the sectors that were obtained based on the sub-graphs created by the spectral bisection method.

## IV. Algorithm Setup

This section describes the three steps that are needed for setting up the sectorization algorithm. The procedure for creating the network flow graph is described in Subsection A. Subsection B defines the occupancy grid and provides details of the process used for assigning numerical values to elements of the corresponding occupancy matrix. Finally, the method employed for assigning cells of the occupancy grid to each node of the flow graph is discussed in Subsection C.

### A. Network Flow Graph

In order to construct the network flow graph, the flow pattern needs to be known. Air traffic flow pattern can be obtained in a few different ways. It can be obtained by plotting the flight plans, given in terms of a sequence of latitude-longitude pairs, on a flat plane. Traffic pattern can also be obtained by plotting the actual track histories of the aircraft by processing the ASDI data. Aircraft position data derived from ASDI data can be used for counting the number of aircraft located in each cell of a grid projected over the airspace. The sum of these counts over the duration of one day then represents the flight trajectories and hence the traffic flow pattern.

A convenient mathematical representation of the grid is a matrix whose elements correspond to the cells within the grid. Each element of the matrix is uniquely labeled by its row and column indices. Unique mapping between the cells and the matrix elements enables assignment of the aircraft count in the cells as numerical values of the elements. The resulting matrix can then be displayed as an image. Fig. 2 shows such an image of the aircraft count matrix derived from 10 February 2007 ASDI data. The intensity level of an element, blue to red, represents the sum of the number of aircraft at or above 18,000 feet altitude that were present at the corresponding grid cell location during the 24-hour period. This figure clearly shows the busy East Coast, West Coast and the transcontinental routes. Highly congested routes in the Northeastern Corridor are observable by the number of intersections between the routes in that region of the image. The other routes which are not part of the main flows are also visible in Fig. 2.

Given an image such as that in Fig. 2, the graph creation procedure seeks to represent the routes with a node-link graph. There are several different techniques that one can employ for constructing a graph from an image. One can start by thresholding the aircraft count matrix to identify the elements that have a value greater than the chosen threshold value. A graph construction technique such as the minimum spanning tree algorithm (see Ref.
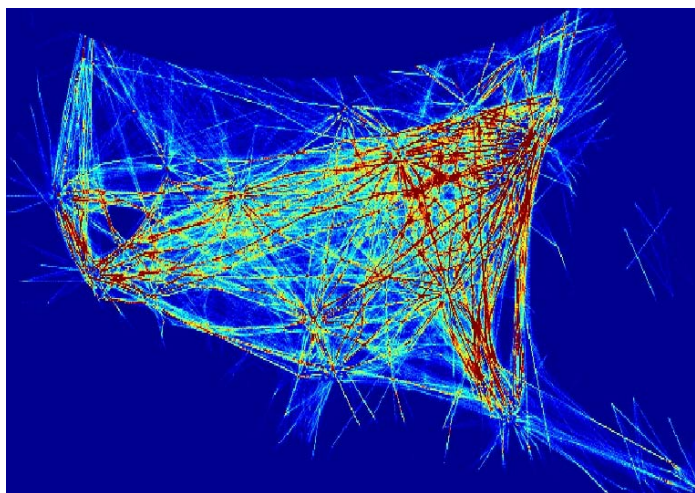


**Figure 2. Image of aircraft count matrix depicting traffic flow pattern in the continental United States.**

11) can then be used for creating a loop-free graph that links all the elements identified in the thresholding step. Image processing techniques such as Hough Transform and contour following can also be used for discovering the underlying flow pattern created by the flight paths.[12] In addition to these graph discovery techniques, track history

and flight plans of flights can be processed using the aggregation technique described in Ref. 13 for flow graph synthesis.

The aggregation technique assumes an underlying sectorization such as the one shown in Fig. 3 with eight sectors surrounding sector ZOA33 in the Oakland Center. Sectors with the first three letters ZOA, ZLC and ZLA are in the Oakland Center, Las Vegas Center and Los Angeles Center. Track history of the flights through these sectors is shown. The first step of the aggregation method consists of determining the entry and exit location through each sector for every flight. The second step consists of identifying all the flights that enter or leave a given sector through a boundary that is shared with a given neighboring sector. For example, this step identifies all flights that cross the ZOA33-ZOA34 boundary when entering ZOA33 from ZOA34 or leaving ZOA33 for ZOA34.

Next, the position coordinates of these flights at the boundary crossing are averaged to determine the location of the boundary crossing node. This process is repeated for every boundary of the sector to determine the boundary crossing nodes of that sector. Note that a single boundary crossing node is associated with a boundary. Pairs of boundary crossing nodes are connected via a link if there are flights that enter the sector through one boundary and leave the sector through the other boundary. For example, Fig. 3 shows that there are flights that enter sector ZOA33 through ZOA34-ZOA33 boundary and leave sector ZOA33 through ZOA33-ZLC45 boundary. Thus, a flow path exists between ZOA34-ZOA33 and ZOA33-ZLC45 boundaries. This path is modeled as a link between the two associated boundary crossing nodes. Figure 4 shows the links and nodes obtained by aggregating the flows between pairs of neighboring sectors of sector ZOA33.

Once boundary crossing nodes and links between them are obtained for every sector in the airspace, the flow graph is completely specified. Since the flow path disregards flow direction by construction, the resulting network flow graph is non-directional. The network flow graph obtained by processing the January 1, 2005 ASDI data with the aggregation algorithm described above is shown in Fig. 5. This graph has 807 boundary crossing nodes.



**Figure 3. Flight paths of aircraft crossing sector ZOA33.**
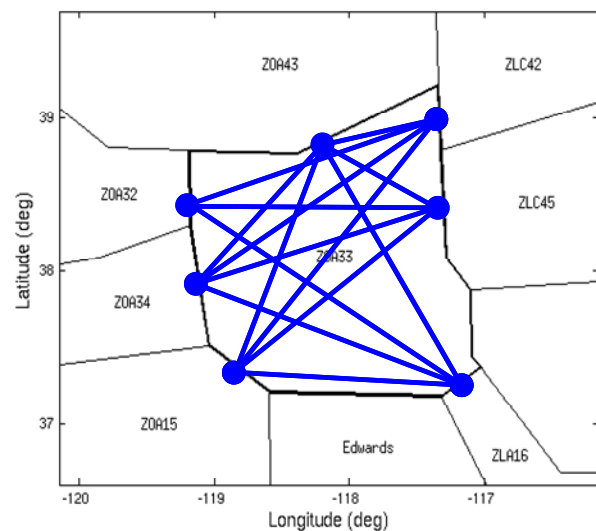


**Figure 4. Flow graph segment through sector ZOA33.**

## B. Occupancy Grid

In order to count aircraft at discrete locations, the entire airspace over the United States is discretized using five nautical miles by five nautical miles square regions, which are termed grid cells. Bounding the US airspace with a rectangular region (bounding box), whose Cartesian coordinates of the top-left-corner correspond to 46.78° latitude and -134.17° longitude and of the bottom-right-corner correspond to 17.19° latitude and -63.27° longitude, results in
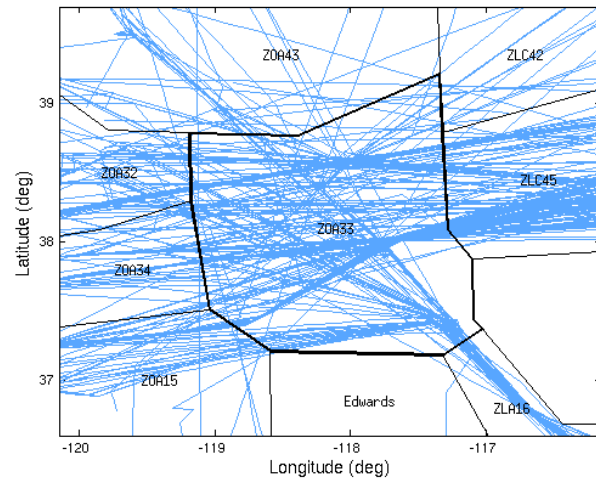
5

the grid with 307,211 grid cells arranged in 361 rows and 851 columns. The two projection equations needed for obtaining the Cartesian coordinates of the corner locations of the bounding box are described below.

Given the track data of all aircraft, each aircraft has to be assigned to one of the grid cells. Since track data, which consists of latitude, longitude and altitude, are provided by the ASDI data at a periodic rate of one minute, the aircraft counts are also computed at one minute intervals. The latitude and longitude coordinates, which are angular measures, have to be transformed into Cartesian coordinates for mapping to the grid cells. A transformation that is commonly used for visualizing the position of aircraft on air traffic displays is known as the Oblique Stereographic Map Projection. The equations for accomplishing this projection are defined in Eq. (1) and (2).

$$X_P = 2R_E \cdot \frac{\cos\lambda \cdot \sin(\tau - \tau_T)}{1 + \sin\lambda_T \cdot \sin\lambda + \cos\lambda_T \cdot \cos\lambda \cdot \cos(\tau - \tau_T)} \tag{1}$$

$$Y_P = 2R_E \cdot \frac{\cos\lambda_T \cdot \sin\lambda - \sin\lambda_T \cdot \cos\lambda \cdot \cos(\tau - \tau_T)}{1 + \sin\lambda_T \cdot \sin\lambda + \cos\lambda_T \cdot \cos\lambda \cdot \cos(\tau - \tau_T)} \tag{2}$$

$R_E$ is the radius of the Earth in nautical miles; latitudes $\lambda$ and $\lambda_T$, and longitudes $\tau$ and $\tau_T$ are in radians.

Using these equations, a location specified by a latitude-longitude pair $(\lambda, \tau)$ is transformed to the location specified by the Cartesian coordinates $(X_p, Y_p)$ with respect to the frame of reference whose origin is located at the point of tangency. The point of tangency is provided as the latitude-longitude pair $(\lambda_T, \tau_T)$. If one were to substitute the coordinates of the tangency point as the latitude-longitude coordinates $(\lambda, \tau)$ in Eqs. (1) and (2), one would obtain $X_p = 0$ and $Y_p = 0$. The x-axis of the frame of reference is tangential to the latitude and the y-axis is tangential to the longitude at the point of tangency. The point of tangency was chosen to be 37.98 degrees latitude and -98.35 degrees longitude, which is near the middle of the bounding box. It should be noted that projection does introduce errors in that the distance between two points on the projected plane is not equal to the great circle distance between the original two points located on the surface of the Earth. The absolute position error increases with increasing distance from the point of tangency. The relative position error between two nearby aircraft however remains small.
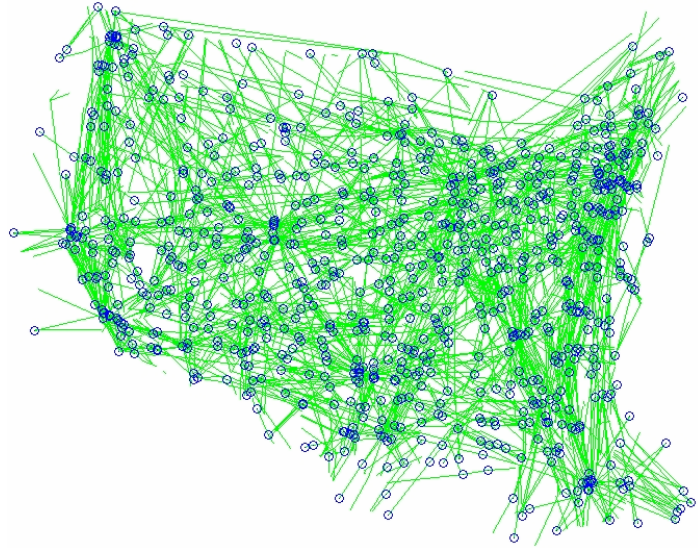
To obtain coordinates in matrix format, the origin of the frame of reference can be translated to the upper-left-corner of the



**Figure 5. Complete network flow graph.**

bounding box. The location of the upper-left-corner of the bounding box corresponds to 46.78° latitude and -134.17° longitude, therefore its position with respect to the x and y axes of the Oblique Stereographic frame of reference are: $U_X = -2149$ and $U_Y = 854$. The translated coordinates $X_T$ and $Y_T$ are then obtained as:

$$X_T = X_P - U_X \tag{3}$$

$$Y_T = Y_P - U_Y \tag{4}$$

These coordinates can be transformed into the row and column indices of the occupancy matrix $A$ that has 361 rows and 851 columns corresponding to the occupancy grid. Assuming that the grid cells are of size $l$, the row index '$i$' and the column index '$j$' are given as follows.

$$i = \left\lfloor \frac{-Y_T}{l} \right\rfloor + 1 \tag{5}$$

$$j = \left\lfloor \frac{X_T}{l} \right\rfloor + 1 \tag{6}$$

where $\lfloor x \rfloor$ is the floor of $x$, defined as the largest integer smaller than $x$.

The procedure for counting aircraft is begun by initializing the value of every element of the occupancy matrix $A$ to zero. Each time an aircraft is mapped to a grid cell, the previous value of the corresponding matrix element indicated by the row and column indices computed via Eqs. (5) and (6) is augmented by one. Once this process is completed for all the aircraft reported at the end of one ASDI update cycle, the accumulated value in each element is the number of aircraft located in the corresponding grid cell. Figure 6 depicts the image of the portion of the $A$ matrix corresponding to the Western region of the United States. This $A$ matrix was computed using track data of aircraft derived from 10 February 2007 ASDI data. The image of the entire $A$ matrix was shown earlier in Fig. 2.
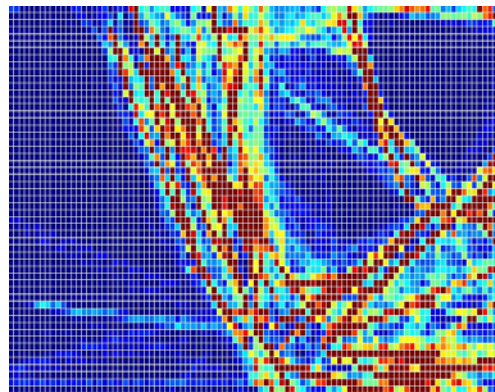


**Figure 6. Image of the portion of the aircraft count matrix corresponding to the Western United States.**

### C. Grid Cell Assignment

Given an occupancy grid of the type shown in Figs. 2 and 6 and nodes of the network flow graph shown in Fig. 5, the grid cell assignment process associates grid cells to nodes based on a distance metric. The guiding notion is that the weight of the node is derived from the weight, aircraft count, of nearby grid cells. Since grid cells are equivalent to the elements of the occupancy matrix $A$, the concept of neighborhood in the grid domain is equivalent to the concept of proximity between elements in the matrix domain. Because of these reasons, the reader should interpret the steps of the grid cell assignment procedure, described by the ensuing examples, in terms of matrix operations.

Consider an occupancy grid with 54 grid cells and a seven-node, six-link network flow graph superimposed on it as shown in Fig. 7. The simplest technique for assigning



**Figure 7. Occupancy grid and seven-node six-link graph.**

the grid cells to the nodes is to compute the Euclidean distance from the center of each grid cell to each node, and then, based on that assign each grid cell to the closest node. Results obtained using this process are shown in Fig. 8. This procedure, however, is computationally taxing. For example, 248 million distance computations are required for assigning the 307,211 grid cells of the occupancy matrix shown in Fig. 2 to the 807 nodes of the network flow graph shown in Fig. 5. A computationally efficient algorithm is therefore desirable. One such algorithm, which is based on the idea of successive assignment of grid cells, is discussed next.

The region growing algorithm is begun by assigning each grid cell that a node lies in to that node. These grid cells are shown in Fig. 9. In the next step, a node is selected, and the grid cells surrounding the previously assigned
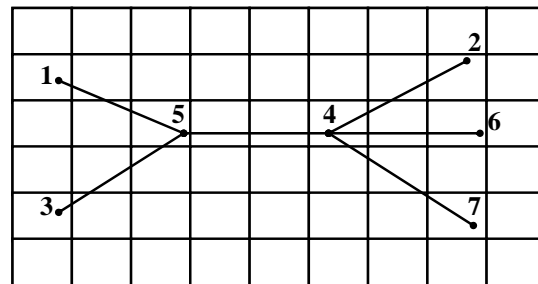
7
American Institute of Aeronautics and Astronautics

grid cell are then assigned to that node. This process is repeated for all the nodes. Once a grid cell is assigned to a node, it is not assigned to a different node. The results of the second step are shown in Fig. 10. Observe that the grid cells were assigned to the nodes in the following order: Node 1, Node 5, Node 3, Node 4, Node 2, Node 6 and Node 7. Although the assignment depends on the order in which the nodes are considered, the effect on the final sectorization can be expected to be small. Nodes can be selected randomly to mitigate such effects. The second step is applied recursively to add the surrounding grid cells to the groups of grid cells assigned in the previous iteration to form new groups. The process is halted once all the grid cells are assigned. The final grid cell assignment, achieved in three steps, for the example in Fig. 7 is shown in Fig. 11.
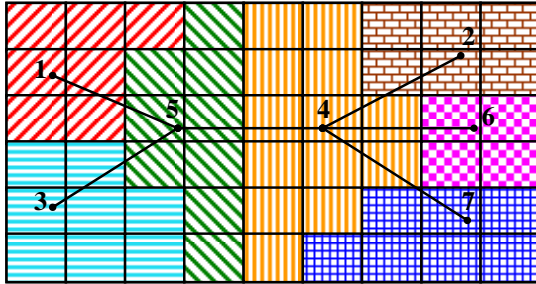


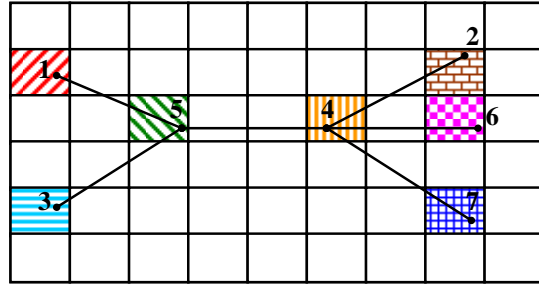**Figure 8. Grid cell assignment based on Euclidean distance.**



**Figure 9. Initialization step of region growing.**
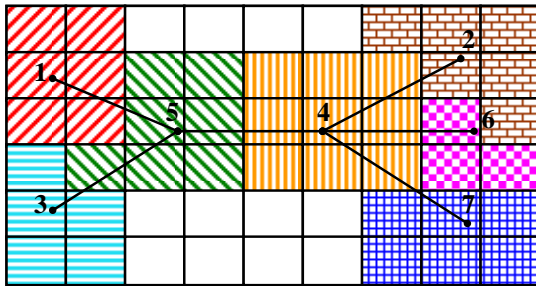


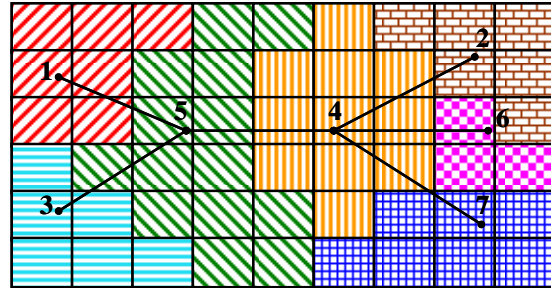**Figure 10. Second step of region growing.**



**Figure 11. Final step of region growing.**

The region growing algorithm is easily implemented using an assignment matrix $\tilde{A}$ of the same size as the occupancy matrix $A$. Initially, each element of $\tilde{A}$ is given a negative value. As grid cells are assigned to nodes, negative values of corresponding elements are replaced with node numbers. For the assignment shown in Fig. 11,

$$
\tilde{A} = \begin{bmatrix}
1 & 1 & 1 & 5 & 5 & 4 & 2 & 2 & 2 \\
1 & 1 & 5 & 5 & 4 & 4 & 4 & 2 & 2 \\
1 & 1 & 5 & 5 & 4 & 4 & 4 & 6 & 2 \\
3 & 5 & 5 & 5 & 4 & 4 & 4 & 6 & 6 \\
3 & 3 & 5 & 5 & 5 & 4 & 7 & 7 & 7 \\
3 & 3 & 3 & 5 & 5 & 7 & 7 & 7 & 7
\end{bmatrix}
\tag{7}
$$

Initializing the elements with a negative value is useful for checking if the corresponding grid cells have already been assigned to nodes or are available for assignment during the recursive step of the region growing algorithm discussed in the previous paragraph.

## V. Algorithm Run Phase

This section describes the two remaining steps consisting of 1) nodal and flow graph weight computation, and 2) flow graph partitioning, which is used recursively to obtain the final sectorization. The procedure for computing the weights of the nodes of the flow graph and of the graph is discussed in Subsection A. Subsection B explains the spectral bisection method and illustrates its utility for partitioning the flow graph into its sub-graphs via a numerical example.

### A. Weight Computation

The primary objective of sectorization is partitioning of the airspace based on workload considerations. Since workload is a function of the number of aircraft in the airspace, metrics that are based on number of aircraft can be used to guide the sectorization process. This subsection describes two such metrics: average traffic-count and peak traffic-count that can be computed as weights of the sub-graphs. Sub-graphs with higher weights can be split further to reduce the weights below an acceptable threshold to meet the workload constraint.

To compute the weights of the nodes and of the graph, the occupancy matrix $A$ and the node assignment matrix $\tilde{A}$ are needed. Recollect, each element of matrix $A$ contains the number of aircraft that occupied the corresponding grid elements at a given time, and each element of matrix $\tilde{A}$ contains the nodal number (node ID) of the node associated with the grid location corresponding to the element. The process of weight computation using these matrices for achieving sectorization based on time-averaged-number-of-aircraft (average traffic-count) is described next.

The average number of aircraft in grid cells over a specified period of time is obtained as follows. An occupancy matrix is created using ASDI data corresponding to the time that marks the beginning of the specified time period. An accumulator matrix is then initialized using this occupancy matrix. Subsequently, the elemental values of the accumulator matrix are updated by adding the number of aircraft found in the corresponding grid cells once a minute in unison with the end of each ASDI data update cycle. The average is then obtained by dividing the elemental values by the number of cycles in the time interval specified for averaging. Thus, the average at the end of $N$ cycles is

$$\overline{A} = \frac{\sum_{1 \le k \le N} A_k}{N} \tag{8}$$

where the occupancy matrix at the '$k$' time-step is denoted by $A_k$.

Nodal weights can now be computed using the assignment matrix $\tilde{A}$ as follows. A nodal weight vector $W$ containing the same number of elements as the number of nodes in the flow graph is initialized to zero. The first element of the $\tilde{A}$ matrix is visited, and the node number stored is retrieved. The value of the first element of the $\overline{A}$ matrix is then retrieved and added to the previously stored value in the element of the $W$ vector associated with the retrieved node number. These two steps of retrieving nodal numbers from $\tilde{A}$ matrix and time-averaged-aircraft-counts from $\overline{A}$ matrix are repeated for all the elements of these matrices to complete the nodal weight computations. This process is compactly described by Eq. (9),

$$W(m) = \sum_{\substack{1 \le i \le I \\ 1 \le j \le J}} \overline{A}(i,j) \times \left[ m = \tilde{A}(i,j) \right] \tag{9}$$

The number of rows and columns are $I$ and $J$. Iverson's square-bracket notation indicates that the logical expression enclosed within the brackets takes a value of one if true and zero if false.[14] Note that $m$ is the nodal number.

Finally, the weights of the sub-graphs are obtained by summing the weights of the associated nodes. This is possible because of the distributive property of the chosen metric, temporally-averaged-number-of-aircraft. If a non-

distributive metric such as peak traffic-count (maximum-number-of-aircraft) is chosen, the computation of weights of individual nodes is not meaningful. All the nodes of the sub-graph have to be considered together for determining the weight. This is described next.

Given a sub-graph '$G$' with '$M$' nodes, grid cell assignment matrix $\tilde{A}$, and occupancy matrices $A$ computed at each time interval, the peak traffic-count within the specified interval is obtained as follows. The weight of the sub-graph at a time instant '$k$' is obtained by summing the weights of the nodes of the sub-graph. Thus,

$$W_k(G) = \sum_{1 \leq m \leq M} \sum_{\substack{1 \leq i \leq I \\ 1 \leq j \leq J}} A_k(i, j) \times \left[ m = \tilde{A}(i, j) \right]$$
(10)

Note that $A_k$ is the occupancy matrix corresponding to the time instant $k$. The weight associated with the peak traffic-count is obtained as,

$$W(G) = \max_{1 \leq k \leq K} (W_k(G))$$
(11)

where $K$ discrete time steps are required for spanning the specified time interval. It should be apparent that the weight of the sub-graph cannot be computed by storing the weights of individual nodes separately till the end of the time interval and then computing the weight of the sub-graph in a single step. One could compute the maximum-number-of-aircraft at each node over the specified interval and then sum these maximum counts to obtain the weight of the sub-graph, but this would not be the peak traffic-count of the sub-graph. Peak traffic-count is obtained as the maximum of the sum (maximum number) of aircraft in a region, which is defined by several nodes taken together as a sub-graph. Peak traffic-count in the sector is used as the primary metric for judging workload in the current air traffic control system.

After selecting a graph or sub-graph based on its weight, it is partitioned into its two sub-graphs using the spectral bisection method, which is discussed next.

### B. Flow Graph Partitioning

The sectorization problem in two or more dimensions is known to be NP-hard; therefore, no deterministic polynomial time algorithm can be used to solve the problem optimally. Heuristic approaches, which do not guarantee optimality, are often employed instead.[5] This section describes a heuristic approach for sectorization based on spectral bisection. The spectral bisection method is used to recursively partition the flow graph into its sub-graphs, and the sub-graphs into their sub-graphs till a termination criterion is reached. The convenient termination criteria are the desired number of sub-graphs and desired maximum weight of the sub-graphs. Since sub-graphs denote sectors, these criteria also mean desired number of sectors and peak traffic-count (weight) in the sectors.

Spectral bisection method splits a given graph into its two sub-graphs such that the node-link connectivity is preserved in each sub-graph.[15] The number of links which are cut during the bisection is also minimized. Considering the seven-node six-link flow graph example in Fig. 7, the topological relationship between the nodes can be described by the Laplacian matrix:

$$L(G) = \begin{pmatrix} 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & 4 & -1 & -1 & -1 \\ -1 & 0 & -1 & -1 & 3 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 1 \end{pmatrix}$$
(12)

The rows and columns are ordered in terms of the nodal numbers. For example, the first row describes how Node 1 is related to the other nodes of the graph. A -1 at the 5$^{th}$ column of this row indicates that Node 1 is linked to Node 5. Similarly, -1 values at the 2$^{nd}$, 5$^{th}$, 6$^{th}$ and 7$^{th}$ columns of the fourth row indicate that Node 4 is connected to nodes 2, 5, 6, and 7. The diagonal element is the number of neighboring nodes that the node corresponding to the selected row is connected to. Since Node 4 is connected to its four neighboring nodes, the fourth diagonal element has a value of 4. Observe that the sum of each row is thus zero. The fact that Node 4 is connected to its four neighboring nodes means that these four nodes are also connected to Node 4. Due to this symmetric relationship between connected nodes, the columns of the Laplacian matrix also adds up to zero.

Since the matrix in Eq. (12) is symmetric and positive-semidefinite, the eigenvalues are non-negative and real, and the corresponding eigenvectors are real. Eigenvectors are orthogonal if the eigenvalues are distinct. There are as many eigenvalues as the number of nodes of the graph. The number of sub-graphs, which consist of a set of connected components, is equal to the number of zero eigenvalues. One zero eigenvalue means a single graph. If the Laplacian matrix describes two sub-graphs, two of the eigenvalues will be zero. Thus, the second eigenvalue $\lambda_2$ (in non-decreasing order: $0 \leq \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_M$ ) is not equal to zero, if and only if the graph is connected. The second eigenvalue is called the algebraic value in graph theory, and the eigenvector corresponding to it is called the *Fiedler vector*. The positive and negative elements of the Fiedler vector organize the nodes into two sub-graphs.

The seven eigenvalues of the Laplacian matrix in Eq. (12) are:

$$\lambda_1 = 0$$
$$\lambda_2 = 0.3983$$
$$\lambda_3 = 1.0000$$
$$\lambda_4 = 1.0000 \quad\quad (13)$$
$$\lambda_5 = 1.0000$$
$$\lambda_6 = 3.3399$$
$$\lambda_7 = 5.2618$$

and the eigenvector for the second eigenvalue is:

$$q_2 = \begin{pmatrix} 0.4929 \\ -0.3560 \\ 0.4929 \\ -0.2142 \\ 0.2966 \\ -0.3560 \\ -0.3560 \end{pmatrix} \quad\quad (14)$$

Observe that the first eigenvalue of zero indicates that this Laplacian matrix describes a single graph. The second eignvalue is non-zero, which confirms that the graph is connected. The three positive values corresponding to elements 1, 3 and 5 of the eigenvector in Eq. (14) state that nodes 1, 3 and 5 belong to one sub-graph. Similarly, the four negative values in the eigenvector place nodes 2, 4, 6 and 7 in the second sub-graph. This example showed that the original flow graph in Fig. 7 was split into two sub-graphs by severing the link between nodes 4 and 5.

The process of sectorization is begun by splitting the initial network graph such as the one in Fig. 5 into two sub-graphs using the spectral bisection method discussed above. The weights of both the sub-graphs are then computed using the technique discussed in Subsection A. The sub-graph with the higher weight is subsequently partitioned into its two sub-graphs. Weights of these sub-graphs are computed. The sub-graph with the largest weight is selected for further partitioning. This process of spectral bisection and weight computation of the sub-graphs is continued until the termination criterion is met.

As bisection proceeds, sub-graphs with a single node can be created. If the weight of that sub-graph, or equivalently the weight of the associated node, is greater than the desired weight, new nodes have to be created to enable further partitioning. Figure 12 illustrates such a situation in which the sub-graph consisting of three nodes $N_1$, $N_2$ and $N_3$ is partitioned into two sub-graphs, the first consisting of only one node $N_1$ and the other consisting of the two remaining nodes $N_2$ and $N_3$. The grid cells associated with Node $N_1$, indicated by the pattern of horizontal lines, define the first region, and the ones associated with nodes $N_2$ and $N_3$ define the second region. To lower the weight of the first region, a node needs to be added.
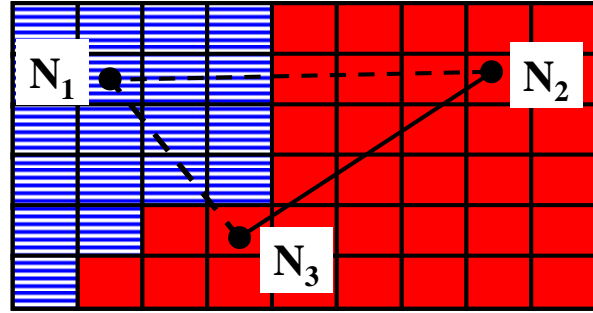


**Figure 12. Two regions obtained after bisection.**

One method for accomplishing this is shown in Fig. 13 in which a node is added in the middle of one of the links that existed between Node $N_1$ and the other nodes in the second region prior to the bisection step. In the example being discussed, Node $N_1$ was connected both to nodes $N_2$ and $N_3$ prior to the bisection step, therefore a node could be inserted in the middle of one of the two links. If a node were inserted in the middle of the link connecting the nodes $N_1$ and $N_2$, it would be outside the first region. Node $N_4$, on the other hand, in the middle of the other link is found to be within the region, therefore this node is added as the new node. If there are several midpoints of the links that are within the region to be partitioned, the midpoint that is farthest from the sole node of the sub-graph is chosen as the location of the new node. For example, if the midpoint of the link connecting Node $N_1$ to Node $N_2$ and the midpoint of the link connecting Node $N_1$ to Node $N_3$ were both within the first region, the one that was farther from Node $N_1$ would be chosen as the location of the new node. Selection of the farther location is driven by the desire to partition the region such that the sub-regions are somewhat equal in size.
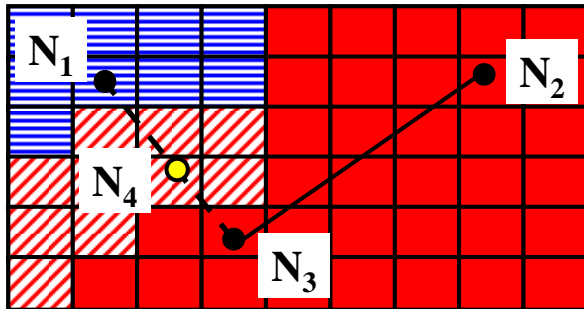


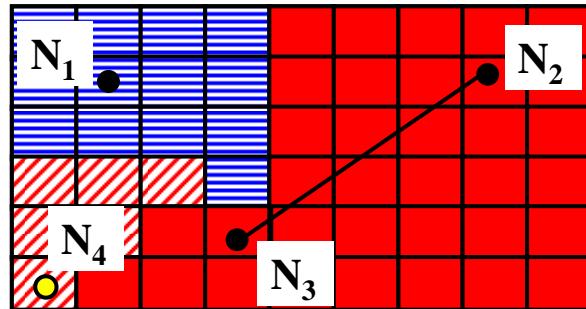**Figure 13. Node addition for splitting the first region.**



**Figure 14. Selecting the farthest grid-cell as location of new node.**

An alternative procedure is to ignore the prior links such as from Node $N_1$ to Node $N_2$ and from Node $N_1$ to Node $N_3$, and select the location of the farthest grid cell in the region as the location of the new node. The concept is illustrated in Fig. 14. This second procedure is used if the previous first procedure fails.

The region growing algorithm, which was described earlier in Subsection C of Section IV, is then used to update the assignment matrix $\tilde{A}$ (see Eq. (7)) to reflect the assignment of grid cells to nodes $N_1$ and $N_4$. The cells assigned to these nodes are shown by horizontal and slanted lines in Figs. 13 and 14. Weights are then recomputed using the updated matrix $\tilde{A}$ and the occupancy matrices $A$. Spectral bisection into sub-graphs can be continued with the updated weights.

Examples of sectorization of the airspace over the continental United States obtained using the setup phase and the run phase of the algorithm discussed in Sections IV and Section V are discussed in the next section.

## VI. Results

This section describes the sectorization results obtained using the dynamic sectorization algorithm, which is outlined in Fig. 1.

Aircraft position data above 18,000 feet altitude for 24-hour period obtained from January 1, 2005 ASDI data, and the high-altitude sectors from November 2003 adaptation data were used for creating the network flow graph using the procedure described in Subsection A of Section IV. The resulting flow graph is shown in Fig. 5. Next, occupancy matrices were computed for each minute using the position data of aircraft above 18,000 feet altitude, derived from the February 10, 2007 ASDI data, and following the method described in Subsection B of Section IV. The final setup step of assigning the 307,211 grid cells to the nodes of the network flow graph for creating the assignment matrix was achieved using the region growing technique described in Subsection C of Section IV.

The sectorization process was started by creating a Laplacian matrix of the entire network flow graph in Fig. 5 and using it in the spectral bisection method, discussed earlier in Subsection B of Section V, to partition this flow graph into its two sub-graphs. Weights, peak traffic-counts, of these two sub-graphs were then computed using the occupancy matrices and assignment matrix following the second procedure described in Subsection A of Section V. Since ASDI data are provided at one-minute intervals, it is possible to create an occupancy matrix for every minute. This implies that it is possible to create a sectorization for every minute of the day. However, changing the sectorization at such a rapid rate is not of any practical value. It is desirable that the same sectorization be kept for a larger duration of time. This paper does not attempt to determine a suitable duration; it assumes an interval of one-hour as a reasonable duration. Sixty occupancy matrices, one for each minute, were processed together with the assignment matrix to compute the weights of the sub-graphs.

These sixty occupancy matrices do not have to be stored at the same time; only one matrix has to be kept in memory at each time. Once the traffic-count for the sub-graphs are computed using the occupancy matrix for that minute, the next minute's occupancy matrix can be created and stored in memory. To obtain the peak traffic-counts of the sub-graphs for the entire hour, a vector of the size of the number of sub-graphs is initialized with the traffic-counts obtained for the first minute. These values are then replaced with the larger of these stored values and the traffic-counts obtained for the next minute. This procedure is then repeated for each subsequent minute of the hour under consideration.

Once the weights, peak traffic-counts, were obtained for each sub-graph, the sub-graph with the largest weight was selected for spectral bisection. The spectral bisection method discussed in Subsection B of Section V was then used for splitting the chosen sub-graph into its two sub-graphs. If splitting resulted in a sub-graph with only a single node, and its weigh was the largest, the node addition procedures discussed in Subsection B of Section V were used to update the occupancy matrix. The steps of computing weights of sub-graphs, selecting the sub-graph with the largest weight and partitioning the selected sub-graph into its two sub-graphs using spectral bisection were repeated till 466 sub-graphs were obtained. The number 466 was chosen because the November 2003 adaptation data had 466 high-altitude sectors.

Sectors were then obtained by assigning common labels to the grid cells, equivalently to the elements of the assignment matrix, associated with the nodes of each sub-graph. For example, if nodes 1, 3 and 5 belong to sub-graph number 1, and nodes 2, 4, 6 and 7 belong to sub-graph number 2, the sectorization matrix- $S$ based on the assignment matrix in Eq. (7) is:

$$S = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 \end{bmatrix} \tag{15}$$

The steps described above were repeated to create sectorization for each hour of the 24-hour period. Figure 15 shows an edge image of the 466 sectors. It was obtained by using the Sobel operator (see Ref. 12) on the sectorization matrix $S$ corresponding to 4 pm Eastern Standard Time (EST) (21:00 Coordinated Universal Time (UTC)). A closer examination of Fig. 15 reveals that there are instances when smaller sectors are completely

enclosed within larger sectors. There are some very small sectors. Some sectors are thin and elongated. These limitations can be overcome by merging the small sectors with the neighboring larger sectors.

Figure 16 shows the plan view slice of the currently used Oakland Center (ZOA) sectorization at 25,000 feet altitude superimposed on the sectors obtained using the algorithm. These sectors are part of the sectorization of the continental United States airspace, such as the one shown in Fig. 15, corresponding to the 6 pm Pacific Standard Time (PST). Note that the sectors created by the algorithm do not lie within the confines of center boundary because it is based on partitioning the airspace over the United States and not just within Oakland Center airspace. It should also be observed that the sector boundaries of the created sectors do not line up
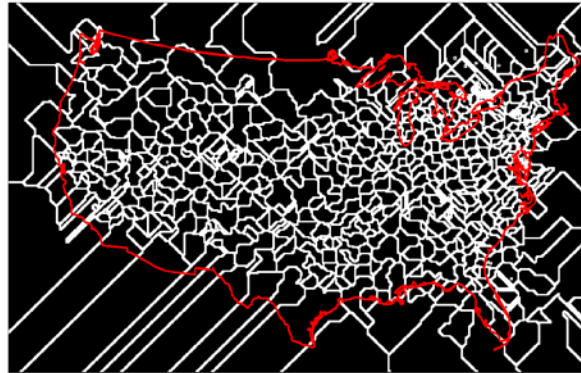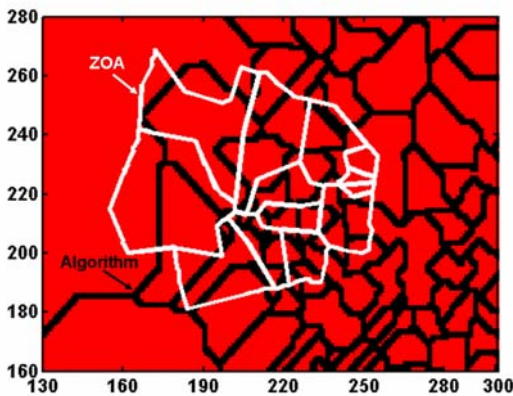


**Figure 15. Edge image of the sectorization.**



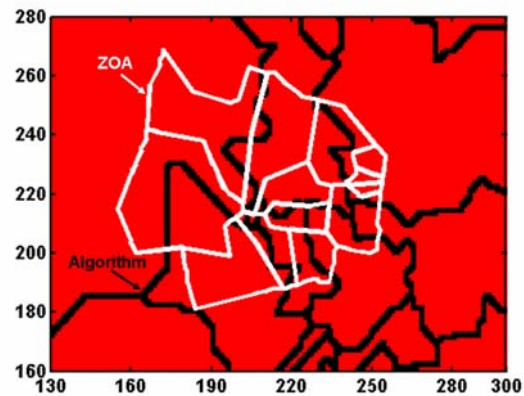**Figure 16. Oakland Center sectorization for 6 pm Pacific Standard Time (2:00 UTC).**



**Figure 17. Oakland Center sectorization for 4 am Pacific Standard Time (12:00 UTC).**

with the sector boundaries of the Oakland Center. Since the grid cell assignment process associates regions surrounding the nodes with nodes of the flow graph, the nodes end up inside the sectors and not on the boundary of sectors created by the spectral bisection process.

Figure 17 shows the sectorization of the United States airspace around Oakland Center airspace based on the air traffic at 4 am PST. Comparing Fig. 16 to Fig. 17 it is seen that the algorithm partitions the airspace into many more regions for traffic at 6 pm compared to for traffic at 4 am. Figure 18 shows the actual traffic-count in the Oakland Center for February 10, 2007 obtained by processing ASDI data. This figure shows that peak traffic was around 6 pm PST (2:00 UTC) and the least traffic was around 4 am PST (12:00 UTC).
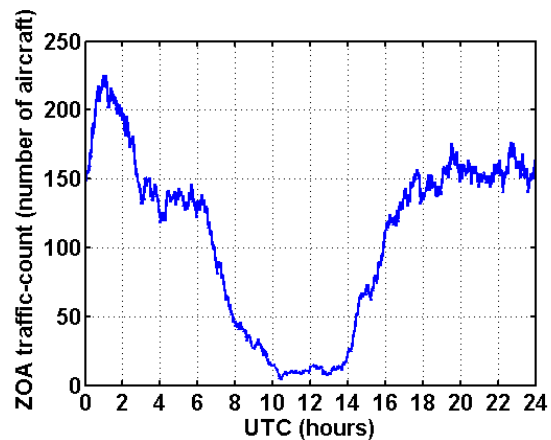
Peak traffic-count statistics for sectorization of the United States airspace at each hour are summarized in Table 1. Recollect that sectorization was changed only once every hour. The second column of the table shows the maximum of the peak traffic-counts in the sectors.



**Figure 18. Oakland Center traffic-counts for 24 hour period.**

Subsequent columns present the number of sectors that had peak traffic-counts within the ranges specified in the

Table 1. Peak traffic-count summary.

| UTC | Max. P | $0 \leq P < 5$ | $5 \leq P < 10$ | $10 \leq P < 15$ | $15 \leq P < 20$ | $20 \leq P$ |
|---|---|---|---|---|---|---|
| 1 | 22 | 24 | 84 | 124 | 184 | 50 |
| 2 | 20 | 36 | 94 | 176 | 160 | 0 |
| 3 | 16 | 38 | 165 | 249 | 14 | 0 |
| 4 | 14 | 53 | 209 | 204 | 0 | 0 |
| 5 | 11 | 93 | 309 | 64 | 0 | 0 |
| 6 | 9 | 175 | 291 | 0 | 0 | 0 |
| 7 | 6 | 396 | 70 | 0 | 0 | 0 |
| 8 | 5 | 466 | 0 | 0 | 0 | 0 |
| 9 | 5 | 466 | 0 | 0 | 0 | 0 |
| 10 | 5 | 466 | 0 | 0 | 0 | 0 |
| 11 | 7 | 164 | 133 | 0 | 0 | 0 |
| 12 | 7 | 257 | 209 | 0 | 0 | 0 |
| 13 | 10 | 145 | 321 | 0 | 0 | 0 |
| 14 | 13 | 94 | 255 | 117 | 0 | 0 |
| 15 | 18 | 118 | 105 | 172 | 71 | 0 |
| 16 | 18 | 39 | 141 | 204 | 82 | 0 |
| 17 | 19 | 32 | 115 | 197 | 122 | 0 |
| 18 | 18 | 25 | 119 | 194 | 128 | 0 |
| 19 | 19 | 30 | 119 | 194 | 123 | 0 |
| 20 | 18 | 34 | 120 | 199 | 113 | 0 |
| 21 | 18 | 39 | 120 | 215 | 92 | 0 |
| 22 | 17 | 35 | 143 | 219 | 69 | 0 |
| 23 | 18 | 40 | 119 | 207 | 100 | 0 |

table header. For example, the value in the first row and the sixth column states that 184 sectors had peak traffic-counts P greater than or equal to 15 aircraft and less than 20 aircraft during the zero UTC to one UTC time period. The range of peak traffic-count values between 10 and 22 during the busier periods of the day agree with the Monitor Alert Parameter (MAP) values for the sectors. MAP values are used in the ETMS to trigger traffic flow management alerts when forecast traffic demand in the sectors exceeds these values.

The distribution of the MAP values derived from April 2007 ETMS is shown in Fig. 19. The heights of the stems in Fig. 19 indicate the percentage of sectors with the particular MAP value marked on the abscissa. The minimum, mean, median, mode and maximum MAP values were found to be 9, 16, 16, 18 and 25.

Results presented in Figs. 15 through 17 and Table 1 show that the weighted-graph technique described in this paper is successful in partitioning the airspace into smaller regions based on the peak traffic-count criterion. Peak traffic-counts in the sectors lie within the range of MAP values used in ETMS. Along with these desirable features of sectorization, results have also revealed the need for further improvement. Some of the undesirable attributes like small or elongated sectors were discussed earlier. Sector shapes were found to be quite complicated. It might be possible to simplify the shapes by using convex polygons to discretize the airspace. Since an efficient method for locating aircraft in hexagonal cells is available, described in Ref. 3, the benefit of using hexagonal grid cells instead of the square cells should be explored for obtaining convex regions. Finally, additional workload metrics need to be considered for guiding the sectorization process.

Sectorization results, of the type shown in Fig. 15, showed that smaller sectors are created close to the West Coast for hours of the day when the ratio of the West Coast traffic to the East Coast traffic is higher. This artifact is
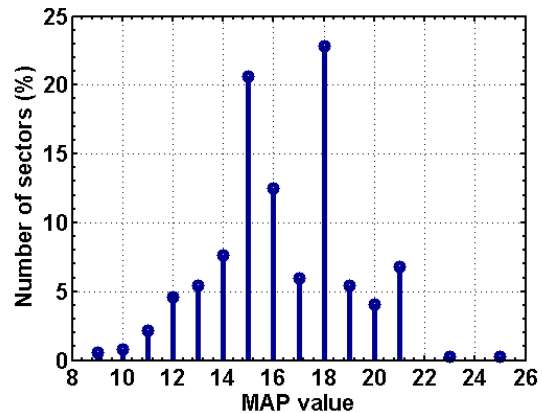


**Figure 19. Distribution of MAP values.**

due to fixed number of sectors. This limitation can be mitigated by using desired peak traffic-count rather than fixed number of sectors as the termination criteria.

## VII. Conclusions

A traffic dependent flow-graph based algorithm for sectorizing the airspace was described. The algorithm was used for partitioning the airspace over the United States into 466 sectors, once for each hour of the 24-hour day, using a single day of air traffic data. These results showed that the peak traffic-counts in the sectors were consistent with the Monitor Alert Parameters specified in the Enhanced Traffic Management System. Examples were presented to illustrate that smaller sectors are created in regions of heavy traffic and larger ones in regions of light traffic. Along with these desirable features, some limitations and opportunities for refinement of the sectorization algorithm were revealed by these examples. The algorithm generated small sectors that were enclosed within larger sectors. Elongated sectors were created in some instances. Some of the areas of possible refinement are: use of hexagonal cells rather than square cells for synthesis of convex sectors, use of peak traffic-counts rather than the fixed number of sectors as the termination criteria, and use of alternative flow graph structures such as a spanning tree connecting the major airports in the United States.

## Acknowledgements

## References

[1]National Aeronautics and Space Administration, "Next Generation Air transportation System (NGATS) Air Traffic Management (ATM)-Airspace Project," version 5.0, 24 May 2006, pp. 31-34.

[2]Yousefi, A., and Donohue, G. L., "Temporal and Spatial Distribution of Airspace Complexity for Air Traffic Controller Workload-Based Sectorization," AIAA Paper 2004-6455, *AIAA 4th Aviation Technology, Integration and Operations (ATIO) Forum*, Chicago, IL, 20-22 September 2004.

[3]Klein, A., "An Efficient Method for Airspace Analysis and Partitioning Based on Equalized Traffic Mass," *6th FAA and EUROCONTROL ATM Conference*, Baltimore, MD, June 2005.

[4]Delahaye, D., Alliot, J.-M., Schoenauer, M., and Farges, J.-L., "Genetic Algorithms for Partitioning Air Space," *10th IEEE Conference on Artificial Intelligence for Applications*, San Antonio, TX, 1-4 March 1994, pp. 291-297.

[5]Basu, A., Mitchell, J. S. B., and Sabhnani, G., "Geometric Algorithms for Optimal Airspace Design and Air Traffic Controller Workload Balancing," *16th Fall Workshop on Computational and Combinatorial Geometry*, Northampton, MA, 10-11 November 2006.

[6]Delahaye, D., Schoenauer, M., and Alliot, J.-M., "Airspace Sectoring by Evolutionary Computation," *IEEE International Conference on Computational Intelligence*, Anchorage, AK, 4-9 May 1998, pp. 218-223.

[7]Trandac, H., Baptiste, P., and Duong, V., "Optimized Sectorization of Airspace with Constraints," *5th Eurocontrol / FAA ATM R&D Seminar*, Budapest, Hungary, 23-27 June 2003.

[8]Kernighan, B., and Lin, S., "An Efficient Heuristic Procedure for Partitioning Graphs," *Bell System Technical Journal*, Vol. 49, February 1970, pp. 291-307.

[9]Paielli, R. A., and Erzberger, H., "Tactical Conflict Detection Methods for Reducing Operational Errors," *Air Traffic Control Quarterly*, Vol. 13, No. 1, 2005.

[10]Erzberger, H., "Automated Conflict Resolution for Air Traffic Control," *25th Congress of International Council of the Aeronautical Sciences*, Hamburg, Germany, 3-8 September, 2006.

[11]Duda, R. O., and Hart, P. E., "Pattern Classification and Scene Analysis," John Wiley and Sons, NY, 1973, pp. 233.

[12]Ballard, D. H., and Brown, C. M., "Computer Vision," Prentice-Hall, Englewood Cliffs, NJ, 1982, pp. 143-146, pp. 123-131, pp. 76-77.

[13]Robelin, C.-A., Dengfeng, S., Guoyuan, W., and Bayen, A., "MILP Control of Aggregate Eulerian Network Airspace Models," *American Control Conference*, June 2006, pp. 5257-5262.

[14]Graham, R. E., Knuth, D. E., and Patashnik, O., "Concrete Mathematics: A Foundation for Computer Science," Addison-Wesley, Reading, MA, 1994.

[15]Simon, H. D., "Partitioning of Unstructured Problems for Parallel Processing," Report RNR-91-008, Applied Research Branch, NAS Systems Division, NASA Ames Research Center, Moffett Field, CA 94035, February, 1991.