ANALYSIS OF AN OPTIMAL SECTOR DESIGN METHOD

Michael Drew, University of California Santa Cruz, Moffett Field, CA 94035-1000, USA

Abstract

An existing Mixed Integer Programming optimal sector design method is implemented, analyzed, and improved. The original model is a powerful and convenient method of designing sectors, but frequently produces geometrically undesirable sector shapes. Also, solutions are sensitive to external parameters not related to the flight data that the solutions are based upon. Near-optimal solutions vary drastically with small changes in objective function value. The model is then altered to reduce this sensitivity and produce sector designs with a more favorable geometry. Additionally, a boundary smoothing method is applied that eliminates jagged boundary edges and produces a more realistic and feasible sector geometry.

I. Introduction

The current U.S. National Airspace System (NAS) is congested, and estimates suggest that traffic volume will increase by 2 to 3 times in the coming decades [1]. One of the options being considered to address this problem and increase throughput while decreasing delay is to redesign the current airspace, which currently consists of 20 Air Route Traffic Control Centers (ARTCC) across the continental U.S. that are each subdivided into sectors. Today's sectors have evolved over the years in an ad hoc fashion according to current flight patterns and volume. Minor changes are occasionally made to accommodate variations in the airspace demand. While the current sector designs are adequate for today's traffic, the existing airborne congestion and predicted future demand is motivation for improving sector capacity using scientific methods. Thus, methods of analytically redesigning sectors according to certain geometric and flight pattern constraints are sought in order to improve the efficiency of the entire NAS.

Many of the current sector design methods are either heuristic-based or optimization-based (e.g., Linear Programming (LP) or Mixed Linear Programming (MIP) methods). One advantage of heuristic methods is their ability to apply several complex design criteria based on sector geometry as well as air traffic flow patterns. In [2] and [3] the authors apply computational geometry techniques to recursively partition a 2D airspace region. They investigate different types of partition cutting methods to maintain equitable workload balance within the region. One attractive feature of this method is that workload metrics are not required to be additive as they are in current LP and MIP models. They are also able to partition the space with controls on the resulting sector convexity and aspect ratio. In [4] Xue uses Genetic Algorithms (GA) to select the location of Voronoi partitions within the region. Here, GA is used to judge the resulting partitions on workload estimates and dominant flow directions to increase the region's total sector capacity. The use of GA simplifies the algorithm design while maintaining the ability to use any number of complex sector evaluation criteria. In [5] a method of optimal ARTCC design is presented while in [6] a MIP model for sector design is described. This model is capable of forming sectors that are aligned with major traffic flows while keeping the workload within the center balanced among the sectors. It is a compelling method of sector design both for its ease of implementation and promise of an “optimal” solution. Yet, in practice, there are some behavioral inconsistencies in the model that are not well understood, and sometimes it produces poor sector designs that would not be feasible without additional processing. A deeper understanding of the MIP model is required before it can be put into practice.

This paper focuses on the implementation, analysis, and improvement of the MIP sector design method introduced in [6]. Section II provides the details of the MIP model and Section III describes the implementation of the model within a simulation framework. Section IV presents some of the resulting sector designs along with an analysis of the benefits and shortcomings of the model. This is the primary thrust of the paper, since it focuses on the major problems with the MIP model performance. Section V presents the improvements made to the original model that address these issues. Finally, Section VI concludes with an overall assessment of this sector design method and its potential for future use.

II. The Optimization Model

The basis of the MIP model presented here is the discretization of the airspace into hexagonal cells. As simulated or historic flight data traverse the airspace, both a workload metric w_i and connectivity metric c_{ij} are recorded for each cell i. The workload metric can be the number of aircraft track counts, or a more complex metric like Dynamic Density (DD). However it is not clear how a workload metric like DD should be discretized in sub-sector increments since the MIP model structure currently restricts the workload value to be an additive quantity. The connectivity metric c_{ij} is the total number of planes that travel from cell i into a neighboring cell j.

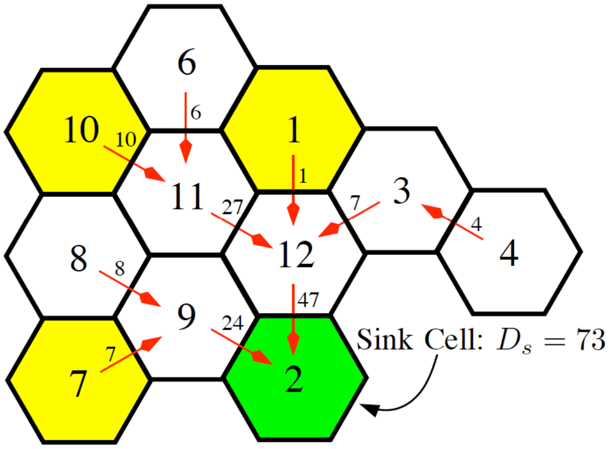


Figure 1. Cell workload flow paths

Referring to Figure 1, the model works on the abstract quantity of workload *flow* (red arrows) given by the decision variables f_{ij} in the model. Flow enters each cell from at least one of its neighbors then exits into exactly one neighboring cell. The workload of that cell is added to the flow, and certain cells called *seeds* (yellow cells) have the option of becoming *sinks* (green cells) that absorb the flow. The number of sinks is constrained to be equal to the number of desired sectors, which is known *a priori*. In this way a sector becomes the contiguous cluster of cells that feed into one sink, and the total workload of the sector is the amount of flow absorbed by that sink. This is topologically a network model, and although workload flow is an abstract quantity, solving for flow paths between cells allows the creation of sectors that are aligned according to dominant aircraft trajectories while maintaining an equitable distribution of workload among all the sectors. A major advantage of this method is the implicit contiguity of all the cells within a sector. Depending on the objective function and constraints, the workload distribution among the sectors can be controlled, and so can the dominant orientation of the resulting cell cluster.

Model Specifics

Yousefi's MIP optimization sector design method as adapted from [6] may be formally defined as follows:

$$\text{minimize} \sum_{i\in\mathbf{I}, j\in\mathbf{J}} \frac{f_{ij}}{c_{ij}} \tag{1}$$

subject to the following constraints:

$$\forall i \in \mathbf{I} : \sum_j f_{ij} - \sum_j f_{N_{ij},\, P_j} = w_i - D_i, \tag{2}$$

$$\forall i \in \mathbf{I} | i \notin \mathbf{S} : D_i = 0, \tag{3}$$

$$\forall s \in \mathbf{S} : L\mu \cdot y_s \le D_s \le U\mu \cdot y_s, \tag{4}$$

$$\sum_{s\in\mathbf{S}} y_s = \text{number of sectors} \tag{5}$$

$$\forall i \in \mathbf{I},\ | \ i \notin \mathbf{S} : \sum_{j\in\mathbf{J}} u_{ij} = 1 \tag{6}$$

$$\forall s \in \mathbf{S} : \sum_{j\in\mathbf{J}} u_{sj} + y_s = 1 \tag{7}$$

$$\forall i \in \mathbf{I},\ \ j \in \mathbf{J} : \frac{f_{ij}}{M} \le u_{ij}. \tag{8}$$

Here f_{ij} , D_i, y_s, and u_{ij} are decision variables with f_{ij}, $D_i \in \Re^+$, and y_s, $u_{ij} \in \{0, 1\}$. f_{ij} represents the workload flow from cell i into cell j . D_i is the workload demand of cell i . y_s is an assignment variable that determines when seed s is selected to be a sink. u_{ij} is the assignment variable controlling the direction of flow exiting cell i .

For n hexagonal cells, $\mathbf{I} = \{1, \dots, n\}$ is the set of all cell indices, and $\mathbf{J} = \{1, \dots, 6\}$ represents a cell's six adjacent neighbors starting with the southwestern neighbor proceeding counter-clockwise around the cell. $\mathbf{S} \subseteq \mathbf{I}$ is the set of all seed cell indices. Seed locations are determined ahead of time and are interspersed throughout the grid as evenly as possible.

Thus, a sector consists of all the cells whose flows converge to one sink, and by minimizing equation 1, the clusters will tend to be oriented along the dominant traffic flows. Constraint 2 maintains the conservation of workload flow from one cell to the next. In words, flow-out minus flow-in equals the workload addition of the cell minus the cell demand (which is zero for all non-seed cells according to constraint 3). This constraint sets up a network model between the cells, which forces cell contiguity within the resulting sector design. Here N_{ij} is an $n \times 6$ look-up table that returns the cell index of cell i 's j-th neighbor. $P_j = [4, 5, 6, 1, 2, 3]$ is another look-up table that returns the neighbor index of cell i relative to its neighboring cell j. The use of these look-up tables reduces the size of the f_{ij} matrix from what would otherwise be a sparsely populated $n \times n$ sized matrix to a more compact $n \times 6$ matrix.

In constraint 4, $y_s = 1$ if seed cell s becomes a sink, and the sector demand D_s is bounded by the upper and lower bounds $U\mu$ and $L\mu$. If $y_s = 0$, seed cell s is not a sink and $D_s = 0$, causing cell s to behave like any other source cell. This constraint is what distributes the resulting sector workload in an equitable fashion. By tuning L and U with $L = (0, 1.0]$, and $U = [1.0, \infty)$ the resulting workload variation among the sectors in the center can be controlled.

There must be exactly one flow path out of each non-sink cell; otherwise, the flow would be divided between more than one sink (sector) leading to erroneous sector workload values and improper workload distribution. This is enforced by the u_{ij} decision variables and the last 3 constraints. Constraints 6 and 7 force $\sum_j u_{ij}$ to be 1 for all non-sink cells and 0 for those cells selected to be sinks. In conjunction with constraint 8, setting $M > U\mu$, forces all non-sink cells to have exactly one out-bound flow path and all sink cells to have no out-bound paths.

This MIP model is attractive for many reasons. For one, it is linear and can be solved in a reasonable amount of time depending on the number of cells, seeds, sinks (sectors), and tightness of bounds (U and L). Even flow constraints 6 - 8, which dramatically increase computation time, do not make the model prohibitively unwieldy to solve. Furthermore, as previously mentioned, because it is a simple network flow model, contiguity of cells within a sector is an implicit constraint. This is not trivial to enforce in other cell-based clustering optimization models.

There are some drawbacks to this model, perhaps the greatest of which is the fact that it has no inherent notion of a sector. That is, there is no cell-sector assignment variable that could be used to enforce additional geometric constraints. It is possible to add this variable with additional constraints, but this will drastically increase solution times. Geometric constraints like convexity and aspect ratio could be a useful addition to the model since, as the results in the next section demonstrate, it frequently produces geometrically undesirable sector shapes. Also, the shapes tend to be sensitive to model parameters and can vary dramatically from one near-optimal solution to the next. Fortunately, there are some simple methods to address these issues that will be presented herein.

III. Model Implementation

The Future ATM Concepts Evaluation Tool (FACET) is a software package developed, maintained, and used by NASA Ames Research Center. [7] It is capable of analyzing both historic and current air traffic and weather data across the entire NAS. It is commonly used as a research tool to evaluate “what if” scenarios of various strategic and tactical air traffic and air space design concepts.

A hex grid application was added to FACET, which records aircraft counts (w_i) and connectivity data (c_{ij}) for each cell. Any historical data can be run through the grid in either “playback” or “simulation” mode. Playback mode uses actual recorded radar locations of the aircraft, whereas simulation mode uses either the filed flight plan or great circle paths for aircraft locations. Flight plan simulations were used for the results presented herein.

Implementation data from certain times of day or from different weather conditions can be used as a basis for sector designs optimized (as defined by the model) for those conditions. Hex grids can be created of any size and at any altitude range, and can be placed over the entire NAS or over individual centers. Care must be taken, however, to consider the size of the cell verses the sampling rate of the data files. Most historic flight data are recorded at 60 second sampling intervals. Thus, hex cells smaller than about 8 nautical miles in height can be missed entirely during a simulation. This is addressed by interpolating flight data between samples, thereby up-sampling the data prior to use in FACET. Of course, flights passing through a cell corner may still be missed by that cell, but this is not a problem because there will be sample points in neighboring cells that are adjacent to each other, so the connectivity data from the flight will remain contiguous. In aggregate, these effects tend to become negligible.

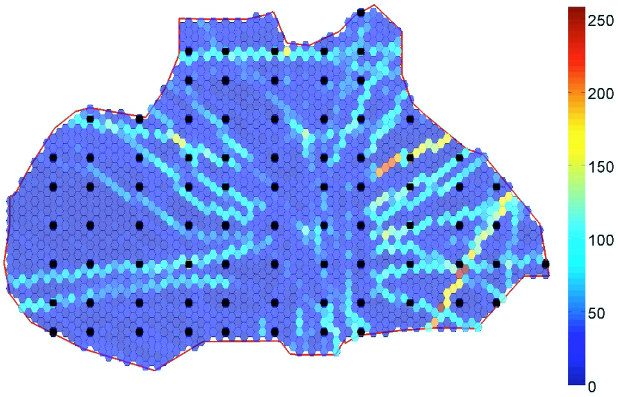


Figure 2. Hex grid aircraft counts from FACET

After running a simulation in FACET, the results are processed by Matlab to convert the data into a form usable by the optimization solver. The grid cell output is shown in Figure 2 as plotted by Matlab. Here, 2205 hex cells 8 nmi high are used to record 24 hours of flight data from April 21, 2005 over ZFW center at flight level 240 – 350. This figure shows the aircraft counts of each cell along with 82 seed cells shown in black. Next, Ilog's OPL Studio or AMPL CPLEX solves the optimization problem described in the previous section. Because the solution to the optimization problem consists of only work flow values and sink demands (f_{ij}, D_s), it must be post-processed to convert it into separate clusters of cells representing sectors. This procedure is diagrammed in Figure 3. Once the sector boundaries are determined, they can be imported back into FACET (or other Air Traffic Management (ATM) simulation tools) for experimental validation and analysis.

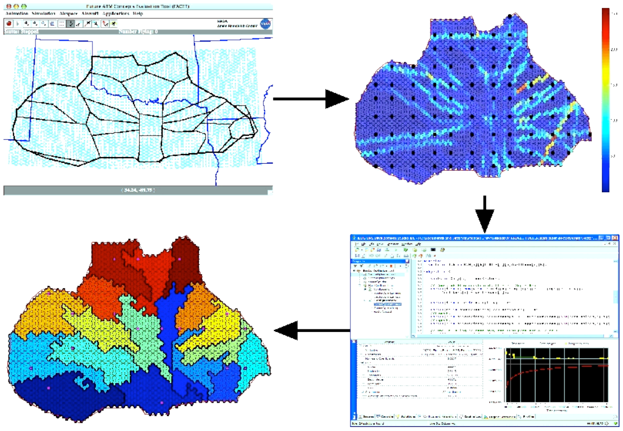


Figure 3. Sector design procedure

IV. Model Results and Analysis

One of the nice features of the MIP model is the ability to capture the dominant flow directions in the shape of the sector while keeping the workload balanced. Figure 4 shows the sector design produced by the MIP model based on the data shown in Figure 2. Note that the dominant direction of most of the sectors is aligned with the dominant trajectories depicted in Figure 2. Sectors that are aligned with their dominant flows are desirable, because the average aircraft dwell time will be higher and likewise the capacity of the sector.

As is obvious from Figure 4, many of the sector shapes are extremely non-convex and have rough geometric features (beyond those produced by the jagged hex cell edges alone) not desirable for sector boundaries. When implementing this model, experimentation revealed additional issues that will now be discussed in detail.

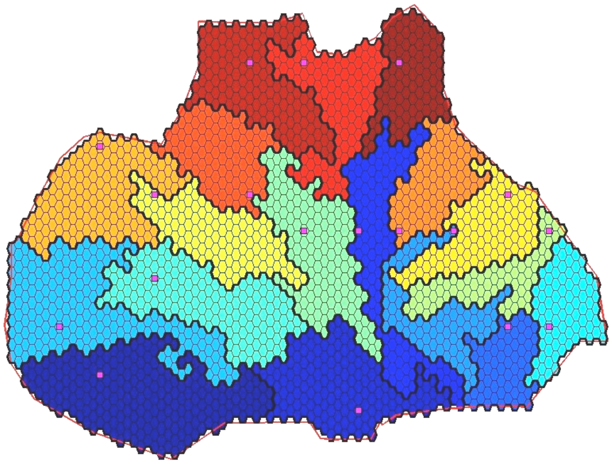
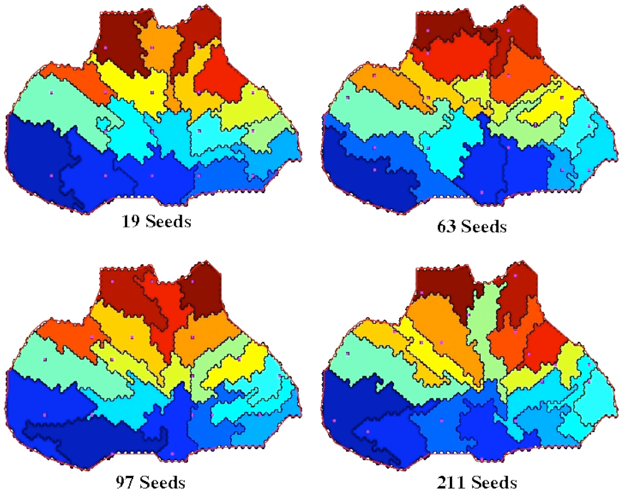


Figure 4. Optimal sector design results

Seed Sensitivity

The number and location of the seed cells can dramatically affect sector design results. Recall that seeds are cells chosen to be candidate sinks that absorb the abstract quantity known as *flow*. Thus, for each center there must be at least as many seeds as there are sectors. Sinks have no inherent meaning in the real world and are only used as a device in the network model for collecting and measuring the total workload present in a sector. It seems reasonable that as long as a lot of seeds are evenly dispersed throughout the center, good solutions should be found. In practice, using 3 - 5 times the number of sinks as there are sectors tends to produce desired results that solve in reasonable time. Yet, the more options the solver has for potential sinks, the lower the best integer bound of the MIP problem goes, which results in a more optimal solution according to equation 1. Unfortunately, allowing all cells to be seeds (potential sinks) makes the problem too complex to solve consistently in tolerable time.

Consider the results of Figure 5, which shows four sector designs based on identical track data using an increasing number of seeds. Some of the sector boundaries remain relatively stable (particularly the western side of the center between the 97 and 211 seeds examples), but many sectors change dramatically as the number of seeds is increased. Note, also, that the sector designs with the largest number of seeds are not necessarily any more geometrically desirable (in terms of convexity, aspect ratio, etc.) than those with fewer seeds—despite producing objective function solutions with lower values. This really comes as no surprise, since there are no geometric functions or constraints within this MIP model.



19 Seeds

63 Seeds

97 Seeds

211 Seeds

Figure 5. Sector designs with increasing number of seeds

Solution Space Sensitivity

Another concern is the variation of the resulting sector designs within the feasible solution space itself. As a MIP model is solved, the objective function value at the current best integer feasible solution is compared with that of the best known lower node bound. This difference is known in CPLEX as the *mip gap*, and often the solver is permitted to terminate once this value drops below a pre-determined limit to prevent excessively long run times. For this model solutions with a mip gap of less than 2% were considered satisfactory. Experience has shown that widely varying sector geometries can result from minor differences in the objective function value—even as the mip gap approaches zero. This solution instability among near-optimal solutions may be more or less drastic depending on the track data, but it is always present to some extent.

Figure 6 shows the relative normalized objective function values of several solutions along with a shape change metric. This is a simple method of quantifying the geometric difference between one sector design and another, and it is based on the relative change in cell-sector assignments. To compare two sector design solutions A and B, the shape change metric *SC* is defined as follows:

$$SC_{\text{A:B}} = \sum_{k}^{K} \frac{n_k - \alpha_k}{n_k} \quad (9)$$

Here K is the number of sectors, and n_k is the number of cells assigned to sector k in solution A. α_k is the number of cells assigned to sector k in solution A that are assigned to sector j in solution B where j is the dominant cell assignment of cells from solution A sector k in solution B.

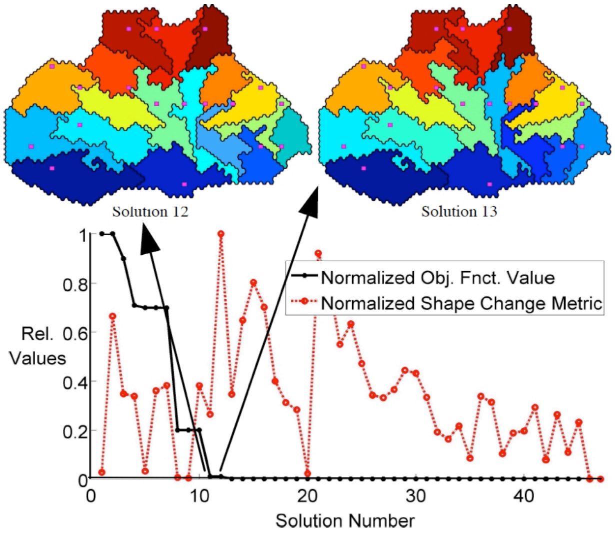


Figure 6. Difference in near-optimal solutions

Figure 6 demonstrates that as the objective function is minimized, there are still major geometric differences between subsequent solutions—even as the solution becomes close to optimal. Notice that at solution 12, there is a spike in the shape change metric even though the objective function values of solutions 12 and 13 are within 0.04% of each other. The sector boundaries for these solutions are shown above the plot. Note that several of the sectors in the east and northwest regions of the center show drastic differences. The fact that the changes in sector geometry are not strongly correlated to changes in the objective function value is especially obvious here.

Considering the objective function of equation 1 this effect is not surprising. It is apparent that small changes in the routing of flow f_{ij} can produce drastic differences in the resulting sector geometry. Conversely, there is no unique mapping from a particular sector design geometry to a feasible set of decision variable solutions in the MIP model. While most of the feasible solutions near the final optimal solution appear relatively stable, major geometric changes can and do occur between solutions. In the next section, methods of improving these results will be discussed.

V. MIP Model Improvements

It is clear that this sector design method has some issues in its original form. However, the promise of an elegant model that provides a reliable and tractable method of producing an optimal design is motivation for investigating ways of improving this model. Here, methods of improving the stability of the solutions and the geometry of the final designs are presented.

Symmetric Connectivity

The rough and extremely non-convex sectors of Figure 4 can partially be explained by considering that the optimization model will seek solutions that direct the abstract flow quantity along paths that minimize equation 1. Recalling that since c_{ij} is the record of planes that flew from cell i to the neighboring cell j , if more planes flew from cell i to cell j than from cell j to i , the flow will be biased to go the same direction. Considering this bias, it is apparent that convoluted flow paths may arise leading to rough sector edges containing unwieldy spurs as shown in Figure 4. Since what is desired, however, is that cell i connect to cell j, it is ultimately irrelevant whether the flow goes from i to j or from j to i. Therefore, prior to running the optimization model, c_{ij} is redefined to be equal to the number of planes that flew from cell i to cell j plus the number of planes that flew from cell j to cell i. In this way, the c_{ij} data across a cell edge will be equal on both sides, and flow will be likely to go in either direction giving the optimization software more options to find a feasible solution with a lower objective function value.

This method is referred to as the "symmetric connectivity method" and is depicted in Figures 7 and 8. In Figure 7 a simple example of a grid segment is shown using the standard data structure and resulting sector shape. Figure 8 demonstrates the effects of applying the symmetric connectivity method which produces smoother sector shapes that better correspond to the dominant flows.

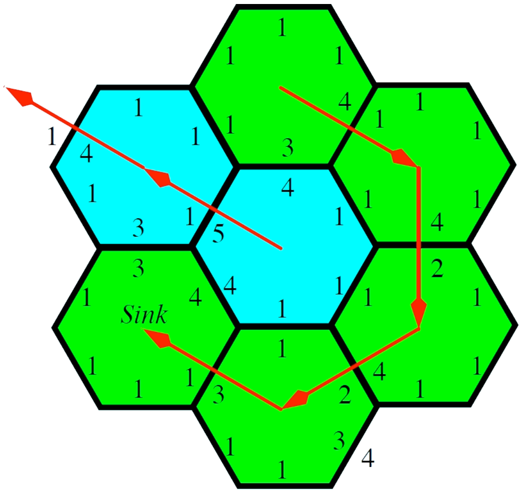


Figure 7. Example grid sector using standard connectivity

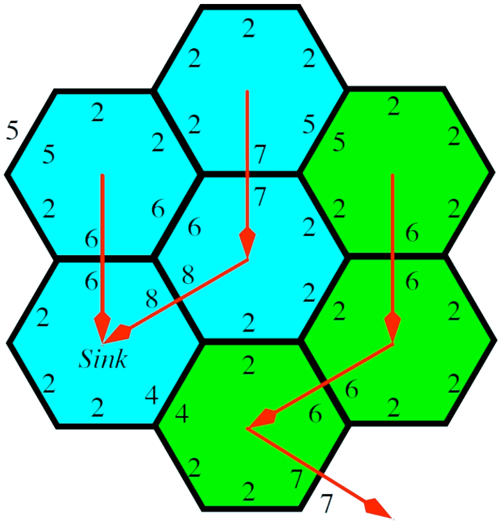


Figure 8. Identical grid using symmetric connectivity

The results of running the same data set through the optimization model using this method is shown in Figure 9. While there are still some undesirable shapes, this is a significant improvement over the original method's results in terms of sector boundary geometry. Furthermore, this minor change reduces the model's seed sensitivity and stabilizes the solution space results as shown in Figure 10. Compared to Figure 6, the symmetric connectivity method exhibits a stronger correlation between shape stability and objective function. Near-optimal solutions tend to be more similar to each other making the model less sensitive to the mip gap stopping criteria.

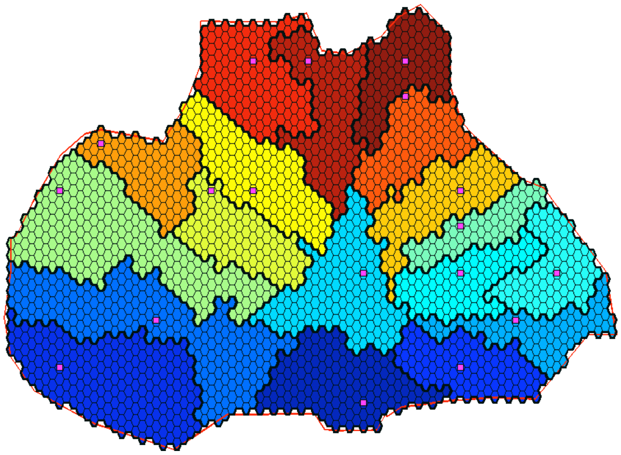


Figure 9. Sector design using symm. conn. method

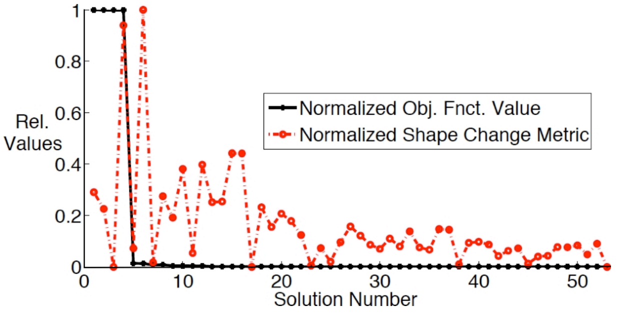


Figure 10. Difference in near-optimal solutions

Convexity Constraints

One of the most unfavorable behaviors of this MIP model is its tendency to produce sector designs with long spurs or thin regions. "Boomerang-shaped" or even "Y-shaped" sectors have been known to arise. These tendencies are reduced by the symmetric connectivity method, but not entirely. If a measure and control of convexity could be enforced the sector shapes could be improved (although non-convexity should probably not be prohibited entirely).

This particular implementation cannot provide a means of controlling the resulting sector geometry, because it does not contain the notion of a sector or of cell-sector assignments. However, methods of extending the model to include a cell-sector assignment variable are available. Yousefi's 3-D sector design method in [6] includes this extension. Also, the author has developed a model that includes some geometric convexity control, but it remains to be seen if it can be applied to representative problems. The additional complexity of the extended model exponentially increases solution time but efforts are underway to improve its efficiency.

A more practical approach may be to sift through a pool of solutions produced by the solver and select those that meet desired criteria. Using Ampl CPLEX's solution pool feature, improvements in sector shape have been realized by selecting from many solutions those that have sector boundaries that are the most convex—thus eliminating some of the extreme spurs and branches of some sectors. Convexity is measured by counting the number of perimeter cells in each sector. Designs that have the least number of perimeter cells are considered the most convex. So far, improvements have only been minor and may not be worth the additional effort, since the near-optimal solutions using the symmetric connectivity method are similar to each other and already more convex than before.

Boundary Smoothing

One of the basic inconveniences of this sector design method is that hex cells will always produce jagged edges that are undesirable for real sector boundaries. It is therefore reasonable to consider ways of smoothing these boundaries to make them simpler and more manageable. Doing so may also improve some of the geometric inadequacies of the MIP model mentioned above.

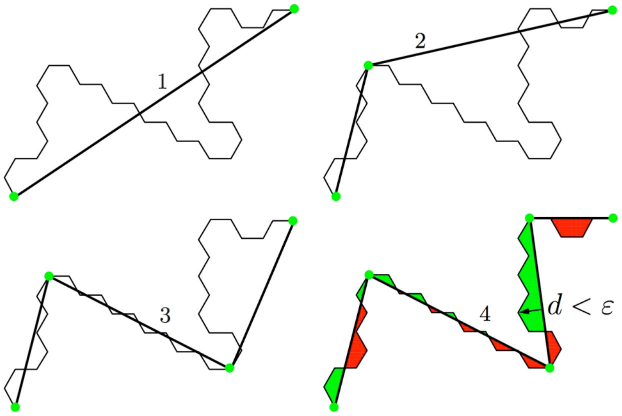


Figure 11. Four iterations of the Douglas-Peucker algorithm

The sector boundaries may be simplified using a common smoothing algorithm from the field of computational geometry known as the Douglas-Peucker (DP) algorithm. [8] This is a recursive algorithm that starts with the first and last vertices of the polyline. Then, the perpendicular distance between each intermediate vertex of the original line and the current simplified polyline segment is measured. The vertex with the largest perpendicular distance is added to the simplified polyline if the distance is greater than some distance tolerance ε. This process repeats as shown in Figure 11 until none of the vertices of the original polyline are greater than ε distance to the segments of the simplified line. The speed of this algorithm is output-dependent running in $O(nm)$ worst case time where m is the number of vertices in the simplified polyline.

Applying this method to the sector design process requires first identifying the shared boundaries of the sectors so they can be smoothed as individual edges. Vertices consisting of the intersection of multiple sector edges are preserved and are not moved during the smoothing process. Once smoothed, the edges are stitched back together to create sector polygons which can be imported into ATM software packages like FACET for further analysis and validation. Figure 12 shows the smoothed version of the sector designs from Figure 9. Values for ε are usually relative to the cell height h. Here a value of $\varepsilon = 3h$ produces good results.

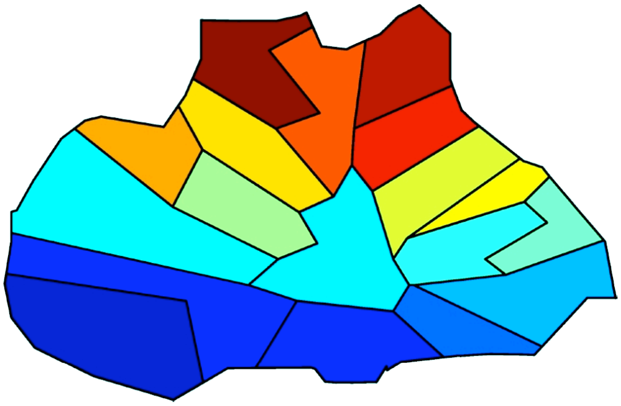


Figure 12. Final smoothed sector design

The results of Figure 12 are promising since many of the resulting sector shapes are fairly reasonable in size and geometry. There are still some obvious problem areas, however, but when applied NAS-wide, the sectors produced by using the symmetric connectivity method in conjunction with this post-optimization smoothing algorithm have shown considerable capacity and performance gains in preliminary ACES (Airspace Concepts Evaluation System [9]) software simulations.

Of course, depending on how ε is tuned, some of the optimality provided by the MIP model may be lost. This is usually not a concern with smaller values of ε. However, if necessary, the loss of optimality in terms of workload balance may be measured. Workload loss (or gain) can be estimated by the fraction of the area of each lost (or gained) cell multiplied by that cell's workload value. The red and green shaded areas of Figure 11 depict this. Although it is not demonstrated here, it is possible to apply the DP algorithm in an incremental fashion to maintain the workload balance within some bound. Such an algorithm may be applied to one edge at a time while keeping track of the change in workload. This is a greedy heuristic with the disadvantage that the last edges to be processed are usually less smooth than the first because there is less available workload deviation in the neighboring sectors.

VI. Conclusions

This paper presents an existing mixed integer programming model that provides an optimal sector design method, but in its implementation may tend to produce some geometrically undesirable sector boundaries. There are also problems that arise from its sensitivity to model parameters and to minor solution space variations within the neighborhood of the optimal objective function value. These issues have been investigated, and the sensitivity and sector geometry problems have been substantially reduced through a modification to the model's input data. Additional improvements were introduced with a post-optimization smoothing algorithm.

There are some remaining disadvantages to this MIP method of sector design including the difficulty in applying geometric constraints and in using more complex workload metrics. It is not clear if any of these can be completely addressed within the MIP model framework. However, with the improvements made herein, this sector design procedure offers a quick method of synthesizing sector designs that can be easily applied to varying traffic patterns and demand. Supplemental heuristics that augment the optimal model results may be developed, but these might ultimately defeat the advantages the optimal design method provides.

Acknowledgements

The author wishes to acknowledge Arash Yousefi and Robert Hoffman of Metron Aviation for their contribution, collaboration, and support of this work.

Email Addresses

michael.c.drew@nasa.gov

References

[1] “Concept of Operations for the Next Generation Air Transportation System,” Version 2.0, Tech. rep., Joint Planning and Development Office, July 24, 2006.

[2] Mitchell, J.S.B., G. Sabhnani, J. Krozel, R. Hoffman, and A. Yousefi, “Dynamic Airspace Configuration Management Based on Computational Geometry Techniques,” *Proc. of AIAA Guidance, Navigation, and Control Conference and Exhibit*, Honolulu, Hawaii, 18-20 August 2008.

[3] Bau, A., Mitchell, J. S. B., and Sabhnani, G., “Geometric Algorithms for Optimal Airspace Design and Air Traffic Controller Workload Balancing,” *Proc. of Workshop on Algorithm Engineering and Experiments*, San Francisco, CA, January 2008.

[4] Xue, M., “Airspace Sector Redesign Based on Voronoi Diagrams,” *Proc. of AIAA Guidance, Navigation, and Control Conference and Exhibit*, Honolulu, Hawaii, 18-21 August 2008.

[5] Yousefi, A., “Optimum Airspace Design with Air Traffic Controller Workload-based Partitioning,” Ph.D. dissertation, George Mason University, 2005.

[6] Yousefi, A., B. Khorrami, R. Hoffman, and B. Hackney, “Enhanced Dynamic Airspace Configuration Algorithms and Concepts,” Metron

Aviation Inc., Tech. Rep. Report No. 34N1207-001-R0, December 2007.

[7] Bilimoria, K., B. Sridhar, G. Chatterji, K.S. Sheth, and S. Grabbe, "FACET: Future ATM Concepts Evaluation Tool," *Air Traffic Control Quarterly*, vol. 9, no.1, pp. 1—20, 2001.

[8] "Ramer Douglas Peucker algorithm," http://en.wikipedia.org/wiki/Ramer-Douglas-Peucker_algorithm.

[9] Meyn, L., R. Windhorst, K. Roth, D. Van Drei, G. Kubat, V. Manikonda, S. Roney, G. Hunter, and G. Couluris, "Build 4 of the Airspace Concept Evaluation System," *Proc. of AIAA Modeling and Simulation Technologies Conference*, Keystone, CO, August 21-24 2006.

27th Digital Avionics Systems Conference
October 26-30, 2008