KEYWORDS ▪ Dependency Structure Matrix ▪ Dynamic Sectorization ▪ Workload ▪ Neighboring Sectors ▪ Clustering

APPLICATION OF DEPENDENCY STRUCTURE MATRIX to Airspace Sectorization and Improving the Distribution of the WORKLOAD AMONG CONTROLLERS

ABSTRACT

This work investigates the application of Dependency Structure Matrix (DSM) to problems with fast dynamics, such as airspace sectorization. The aim of this paper is to use a powerful mathematical tool to distinguish relevant agents in a busy airspace with a logical and meaningful distribution of the workload among air traffic controllers. This approach will help prevent controllers from getting exhausted in busy airspaces and increase the overall capacity of the airspace. It could also serve as a logical interface to simulate the chance of human error in a controlled aerodrome. Different case-studies by the authors reveal that adding real-time capabilities to the existing platform can be very effective with the re-architecting of neighboring sectors so that the domino effects of delays imposed on neighbors is decreased.

Mahsa Farsad
Center for Precision Metrology, University of North Carolina, Charlotte, North Carolina, USA
mfarsad@uncc.edu

Seyed Mohammad-Bagher Malaek
Department of Aerospace Engineering, Sharif University of Technology, Tehran, Iran
malaek@sharif.edu

1. Introduction

The demand for air transportation is increasing. If the airspace capacity does not grow to match this demand, then flight delays will increase significantly. One of the main sources of flight delays is the inefficiency in airspace management. In order to monitor the air traffic, the airspace is usually divided into multiple sectors, and each sector is monitored by one or more controllers. This type of physical sectorization is completely static while the air traffic is dynamic and changes seasonally, weekly, and daily. Even severe weather conditions and safety issues might shift the air traffic from one sector to the nearby sectors that can suddenly increase the workload of the neighbor sectors. It is also possible that the traffic increases beyond the allowable capacity of one sector. In this situation, no aircrafts would be allowed into that sector until the traffic decreases again. Uneven distribution of the workload among the controllers not only has a direct effect on the flight delays but also decreases the controllers efficiency. This is because dealing with a high workload for a significant amount of time increases human error. Also, a controller who has not had to deal with high workload for a long time would not be able to handle a high sudden workload.

In order to optimize the distribution of the workload among the controllers, many ideas have been proposed including dynamic sectorization. Based on this method, the sectorization should be dynamic and change proportional to the flow of traffic so that the traffic flow can be efficiently managed. Many studies have been done in this field, among them, dynamic programming *(Kulkarni et al., 2011)*, the weighted graph approach *(Martinez et al., 2007)* combined with the genetic algorithm *(Chen et al., 2013)*, combining neighbor sectors based on the traffic demand *(Bloem and Kopardekar, 2008)*, and the geometrical methods for optimal airspace design *(Basu et al., 2009)*. In all these methods, the controller workload is quantified by the number of aircrafts in each sector as a function of time. However, the number of aircrafts per sector is not an accurate estimate of the controllers' workload. **Figure 1** is an example of such a condition. This figure shows two similar sectors with 4 aircrafts passing through them. The aircrafts shown in **Figure 1**-a are flying in parallel routes while in **Figure 1**-b, the aircrafts are flying toward a single point. In this example, although the number of aircrafts in the two sectors are similar, the workload of sector b is higher than that of sector a.

References *(Cummings et al., 2005)*, *(Laudeman et al., 1998)*, *(P. H. Kopardekar et al., 2009)*, *(P. Kopardekar et al., 2008)*, *(Athènes et al., 2002)* and *(Averty et al., 2004)* introduce various parameters that participate in evaluating the workload. In all these papers, conflicts between the aircrafts is one of the most important aspects of the workload. As a result, the goal of this paper is to identify and display the clusters of conflicting aircrafts in a traffic flow using DSM. This helps to sectorize the airspace so that situations similar to **Figure 4**-b would be distributed evenly among the controllers. It should also be noted that the purpose of sectorization in this research is not to define new physical boundaries for the sectors, but it to propose a method that helps to divide the air traffic into manageable parts. Distributing these parts evenly among the controllers assigns one part of the traffic to each controller, which is not physically separable from the parts assigned to other controllers. At the same time, the part of the traffic that each controller monitors does not have any conflict with other parts of the traffic. The proposed sectorization technique changes based on the air traffic flow and, as a result, is considered a method of dynamic sectorization. This paper summarizes the work done in a study on dynamic sectorization *(Farsad, 2010)*.

2. Modeling the airspace sectorization with DSM

The purpose of this research is to develop an algorithm based on a DSM clustering algorithm that enables the sectorization of the traffic flow and graphically demonstrates the sectors. To explain this algorithm, first the workload parameter and the airspace division under this study are defined. Next, simulating the traffic flow, determining and clustering the system's DSM are explained in detail.

In air traffic, some airways are parallel and some cross. The crossing airways can cause conflicts in aircrafts' routes. The workload

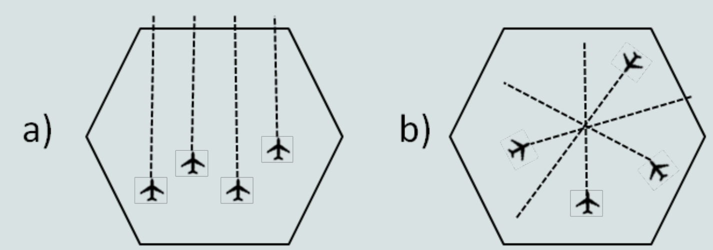

FIGURE 1. *Different workload for the same traffic density. a) Lower workload condition. b) Higher workload condition*

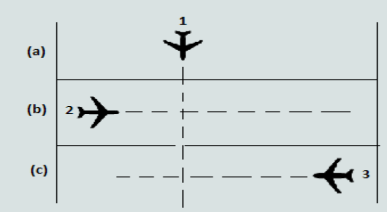

FIGURE 2. *Improper sectorization that requires multiple conflict resolutions*

problem arises when inappropriate sectorization increases the number of conflicts that a controller should resolve so that it surpasses the allowable limit. Determining this limit is not one of the goals of this paper, but this paper helps to develop an optimized sectorization method that minimizes the number of conflicts and divides the remaining conflicts evenly among the controllers. In order to clarify how proper sectorization decreases the number of conflicts, consider **Figure 2**. In this figure, there are three sectors. When aircraft 1 enters sector b, it has a conflict with aircraft 2. When aircraft 1 enters sector c, it has another conflict with aircraft 3. In this situation, when different controllers monitor different sectors, two conflicts need to be resolved. However, if one controller monitors the three aircrafts, aircraft 1's conflicts with aircrafts 2 and 3 can be identified in advance and resolved by changing the speed of aircraft 1 or re-routing this aircraft only once. Clustering the conflicting aircrafts also provides a mental image of the traffic for the controllers that helps them prepare for the upcoming traffic.

Air traffic management has different levels *(Thompson and Viets, 2000)*. This paper evaluates the application of DSM to air traffic management at a multi-sector level. This idea is extendable to higher management levels, such as managing an air traffic control center or multiple centers.

In order to divide the traffic flow among the controllers, this research has developed a code that observes the traffic

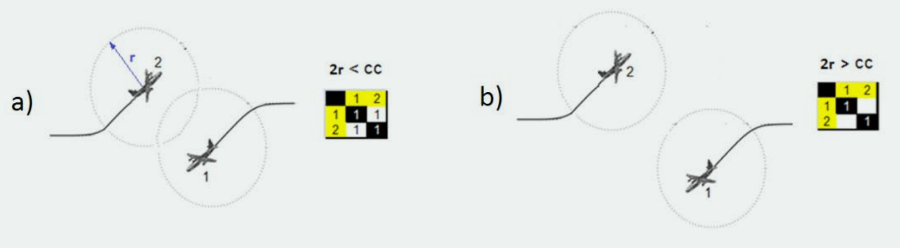

FIGURE 3. *Possibility of aircrafts conflict and the corresponding DSM. a) Conflicting aircrafts. b) Non-conflicting aircrafts*

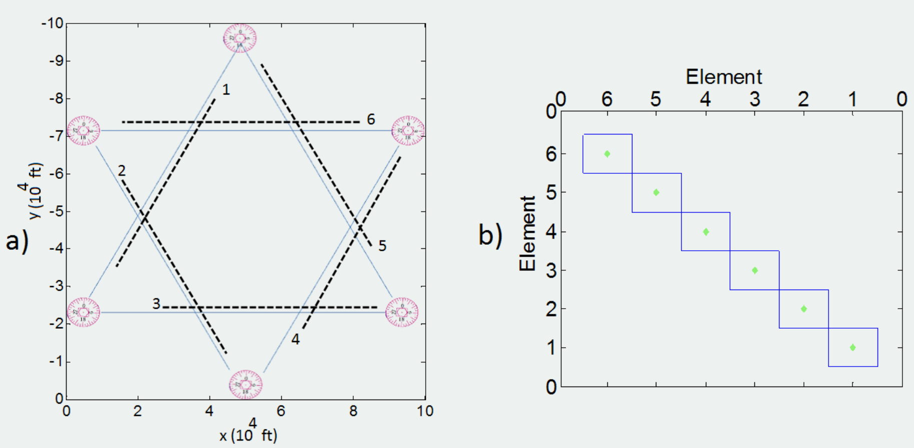

FIGURE 4. *The flow of 6 non-conflicting aircrafts. a) The traffic flow. b) The clustered DSM*

in multiple neighbor sectors, identifies possible conflicts among aircrafts, and demonstrates the conflicts in a DSM that is assigned to the system. The system is a group of aircrafts flying in the sectors under study, and the goal of this paper is to manage their flow so that the secure distance between the aircrafts is not violated. After creating the system's DSM, the aircrafts are divided into different clusters so that the conflicting aircrafts are in the same clusters. At the end, the clusters are graphically demonstrated.

For simulating the traffic flow, the code allows the user to generate aircraft routes by selecting multiple waypoints for each route. The user should define the number of aircrafts and waypoints as inputs. Then the code interpolates the waypoints with a linear spline to generate and demonstrate the routes. The waypoints can demonstrate the spots where radars or pilots determine aircrafts' positions. Next, the code determines which aircrafts have conflicts. A conflict happens when the distance between two aircrafts is less than the radius of the circular protected area *(CPA) (Alaeddini, 2008)*, plus an uncertainty term due to the uncertainty in aircraft position and speed. After determining all the conflicts, the next step is to develop the system's DSM, a square matrix the size of the number of aircrafts. Each row and its corresponding column is assigned to one aircraft and if there is a conflict between two aircrafts, the corresponding DSM coefficient is set to one. For example, if aircrafts 1 and 2 have a conflict and the system's DSM is called A, then a12 and a21 are set to 1. **Figure 3** shows an example of a conflicting and non-conflicting route and the corresponding DSMs. The diagonal elements of the DSMs are set to 1 with the purpose of conforming the DSM to the input of the selected clustering algorithm. The system's DSM is complete when all the conflicts are specified in the matrix. After determining the system's DSM, a clustering algorithm divides the matrix into blocks that demonstrate aircraft clusters.

A study on product development processes *(Thebeau, 2001)* introduces the clustering algorithm used in this paper. This algorithm optimizes a cost function that puts a random element in a cluster that has the highest number of interacting elements with it. This improves the cost function. This algorithm has a random characteristic that allows the reduction of the number of steps required to find the optimum answer and decreases the chance of getting stuck in a local optimum. The algorithm starts with putting each element in a separate cluster. It then randomly selects an element, determines which cluster has the most interacting element with it, and moves it to that cluster if this change decreases the cost function.

3. Case studies

This paper applies the proposed method of using DSM in air traffic control management to some virtual cases of traffic. In order to verify the method, the case studies start with a simple case in which the aircrafts have zero conflicts. Then it analyzes more complicated cases to determine the limits of the method. The designed algorithm lets the user define the virtual traffic flow. Then it runs the defined traffic, finds the conflicts, sets up the system's DSM, determines the clusters, and demonstrates the traffic flow so that the aircrafts in the same cluster are in the same color.

Figure 4 shows a traffic case in which the aircrafts have zero conflicts. As a result, the algorithm sets each aircraft into a separate cluster.

Figure 4-a shows the simulated traffic flow. In this figure, each aircraft is represented with a number. The circles demonstrate the hubs, the solid lines show the airways and dashed lines show part of the predicted route for each aircraft. **Figure 4**-b shows the clustered DSM for this case. As mentioned earlier, each row and the corresponding column of the DSM is designated to one aircraft. A dot placed as a coefficient of this matrix shows a conflict between two aircrafts, and the solid rectangles show the boundaries of a cluster of conflicting aircrafts. In order to see which aircrafts are in a cluster, the easiest way is to observe the rows' or the columns' numbers associated to the clustered diagonal elements.

Figure 5-a shows the traffic flow of 24 aircrafts that is designed so that the traffic can be divided into three non-conflicting clusters. One of the clusters contains aircrafts 1 to 10, another one contains aircrafts 11 to 18, and the last one contains aircrafts 19 to 24. The code successfully determines and demonstrates the clusters. **Figure 5**-b shows the clustered DSM.

Figure 6 shows a more complicated case of 36 aircrafts, in which it is possible to divide the aircrafts into separate clusters. However, an aircraft from one cluster has a conflict with an aircraft from another cluster. Based on **Figure 6**-b, there are four aircraft clusters. The first cluster contains aircrafts 19, 22 to 30; the second cluster contains aircrafts 1, 2, 5 to 12; the third cluster contains aircrafts 20, 21, 31 to 36; and the fourth cluster contains aircrafts 3, 4, 13 to 18. Aircrafts 19 and 20 can create conflicts between the first and the third clusters, and aircrafts 3 and 1 can create conflicts between the second and the fourth clusters.

In order to manage the conflicting clusters, a single controller should manage all of them. If this is not possible, the conflict between the clusters should be resolved in advance by rerouting the associated aircrafts.

As the total number of aircrafts and the size of the clusters increase, the required computational effort goes up because the number of possible ways of clustering significantly increases. Also, it is not possible to analyze all the possible answers to find the optimum response; therefore, the chance of getting stuck in a local optimum goes up. As a result, it is necessary to make sure the cost function reaches a stable amount as a function of the changes in the clusters. The repeatability of the final answer should also be analyzed. In this research, this is done by analyzing the similarity of the clusters in multiple runs by the method introduced in a study on product development processes *(Thebeau, 2001)*. The details of the repeatability analysis, the sensitivity analysis to increasing the total number of aircrafts, the size of the largest cluster, and the number of conflicts between the clusters can be found in a study on dynamic sectorization *(Farsad, 2010)*.

4. Discussion

The methodology described in this paper challenges the concept of physical sectorization of the airspace. In simple terms, not all aircraft flying routes in a given sector are dependent on one another. By appropriately identifying those aircrafts whose flying routes are dependent, it is possible to effectively decrease the controllers' workload and lessen the chance of any undesirable conflicts. The method and tool described in this work can also be helpful to dynamically re-architect air spaces into sectors with a comparable number of aircraft so that the number of conflicts that air traffic controllers need to consider is minimized.

The results presented in this work effectively show the feasibility of applying DSM to air traffic management. Nonetheless, one should note that for real scenarios, we need to have access to each individual aircraft's flying capabilities to model its true 4D trajectory. Obviously, in absence of such data, RADAR coverage could provide the necessary information to approximately model any aircraft trajectory and therefore its flight-path dependencies on other aircraft in its neighborhood. In fact, we are in the process of applying the method to old-real traffic data. In this work, the boundaries of the traffic have been confined to multiple neighboring sectors; nonetheless, the application of such a method to global traffic management is the real challenge.

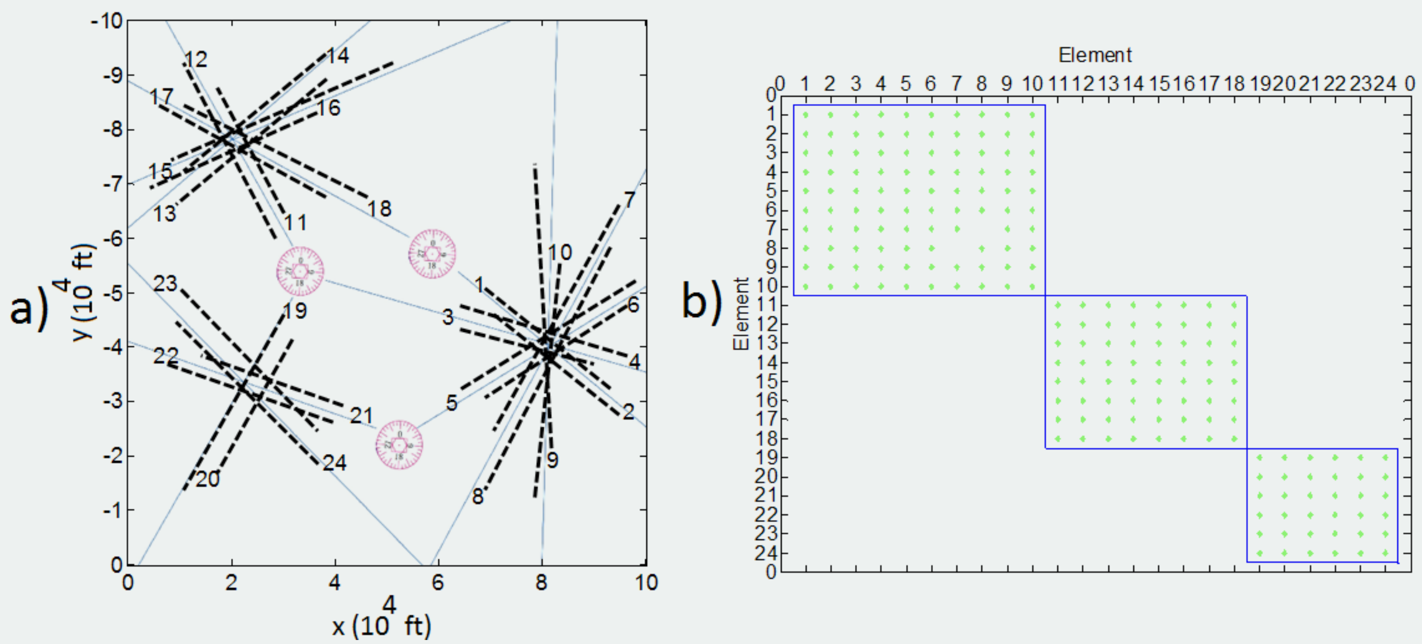

FIGURE 5. *The flow of 24 aircrafts dividable into three non-conflicting clusters. a) The traffic flow. b) The clustered DSM*

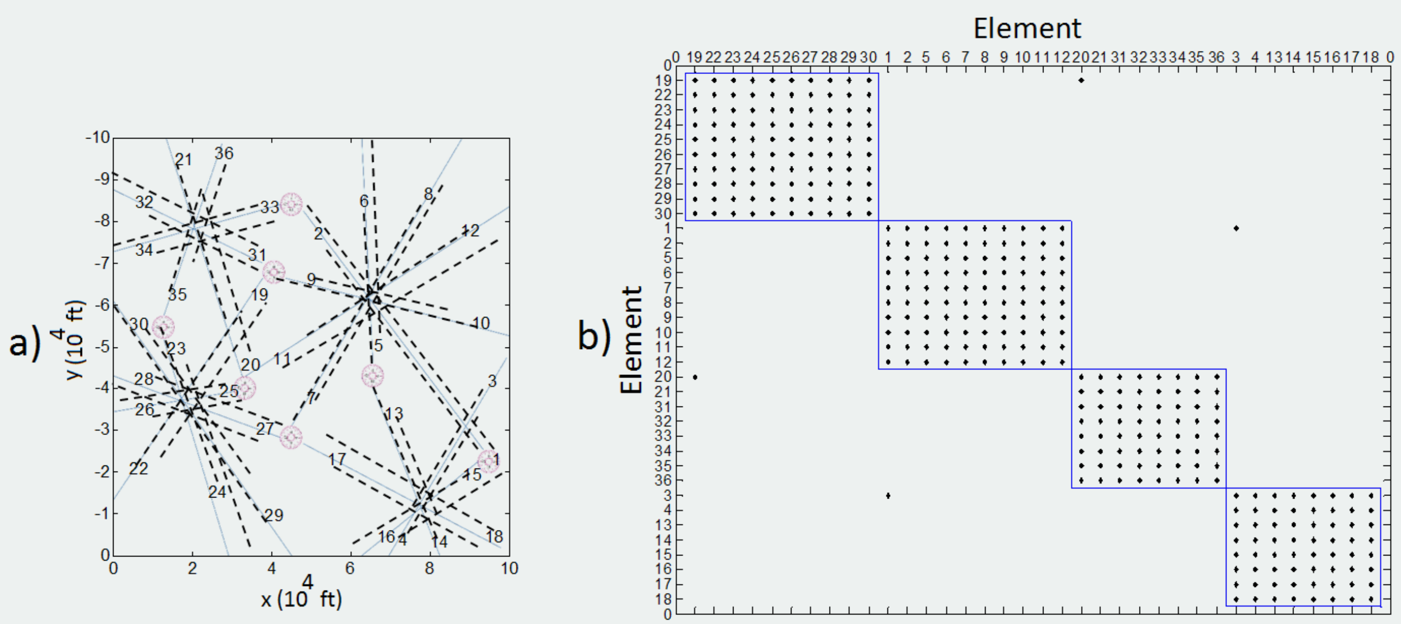

FIGURE 6. *The flow of 36 aircrafts dividable into four conflicting clusters. a) The traffic flow. b) The clustered DSM*

authors

Mahsa Farsad earned her PhD in Optical Science and Engineering (OSE) from University of North Carolina at Charlotte (UNCC) in May 2015, where she worked on development of a displacement scale based on speckle patterns with Prof. Chris Evans. She holds a masters degree in OSE (2014, UNCC), a masters and a bachelor's degree in Aerospace Engineering (Sharif University of Technology, Tehran, Iran, 2011 and 2008 respectively). In 2012, the was awarded the Small Business Innovation Research Award by NIST to work as the Principal Investigator on a project titled "Low cost stabilized laser diode system".

Seyed M. Malaek is a Professor at the Sharif University of Technology in Tehran, Iran. His research interests include Engineering Systems as well as Systems Engineering together with Aerial Transportation, Performance optimization and Air Traffic Management. He is the author of two books in flight dynamics and Modeling Techniques and three books related to the Socio-Technical Systems (all in Persian). He has also published over 65 papers in the areas indicated above. He also holds a patent on rotary cockpit for single seat aircraft and Auto-Shut-off an Ignition Systems. Dr. Malaek obtained his M.S. degree in 1986 and his Ph.D. in 1990 from University of Kansas U.S.A. and his BSc. From Sharif University of Technology in Iran.

references

Alaeddini, A [2008], 'Efficient Webs for Conflict Resolving Maneuvers based on Genetic Algorithms', Master of Science Dissertation, Sharif University of Technology, Tehran, Iran, Vol.

Athènes, Sylvie, et al. [2002], 'ATC complexity and controller workload: Trying to bridge the gap', Proceedings of the International Conference on HCI in Aeronautics, pp. 56-60.

Averty, Philippe, et al. [2004], 'Mental workload in air traffic control: an index constructed from field tests', Aviation, space, and environmental medicine, Vol. 75, No. 4, pp. 333-41.

Basu, Amitabh, Mitchell, Joseph SB, and Sabhnani, Girish Kumar [2009], 'Geometric algorithms for optimal airspace design and air traffic controller workload balancing', Journal of Experimental Algorithmics *(JEA)*, Vol. 14, pp. 3.

Bloem, Michael and Kopardekar, Parimal [2008], 'Combining airspace sectors for the efficient use of air traffic control resources', Proc. of AIAA Guidance, Navigation, and Control Conference and Exhibit, Honolulu, HI.

Chen, Yangzhou, et al. [2013], 'Dynamic airspace sectorization via improved genetic algorithm', Journal of Modern Transportation, Vol., pp. 1-8.

Cummings, ML, Tsonis, CG, and Cunha, DC [2005], 'Complexity mitigation through airspace structure', 13th International Symposium on Aviation Psychology, Oklahoma City, Oklahoma.

Farsad, M [2010], 'Application of Dynamic Sectorization to Air Traffic Controller Workload Balancing', Master of Science Dissertation, Sharif University of Technology, Tehran, Iran, Vol.

Kopardekar, Parimal, et al. [2008], 'Relationship of maximum manageable air traffic control complexity and sector capacity', 26th International Congress of the Aeronautical Sciences *(ICAS 2008)*, and AIAA-ATIO-2008-8885, Anchorage, Alaska, Sept, pp. 15-19.

Kopardekar, Parimal H, et al. [2009], 'Airspace complexity measurement: An air traffic control simulation analysis', International Journal of Industrial Engineering: Theory, Applications and Practice, Vol. 16, No. 1, pp. 61-70.

Kulkarni, S., Ganesan, R., and Sherry, L. [2011], 'Static sectorization approach to dynamic airspace configuration using approximate dynamic programming', Integrated Communications, Navigation and Surveilance Conference *(ICNS)*, 2011, pp. J2-1-J2-9.

Laudeman, IV, et al. [1998], 'Dynamic density: An air traffic management metric'.

Martinez, S, et al. [2007], 'A weighted-graph approach for dynamic airspace configuration'.

Thebeau, Ronnie Emile [2001], 'Knowledge management of system interfaces and interactions from product development processes', *(Massachusetts Institute of Technology)*.

Thompson, Kenneth H and Viets, Karen J [2000], 'Layered Strategic Planning in the En Route National Airspace System *(NAS)*: Air Route Traffic Control Center *(ARTCC)* Perspective', The MITRE Corporation, McLean, VA, MTR 00W0 000, Vol. 64.