Technical Note

Airspace sectorisation via a weighted graph model

D.F. Zhang
defuzhang@emails.bjut.edu.cn
Y.Z. Chen
yzchen@bjut.edu.cn
School of Electronic Information and Control Engineering
Beijing University of Technology
Beijing, China

1.0 INTRODUCTION

With the development of air traffic, flight delays happen frequently due to bad weather and traffic congestion. The problem can be solved partly by certain strategies, such as changing air routes. However, rerouting leads to a global imbalance in controller workload of current sectors and to an increase in co-ordination workload, and the workloads of some sectors may be beyond the controller's ability to manage. Thus, airspace sectorisation is expected to migrate from the current static sectors to dynamically-changing ones capable of adapting to traffic demand. Besides addressing imbalance and controlling the increase in workload, the sectorisation has to meet additional geometric constraints such as convexity, connectivity, and minimum distance constraint[1].

Quite a number of sectorisation methods have been developed in recent years. These methods proposed the strategy that described airspace as a model and then applied a proper algorithm to partition airspace into sectors. In the light of the available models, these methods are classified into four types, i.e. cell, flight trajectory, Voronoi diagram, and graph. In the cell model, airspace is

Paper No. 4102. Manuscript received 3 October 2013, accepted 9 December 2013.

https://doi.org/10.1017/S0001924000009118 Published online by Cambridge University Press

first divided into hexagonal grids, and then Yousefi *et al.*[2], Klein[3], and Drew *et al.*[4,5] combined those grids into sectors using the standard facility location algorithm, region-growth algorithm, and mixed integer programming separately. Regrettably, designed sectors may have undesired shapes due to their 'jagged' sector boundaries. In the flight trajectory model, Briton *et al.*[6] grouped flight positions into sectors using *k*-means algorithm and Basu *et al.*[7] developed geometric algorithms for sectorisation. In the Voronoi diagram model, Delahaye[8] proposed initial sectors constructed arbitrarily via Voronoi diagram to partition airspace into units with convex shape and then optimised them by evolutionary algorithm. Furthermore, Xue[9] improved Delahaye's scenario via the iterative deepening algorithm. It should be noted that a common limitation in the three models is that researchers failed to make use of information on airspace structure. This might lead to the case that the designed sectors may dissatisfy those geometric constraints. Encouragingly, the static airspace structure was considered sufficiently in the graph model, where vertices represent airports, waypoint and crossing points while edges represent air routes. Based on the graph model, Trandac[1], Martinez *et al.*[10], Zhang[11] and Li *et al.*[12] applied a constraint algorithm, spectral bisection algorithm, and spectral clustering to airspace sectorisation respectively.

This note presents an airspace sectorisation method that aims at the workload balance in a way that minimises co-ordination workload while designed sectors meet geometric constraints. The given airspace is first decomposed into units called cells by a Voronoi diagram[13]. Then, the workload of each cell and the co-ordination workload along the air route between cells are calculated. Furthermore, the Voronoi diagram is simplified into an accessorial weighted graph, whose vertices represent the Voronoi cells with workloads and edges representing air routes with workloads between the cells respectively. In this way, the airspace sectorisation problem is converted into the partitioning problem of the weighted graph. Secondly, the weighted graph model is divided into sub-graphs by a partition algorithm we develop. The algorithm harmonises an Improved Spectral Graph Partitioning (ISGP) algorithm[14], an Optimal Dynamic Load Balancing (ODLB) algorithm[15] and a heuristic algorithm inspired from the concept of relative gain[16]. After the ISGP algorithm is applied to the weighted graph iteratively to partition it into a series of sub-graphs, ODLB algorithm together with the heuristic algorithm is applied to improve the workload balancing. Lastly, we combine those cells together corresponding to each sub-graph to construct sectors.

2.0 AIRSPACE SECTORISATION METHOD

2.1 Construction of weighted graph model

The objective is to construct a weighted graph model in which the static airspace structure is taken into account and air traffic data is utilised. Assuming that a given airspace is known in advance in a form of an undirected graph $G = G(V,E)$, where the vertex set $V = \{1,2,\ldots,n\}$ consists of the airports, waypoints and crossing points, and the edge set $E = \{ (ij): i,j \in \mathrm{V} \}$ represents all air routes.

According to G, a Voronoi diagram D[13] is first built whose sites are the vertices of G. D can decompose by its borders the airspace into a series of units called Voronoi cells C_i ($i = 1,2,\ldots,n$), where a cell corresponds to only one site. When cells are combined to form sectors, part of the borders will be the sector boundaries and the convexity of designed sectors will be automatically satisfied.

From the construction of D, one can see that there may be a case for some of the sites or air routes being close to some cell borders. This leads to the result that some of the designed

https://doi.org/10.1017/S0001924000009118 Published online by Cambridge University Press

sectors will not satisfy the minimum distance constraint if the sector boundaries coincide with these borders. So, the pretreatment of these borders is necessary. The available practice is that such borders are removed, and cells connected by the border will be merged into a new cell. The pretreatment will produce some new cells, and we assume that there will be r cells C_i ($i = 1,2,\ldots,r$, and $r \leq n$) in the pretreated Voronoi diagram. In this way, the designed sectors will satisfy the minimum distance constraint.

The workload objective in the note is to maximise workload balance and to minimise co-ordination workload when cells are combined into sectors. Aircraft count is adopted as workload metric. While the sectorisation is implemented, we expect fewer aircraft count flying across the boundaries of the sectors and fewer aircraft count means less co-ordination workload. Thus, aircraft counts for each cell and along each air route are required to be computed ahead of time. Here, it takes into account aircraft count w_i flying in cell C_i at peak-traffic time and aircraft count along each air route is considered over a period.

From the pretreated Voronoi diagram, one can see that there may be several or more edges between two cells, and there may be several vertices in a cell. This leads to a little difficulty in further analysis. In order to facilitate the discussion, the pretreated Voronoi diagram with workloads is further simplified into an accessorial weighted graph model C_w, whose vertex v_i represents the cell C_i and is endowed with w_i in C_i as the weight, and whose edge e_{ij} represents all air routes between cells C_i and C_j and the edge weight w_{ij} describes the sum of aircraft counts along all air routes between C_i and C_j. Here, the accessorial weighted graph differs from undirected graph without weights on both edge and vertex used in Ref. 10, and is also at variance with the traditional weighted graph in which only edges are endowed with weight values in Ref. 12, but an undirected graph with the weights on both vertices and edges. This is a key feature of our graph model. The advantage lies in the fact that the accessorial weighted graph being embedded in the information concerning workload and co-ordination workload provides the convenience of applying a proper algorithm to achieve the workload objective. Furthermore, the weights on vertices are described by a vector **w** as Equation (1), and the weights on edges are described by a matrix **W** as Equation (2) as follows:

$$\mathbf{w} = (w_1, w_2,\ldots,w_r)^T \quad \ldots (1)$$

$$\mathbf{W} = (w_{ij})_{r\times r},\ w_{ij} = w_{ji} \quad \ldots (2)$$

From the construction of G_w, one can know that there is an exact one-to-one relationship between the vertices of G_w and the cells. The relationship offers real convenience for the property obtained from the operation on G_w being propagated back to the pretreated Voronoi diagram. In this way, the airspace sectorisation problem with the objective that balances sector workload and minimises the co-ordination workload is converted into the partitioning problem of the weighted graph to maximise sub-graph weight balance and to minimise edge weight crossing sub-graphs.

2.2 Partition of weighted graph model

Now, assume that a weighted graph model obtained in Subsection 2.1 is $G_w = (V_w, E_w, w, W)$, where $V_w = \{v_1, v_2,\ldots,v_r\}$ is the vertex set, $E_w = \{e_{ij}: v_i, v_j \in V_w\}$ is the edge set, **w** is the vector described in Equation (1), and **W** is the matrix described in Equation (2). G_w will be partitioned into a set of k disjoint sub-graphs $G_w^{i'}$ ($i = 1,2,\ldots, k$) with the objective of balancing sub-graph weight $w(G_w^{i'})$ and minimising the edge weight crossing sub-graphs described mathematically as a function:

https://doi.org/10.1017/S0001924000009118 Published online by Cambridge University Press

$$\min_{G_w^{1'},\ldots,G_w^{k'}} \sum_{c=1}^{k} \frac{\text{cut}(G_w^{i'}, G_w / G_w^{i'})}{w(G_w^{i'})} \quad \ldots (3)$$

$$\text{subject to } w(G_w^{i'}) = w(G_w^{j'}) \quad \ldots (4)$$

where: $$w(G_w^{i'}) = \sum_{v_c \in G_w^{i'}} w_c, \ \text{cut}(G_w^{i'}, G_w / G_w^{i'}) = \sum_{v_c \in G_w^{i'}, v_d \notin G_w^{i'}} w_{cd} \quad \ldots (5)$$

Focusing on the above objective functions (3) and (4), we develop a partition algorithm to divide G_w. The algorithm includes two steps, that is 1) partitioning the graph into k sub-graphs iteratively by ISGP algorithm, and 2) migrating vertices via ODLB algorithm together with a heuristic algorithm inspired from the relative gain to improve sub-graph weight balancing.

We first use the ISGP algorithm[14] to divide G_w into two sub-graphs G_w^i ($i = 1,2$) holding connectivity. The algorithm can achieve an objective as follows:

$$J = \min_{G_w^1, G_w^2} \sum_{i=1}^{2} \frac{\text{cut}(G_w^i, G_w / G_w^i)}{w(G_w^i)} \quad \ldots (6)$$

where $$w(G_w^i) = \sum_{v_c \in G_w^i}^{2} w_c \ \text{cut}(G_w^i, G_w / G_w^i) = \sum_{v_c \in G_w^i, v_d \notin G_w^i} w_{cd} \quad \ldots (7)$$

From Equation (6), we know that 1) the smaller the cut in Equation (7), the smaller J; 2) with the given cut, J is minimised while balancing $w(G_w^i)$ between two sub-graphs. In fact, ISGP algorithm proposes a compromise scheme for dividing G_w in view of both balancing $w(G_w^i)$ and minimising the cut. That is to say, $w(G_w^i)$ on the condition that minimises the cut may be not equal, i.e. $w(G_w^1) \neq w(G_w^2)$ (assuming that $w(G_w^1) > w(G_w^2)$). Thus, it is necessary to transfer the vertices form from G_w^1 to G_w^2 to balance sub-graph weight via ODLB algorithm together with a heuristic algorithm.

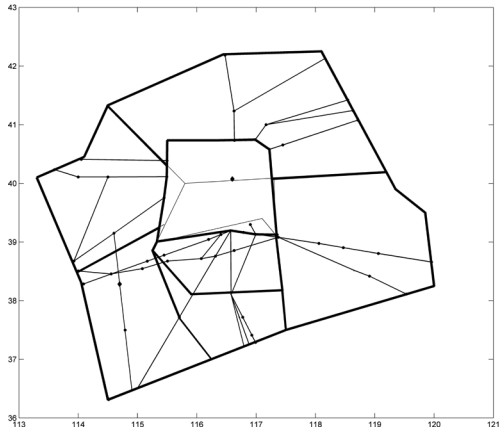
The migration of vertex will be divided into two steps. In the first step, the ODLB algorithm[15] gives the amount of weight x_{12} transferred from G_w^1 to G_w^2. x_{12} may not be an integer, while the weight representing aircraft count requires x_{12} to be an integer. So instead of x_{12}, we use an integer x_{12}^r by rounding x_{12}. In the second step, we will transfer vertices with weights from the G_w^1 to G_w^2 according to x_{12}^r. However, if x_{12}^r can be satisfied by migrating vertices optionally, it is impossible to minimise the cut as Equation (3) in the original scheme. So, a heuristic algorithm is proposed inspired by the concept of the relative gain[16] to ensure that the cut is minimised. Let the vertices in G_w^1 adjacent to G_w^2 be denoted as B_{12}, and let the sum of weights corresponding to vertices in B_{12} be a_{12}, the gain of v_d can be calculated by the equation as:

$$g_d = \sum_{v_d \in G_w^i, v_f \in G_w^j} w_{df} - \sum_{v_d, v_e \in G_w^i} w_{de} \quad \ldots (8)$$

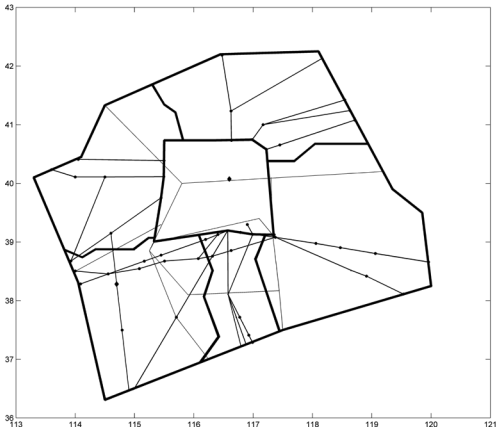
Similarly, let the vertices in G_w^2 adjacent to G_w^1 be denoted as B_{21}, and let the vertex set in B_{21} shares common edge with the vertex **d** in B_{12} denoted as C, the number of vertex in C is $|C|$. The relative gain g_{td} of v_d in B_{12} is determined by the following equation as (9):

$$g_{rd} = g_d - \sum_{v_f \in C} \left(g_f\right) / |C| \quad \ldots (9)$$

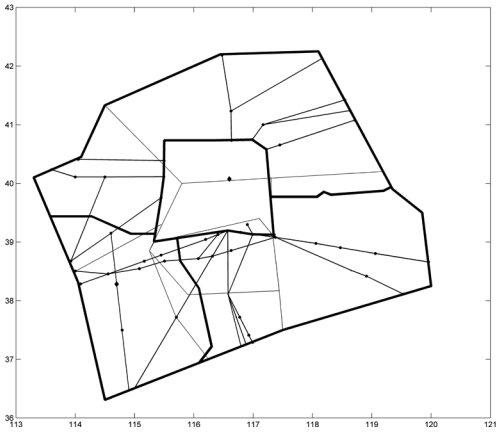
https://doi.org/10.1017/S0001924000009118 Published online by Cambridge University Press



(a)



(b)



(c)

Figure 1. Beijing air traffic area (a), new sectors of BJA via the method for two different time intervals: 15:00-17:00(b), 21:00-23:00(c). (Thick lines represent boundaries of sectors, thin lines represent air routes, the horizontal axis represents east longitude, the vertical axis represents north latitude).

https://doi.org/10.1017/S0001924000009118 Published online by Cambridge University Press

And the vertices in B_{12} be sorted according to their relative gains by descending order. There are two cases for migration. One case is $a_{ij} > x^r_{12}$. In this case, the vertex B_{12} in with the largest gains first moves to G^2_w, and the rest of the vertices follow according to the descending order until x^r_{12} is satisfied. The other case is $a_{ij} > x^r_{12}$. After all vertices in B_{12} move to G^2_w, new vertices in G^1_w adjacent to G^2_w will appear. The above procedure is repeated for x^r_{12}. Finally, two sub-graphs obtained from G^1_w to G^2_w are denoted by $G^{1'}_w$ and $G^{2'}_w$, which have equal weights. Using the heuristic algorithm, the cut between $G^{1'}_w$ and $G^{2'}_w$ is minimised.

If $w(G^{1'}_w)$ and $(G^{2'}_w)$ and are below the given maximum L_{max}, there is no need to further bisect the sub-graphs. Otherwise, the sub-graph G^1_w with large weight is selected, and is bisected into its sub-graphs $G^{1_1}_w$ and $G^{2_2}_w$. Then, ODLB algorithm is used to calculate the amount of weight required to be migrated among $G^{1_1}_w$, $G^{1_2}_w$ and G^2_w, and vertices are transferred via the heuristic algorithm. Such iterative bisection and migration are repeated until the weights of sub-graphs are below L_{max}. Finally, we obtain a series of sub-graphs $G^{i'}_w$ ($i = 1,2,\ldots,k$) with balanced sub-graph weight and minimum edges weight crossing sub-graphs. Precisely, $G^{i'}_w$ satisfies the properties as Equation (10):

$$w(G^{i'}_w) \le L_{\max}\ w(G^{i'}_w) = w(G^{j'}_w) \min_{G^{1'}_w,\ldots,G^{k'}_w} \sum_{i=1}^{k} \mathrm{cut}(G^{i'}_w, G_w / G^{i'}_w)\ G_w = \bigcup_{i=1}^{k} G^{i'}_w\ G^{i'}_w \cap G^{j'}_w = \varnothing \qquad \ldots (10)$$

2.3 Determination of Sectors

From the construction of the weighted graph G_w, there is an exact one-to-one relationship between its vertices and the corresponding cells in the pretreated Voronoi diagram. The vertices in $G^{i'}_w$ are mapped back to the corresponding cells, and those cells are combined together to form the sector S_i. During sectorisation, the properties as Equation (10) are propagated back to the designed sectors. So, we finally obtain k sectors from k sub-graphs $G^{i'}_w$ ($i = 1,2,\ldots,k$) with balancing the aircraft count and minimising co-ordination workload.

3.0 EXAMPLE AND SIMULATION

In this section, the method is applied to the airspace of Beijing air traffic area (BJA) above 18,000 feet altitude with real traffic data to verify its feasibility and effectiveness, see Fig. 1(a). The centre of BJA has been specifically designed for certain reasons, so we focus on the rest of BJA. From Fig. 1(a), we know that there are six sectors for current air traffic management. For current sectors, a fact comes before us that some sector boundaries overlap with the waypoints and crossing points, and a lot of boundaries are too close to certain air routes. Moreover, there is an imbalance of sector workloads. Such a situation can potentially have a negative effect on flight security, so it is necessary to redesign sectors for BJA.

Some parameters for sectorisation are set as follows: 1) the aircraft count of each sector is set to 10, and the redundancy of 20% is adopted for the sake of the reliability of DAC and the safety of aircraft; 2) the minimum distance between airports and the sector boundary, between waypoints and the boundary, between waypoints and the boundary is set to 15nm, 9nm and 3nm respectively. We take air traffic every two hours into account. The peak traffic of each interval is applied for sectorisation, for it takes into account the controller's maximum ability to monitor and provide traffic flow control within the confines of each sector.

Fig.1 (b) and (c) give different sectorisations of BJA computed by the method for two different time intervals. It is seen that 1) all sectors satisfy the geometric constraints, and 2) the number of sectors computed by our method varies over time, for the air traffic is varying over time.

https://doi.org/10.1017/S0001924000009118 Published online by Cambridge University Press

For the air traffic, 1) during 15:00-17:00, there are five new sectors with aircraft count: 7, 6, 7, 7, 8, and total co-ordination workload is 71, while aircraft count for current sectors are 4, 5, 6, 4, 7, 9 and total co-ordination workload is 109; 2) during 21:00-23:00, three new sectors come into being with aircraft count: 5, 7, 6, and total co-ordination workload is 11, while aircraft count for current sectors are 4, 2, 0, 3, 4, 5 and total co-ordination workload is 61. By comparison, it is obvious that the sectors designed by our method have more balanced workload and less co-ordination workload than current sectors.

4.0 CONCLUSION

The note presents a sectorisation method for transforming the current static and structured airspace sectorisation to a more dynamic airspace sectorisation which can better accommodate the changing traffic pattern and user demand. Simulation results show designed sectors not only have balanced workloads and minimise co-ordination workload, but also satisfy geometrical constraints, such as convexity constraint, connectivity constraint and minimum distance constraint. Furthermore, a cause of jubilation is that when traffic is low dynamically, changing sectors may result in fewer sectors than those of current sector, followed by a promising reduction in controller count which cut down the administrative costs. The performance of simulation validates the feasibility and effectiveness of the method.

ACKNOWLEDGEMENTS

The authors wish to acknowledge the support of the Joint Funds of the National Natural Science Foundation of China (Grant No. 61079001).

REFERENCES

1. TRANDAC, H. and DUONG, V. Optimised Sectorisation of Airspace with Constraints, Proceedings of the 5th Eurocontrol/FAA ATM R&D Seminar, Budapest, Hungary, June 2003.
2. YOUSEFI, A. and DONOHUE, G.L. Temporal and Spatial Distribution of Airspace Complexity for New Methodologies in Airspace Design, Proceedings of the 4th AIAA Aviation Technology, Integration, and Operations Conference (ATIO), Chicago, IL, USA, September 2004.
3. KLEIN, A. An Efficient Method for Airspace Analysis and Partitioning Based on Equalized Traffic Mass, Proceedings of the 6th USA/Europe Seminar on Air Traffic Management Research and Development, Baltimore, MD, USA, June 2005.
4. DREW, M. Analysis of an Optimal Sector Design Method, The 27th Digital Avionics Systems Conference, IEEE/AIAA, St. Paul, MN, USA, 2008, p 3, .B.4-1-3.B.4-1-10.
5. TIEN, S.-L. and HOFFMAN, R. Optimizing Airspace Sectors for Varying Demand Patterns Using Multi-Controller Staffing, 8th USA/Europe Air Traffic Management Research and Development Seminar, California, USA, June 2009.
6. BRINTON, C. and PLEDGIE, S. Airspace Partitioning Using Flight Clustering and computational Geometry, 27th Digital Avionics Systems Conference, NJ, USA, October 2008, **1**, p 3, .B.3–1–3.B.3–10.
7. BASU, A., MITCHELL, J. and SABHNANI, G. Geometric Algorithms for Optimal Airspace Design and Air Traffic Controller Workload Balancing, 16th Fall Workshop on Computational and Combinatorial Geometry, Cambridge, UK, November 2008.
8. DELAHAYE, D., SCHOENAUER, M and ALLIOT, J.M. Airspace Sectoring by Evolutionary Computation, Proceedings of the IEEE International Congress on Evolutionary Computation, Anchorage, USA, 1998.
9. XUE, M. Airspace Sector Redesign Based on Voronoi Diagrams, AIAA Guidance, Navigation, and Control Conference, August 2008.
10. MARTINEZ, S., CHATTERJI, G., SUN, D and BAYEN, A. A Weighted-graph Approach for Dynamic Airspace Configuration, Proceedings of the AIAA Guidance, Navigation, and Control Conference, Hilton Head, SC, USA, August 2007.

https://doi.org/10.1017/S0001924000009118 Published online by Cambridge University Press

11. ZHANG, D.F., CHEN, Y.Z., BI, H. and SONG, Z.X. Airspace Sectorisation Based on Weighted Graph Spectral Bisection Algorithm, 12th COTA International Conference of Transportation Professionals, Beijing, China, August 2012.
12. LI, J.H., WANG, T., SAVAI, M. and HWANG, I. Graph-based algorithm for dynamic airspace configuration, *J Guidance, Control, and Dynamics*, 2010, **33**, (1), pp 1082-1095.
13. FORTUNE, S.J. A sweepline algorithm for voronoi diagrams, *Algorithmica*, 1987, **2**, pp 153-174.
14. HENDRICKSONAND, B. and LELAND, R. An improved spectral graph partitioning algorithm for mapping parallel computations, *J SIAM Sci Comput*, 1995, **16**, (1), pp 452-469.
15. HU, Y.F. and BLAKE, R.J. An improved diffusion algorithm for dynamic load balancing, *J Parallel Computing*, April 1999, **25**, (4), pp 417- 444.
16. WALSHAW, C., CROSS, M and EVERETT, M.G. Parallel dynamic partitioning for adaptive unstructured meshes, *J Parallel And Distributed Computing*, 1997, **47**, pp 102-108.

https://doi.org/10.1017/S0001924000009118 Published online by Cambridge University Press