Article

A Method Framework for Automatic Airspace Reconfiguration-Monte Carlo Method for Eliminating Irregular Sector Shapes Generated by Region Growth Method

ZhiJian Ye[1*] FanHe Kong[2] BaoCheng Zhang[3] Wei Gao[4] JianFeng Mao[5]

1. College of Air Traffic Management, Civil Aviation University of China, Tianjin 300300, China
2. Dalian ATM Station of Northeast ATMB, Dalian 116033, China; DalianATM@163.com
3. College of Air Traffic Management, Civil Aviation University of China, Tianjin 300300, China; BCZhang@cauc.edu.cn
4. College of Air Traffic Management, Civil Aviation University of China, Tianjin 300300, China; WeiG@cauc.edu.cn
5. School of Science and Engineering, the Chinese University of Hong Kong, Shenzhen; jfmao@cuhk.edu.cn

* Correspondence: zjye@cauc.edu.cn; Tel.: +86-130-2222-6938

Abstract: With the growth of air traffic demand in busy airspace, there is an urgent need for airspace sectorization to increase air traffic throughput and ease the pressure on controllers. The purpose of this paper is to develop a method framework that can perform airspace sectorization automatically, reasonably, which can be used as an advisory tool for controllers as an automatic system, especially for eliminating irregular sector shapes generated by simulated annealing algorithm (SAA) based on region growth method. Two graph cutting method, dynamic Monte Carlo method by changing location of flexible vertices (MC-CLFV) and Monte Carlo method by radius changing (MC-RC) were developed to eliminating irregular sector shapes generated by SAA in post-processing. The experimental results show that the proposed method framework of AS can automatically and reasonably generate sector design schemes that meet the design criteria. Our methodology framework and software can provide assistant design and analysis tools for airspace planners to design airspace, improve the reliability and efficiency of airspace design, and reduce the burden of airspace planners. In addition, this lays the foundation for reconstructing airspace with more intelligent method.

Keywords: Airspace Reconfiguration; irregular boundary smoothing; dynamic Monte Carlo method by changing location of flexible vertices; Monte Carlo method by radius changing; Voronoi diagram; graph cutting; multi-objective optimization

1. Introduction

In recent years, the air traffic flow of China has been increasing at an average annual rate of 11 percent, and air transport is developing rapidly. With the rapid development of national traffic volume, China's airports and airspace system are under increasing pressure, which leads to frequent flight delays. In 2010, the fixed assets investment of Chinese airports increased from 44.150 billion Yuan to 56.08 billion Yuan, with a compound annual growth rate of 2.54% [1]. Thus in order to solve the imbalance between demand and capacity, specific measures include strengthening the construction of infrastructure and expanding fleet was taken. However, according to the statistics of CAAC (Civil Aviation Administration of China), average flight punctuality rate in china maintained at a low level (as shown in Figure 1) in the past few years [2].

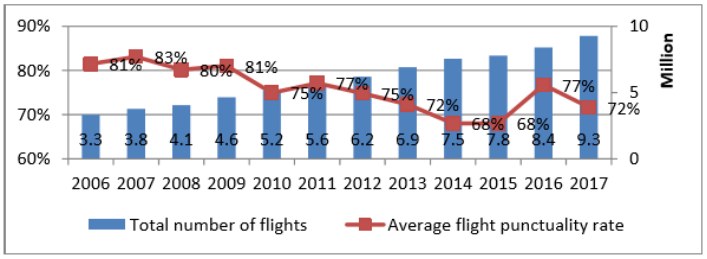

Figure 1. Average flight punctuality rate and total number of flights

This indicates that the government needs to not only strengthen the construction of infrastructure, but also give advice on how to manage the airspace effectively. The current Chinese National Airspace architecture is reaching the ability limits to accommodate the increases in traffic demand. As the key limiting factor, today's sector boundaries are largely determined by historical user profiles which have evolved slowly over time. Consequently the sector geometry has stayed relatively constant despite the fact that route structures and demand have changed dramatically over the years. There are usually two forms of traffic flow growth in airspace. The first is that the traffic flow of different routes has different growth rates with time. Some sectors have excessive task load, while others have relatively small task load. In this case, the problem that throughput is limited by sector capacity can be solved by adjusting sector structure and keeping the number of sectors unchanged. The second is that the traffic flow of all routes in the whole airspace has increased significantly. The control workload cannot be effectively reduced by sector structure adjustment alone. The throughput can only be increased by opening more sectors and airspace sectorization (AS).

Although AS is the most potential method to increase throughput of airspace as well as to better use scarce resources, there are a serious of challenge to realize it. Except for the accessible of airspace sectorization by air traffic controller, a lot of constraints must be considered in accordance with the demand of Air Traffic Control operation. Because AS needs to consider the constraints of balance of task load, shape, connectivity and compactness, which are beyond the limits of human ability, a mathematical optimization tool is needed to support this process [3, 4].

A lot of AS research has been carried out. Delahaye, et al. [5] initiated the study of balanced sector of airspace in order to increase air traffic control capacity in high density areas. Genetic algorithms were used to compute a balanced regrouping of air traffic control sectors thus to optimally reduce the number of controller teams [6]. A decision support tool based on fuzzy logic was proposed to determine the number of open sectors during a given time period [7]. Related researches include dynamic airspace configuration (DAC), dynamic airspace sectorisation (DAS) and airspace configuration. In DAC, the airspace is adjusted in real-time to accommodate fluctuating demand patterns [3, 8-12]. A detailed overview of recent work of DAC is given by Zou, et al. [13]. Transitional workload is considered in most of these DAC models. Gerdes, Temme and Schultz [11] clearly distinguished DAC from DAS. DAS is a flexible method, encompassing the idea of unstructured and (rigid) structured airspace, and differing from DAC, where predefined airspace blocks are combined to form new structures (merging and splitting). DAS approach creates a less familiar setting for controllers but can be used for situations where no basic knowledge concerning the best airspace structure is available. This paper mainly solves the problem of increasing number of sectors or improving sectors caused by the rapid growth of China's traffic volume. Our aim is to adjust the sector structure to increase capacity. Civil Aviation of China stipulates that as long as the sector structure changes, controllers must carry out pre-job training. The problem of transitional workload is another problem of controller's assignment with minimizing training cost. AS in this paper is closer to DAS，because it is a strategic or tactical AS before the implementation of AS scheme.

Flener and Pearson [14] systematically surveyed the algorithmic aspects of methods for automatic airspace sectorization , for an intended readership of experts on air traffic management. Most of the airspace sectorization models were based on two approaches. The first one is graph-based model. In this type of model, a graph is constructed whose vertices represent the intersections of the existing trajectories while edges represent segments [5, 15-17]. Several recent published papers also use this type of model [13, 18, 19]. Graph-based model could use some mature technology in graph partition, such as MIN-CUT graph partition used as minimizing coordination task load. However, a graph partition does not define the sector boundaries,

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 July 2019 doi:10.20944/preprints201907.0287.v1

so actual sectors have to be constructed from the resulting vertex sets in a geometric post-processing step. The second one is region-based model in which the airspace is initially partitioned into some type of regions which are smaller than the studied sectors, so that the combinatorial problem of partitioning these regions in principle needs no geometric post-processing step [20-28]. Unfortunately, shapes of obtained sectors which use region-based model still have to need post-processing. Although region-based model is essentially a down-top combination approach, in which workload among the sectors was balanced by maintaining the sector shape parameters within acceptable limits, but it made them operationally undesirable that there were some corners or Z-shape among these sectors even after post-processing[29, 30].

[31, 32] developed another top-down bi-partition approach to cut airspace into sectors. Xue [8], Gerdes, Temme and Schultz [11], Xue [33], Gerdes, et al. [34] also use top-down approach to redesign airspace sector while the generation of sector boundary was based on Voronoi diagrams. With the Voronoi diagram, the convexity requirement is automatically satisfied. The success of this approach depends heavily on the location of Voronoi generating points. For example, in airspace of 200km*200km, if the alternative Voronoi generating points are distributed every 10 km, there will be 400 points. In the case of dividing the airspace into five sectors, the number of alternative locations at five production points is 400^5. If this approach is applied to larger airspace and divided into more sectors, the computation time may be unacceptable. When estimating the coordinating task load, Xue [8] pointed out that the boundary-crossing of air traffic control center was neglected in his work. Obviously, this will lead to unbalanced sector task load. If the sector generated by the algorithm is applied to the actual sector, the actual workload of sector near the studying airspace boundary will be larger than the result of calculation. In addition, if all sectors are Voronoi diagrams, in some cases, it is difficult to find a suitable seed point for VC generation to balance workload and meet other constraints.

Delahaye, et al. [35] have proposed the use of floating-point chromosomes, and have developed special operators to maintain the integrity of the chromosome for use in a floating point representation, contending that the convergence rate is faster in the floating-point case. When the studying space is discretized by n VCs (Voronoi cells) and the number of sectors is k, the number of possible groupings will be ϑ_n^k as following.

$$\vartheta_n^k = \frac{1}{n!}\sum_{i=0}^{i=n-1}(-1)^i\left(\frac{n!}{i!\,(n-i)!}\right)(n-i)^k \qquad (1)$$

They pointed out that this kind of combination is NP-hard and that stochastic optimization is a good candidate to address it. Pawlak, et al. [36] also used this type of floating-point chromosomes to group cells by Finite-Element method. Sergeeva, et al. [37] developed a more complex floating-point chromosome for 3D reconfiguration, which includes two parts, the first part represents coordinates of sector centers and the second part contains the associated vertical extension of each sector center. But these chromosomes are hard to understand especially how they work in genetic algorithm.

Reconfiguration problem is well-known for NP-hard problem [35, 38, 39]. Simulated annealing (SA) is a well-studied local search and meta-heuristic used to address discrete and, to a lesser extent, continuous optimization problems. It is a probabilistic technique for approximating the global optimum of a given function. Specifically, it is a meta- heuristic to approximate global optimization in a large search space. It is often used when the search space is discrete. The key feature of simulated annealing is that it provides a mechanism to escape local optima by allowing hill-climbing moves (i.e., moves which worsen the objective function value) in hopes of finding a global optimum [40]. Johnson, et al. [41], [42] is the pioneer who uses SA to graph coloring and number partitioning. Rahimian, et al. [43] proposed a fully distributed algorithm called JA-BE-JA which uses local search and simulated annealing techniques for two types of graph partitioning: edge-cut partitioning and vertex-cut partitioning. These partitioning algorithms have some similar objective,

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 July 2019 doi:10.20944/preprints201907.0287.v1

such as minimizing cut cost and balancing weights of vertexes. But they still are different from AS which require balancing total workload (monitoring and coordinating task loads), minimizing coordination task load, as well thinking about dwell time, convex shape, connectivity and compactness.

In summary, the bottom-up method is prone to appear C-shaped, zigzag, trapezoidal and other irregular shapes, while the top-down method is prone to appear narrow triangle, which is not easy to ensure convexity. In addition, the method of random Voronoi diagram is like looking for a needle in the haystack to get an optimal solution. It is doubtful that the multi-layer random Voronoi graph without boundary reprocessing not only guarantees the workload balance of the graph, but also guarantees the convexity of the sector. Therefore, we must design an innovative solution to the AS problem.

The aim of this paper is to establish a framework of solving methods including a series of multi-objective (balance task load, Minimize task load imbalance, Minimize total coordinating task load, Minimize total coordinating task load, Minimize cost of short dwell time, Minimize cost of reenter the same sector) solving methods that can perform AS automatically at strategically planning stage, while meeting the constraints of connectivity and compactness in optimal process, as well to smooth boundary for accessible shape of sector. In order to improve the reliability of the design, we synthesized the region growth method and the graph cutting method. Firstly, we use the region growth method based on Voronoi diagram to generate the initial optimal solution. Then, two graph cutting algorithms are used to eliminate irregular sector boundary caused by region growing algorithm.

This paper makes the following specific contributions:

1. Both of two graph cutting algorithms proposed by us can eliminate the irregular sector boundary caused by region growing algorithm as well as other criteria can also be guaranteed.
2. MC-CLFV algorithm based on flexible vertices generated deducted from SAA result greatly simplifies the complexity of graphic cutting algorithm.
3. MC-RC algorithm shows some advantages to reduce narrow blocks while MC-CLFV algorithm is superior to MC-RC algorithm for increasing traffic throughput.
4. We also found that appropriately allowing some non-convex boundaries is a measure to reduce the total workload, and blindly pursuing convex boundaries will increase the total workload.
5. The solution framework proposed in this paper improves the reliability of obtaining the optimal design scheme in airspace, and will reduce the design load and completion time of designers.

These studies are of great significance to improve air traffic throughput and maintain the task load of controllers at a reasonable level. This paper consists of five sections. Beginning with the introduction, section 2 describes problem formulation. Section 3 elaborates the details of solution method which including SAA and two post-processing method. Experimental results for a realistic scenario are reported in section 4. Section 5 presents discussions. Conclusions and opportunities for further work are presented in section 6.

2. Problem formulation

2.1. Boundary and route network

The boundary points are expressed as B. B is a $n \times 2$ matrix, and is listed anti-clockwise. The boundary line is the red dotted line in Figure 2. Each row of a matrix B represents geographic coordinate of a point.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 July 2019 doi:10.20944/preprints201907.0287.v1

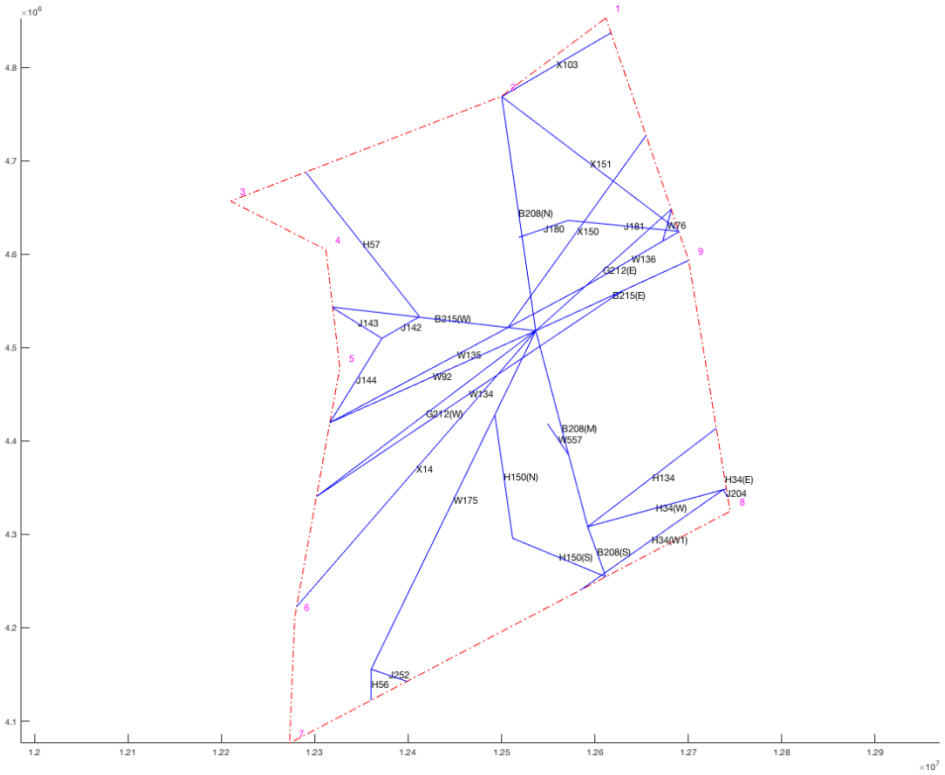

Figure 2. Route network and boundary in TYN airspace

Route network consists of all segments (blue line in Figure 2.) in airspace. Table1 is an example of the route network shown in Figure 2. Due to the paper length limit, we listed only a part of data. Despite of having the same name, the segment is distinguished by route course, such as B208 (N), B208 (M), and B208(S). Each segment includes the coordinates of the starting and ending points and the daily traffic volume.

Table 1.Route network and daily traffic volume

Route	Starting point coordinates (φ_1, λ_1)		End-point coordinates (φ_2, λ_2)		Starting point	End point	Daily traffic volume f (flights/day)
B208(N)	39 29.5	112 12.1	37 45.0	112 37.1	TODAM	TYN	146.1
B215(E)	37 45.0	112 37.1	38 17.3	114 05.9	TYN	ISGOD	90.3
G212(W)	37 45.0	112 37.1	36 28.4	110 30.6	TYN	OKVUM	17.2
H134	36 59.4	114 19.7	36 14.3	113 07.1	P346+1	SQ	2.0
…	…	…	…	…	…	…	…

2.2. Voronoi diagrams generating method

VCs are adopted to discretize the reconfiguration airspace in this paper. Given some generating points (Seeds of Voronoi diagram), the Voronoi diagram divides airspace into a group of convex polygons with no overlaps. In theory, the more the number of generating points, the more the solution space. Although the small granularity can result in more balanced sectors, it will take more time to calculate. The distribution of generating points in space also affects the solution. The following three methods are used to generate seeds of Voronoi diagrams.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 July 2019 doi:10.20944/preprints201907.0287.v1

2.2.1. Seeds were generated randomly in the studying airspace

Within the square airspace surrounding the airspace boundary, random point is generated. If this point falls within the airspace boundary, a random seed is successfully generated. Repeat this step over and over again to generate n generating points. Then the Voronoi diagrams can be generated. Figure 3 is an example of a Voronoi diagram generated by 90 randomly generating points.

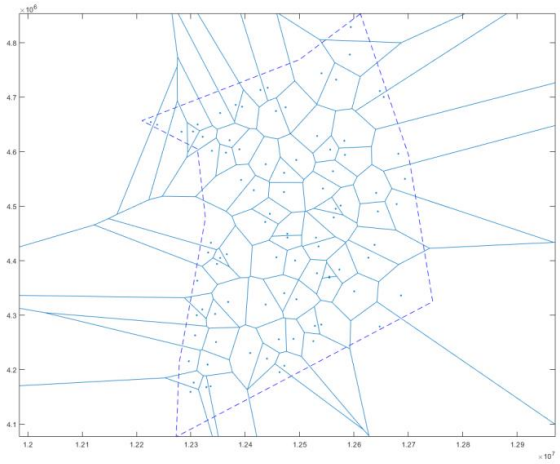

Figure 3. Random generating seeds

2.2.2. Seeds were taken every D_{min} kilometers along routes

Generating points for constructing VCs include starting and end points of every route, as well as some intermediate points along routes separated by at least D_{min} kilometers. If the length of r segment is D_r, then the number of intermediate points equal the following.

$$m_r = \left\lfloor \frac{D_r}{D_{min}} \right\rfloor - 1 \qquad (2)$$

where $\lfloor \cdot \rfloor$ is the operator of rounding down. Less D_{min} means more VCs with small granularity when discretizing the airspace. Figure 4 is a Voronoi diagram generated by taking a point every 30 kilometers along the route (Pink coloring) as generating points.

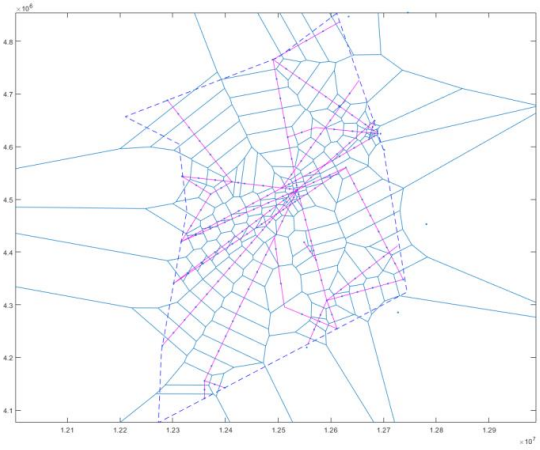

Figure 4. Seeds generated by taking a point every 30 km along route

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 July 2019 doi:10.20944/preprints201907.0287.v1

2.2.3. Hexagonal seed

The generating points are evenly and staggeringly distributed in the studying airspace. The resulting Voronoi diagram is a regular hexagon. Figure 5 is an example of a hexagonal Voronoi diagram generated in the studying airspace.

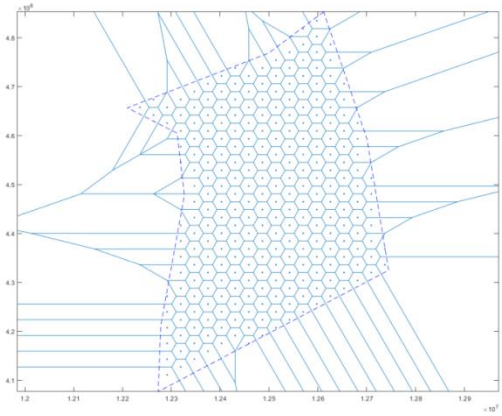

Figure 5. Hexagonal seed

2.3. Reconstruct Voronoi cells according to the airspace boundary for coloring correctly

After discretizing the airspace with Voronoi cells (VCs), in order to visualize which sector these Voronoi cells belong to, the Voronoi diagram needs to be colored. Voronoi cell coloring in the airspace boundary can be easily colored by calling fill or pitch function of MATLAB, but it is not easy to color Voronoi cells near the airspace boundary. Ordinary Voronoi diagrams have the following basic properties [44].

- All the points on the common edges have equal distances to their neighbor generating points.
- The generating point of VC is the closest generating point to any point within this VC.
- The line between two generating points is perpendicular to the common edge of the two Voronoi polygons.

These characteristics of VCs will be utilized to calculate the intersection point between edge of boundary and edge of VC.

Reconstructing VCs and coloring all cells for visualization are very important for reconfiguration. We noticed that color cannot be filled in some open VCs according to the index of which sector these cells belong to. In the process of coloring, just the inner part of the VC crossing by airspace boundary is preferred to retain. So we develop a method to reshape these VCs at next part for coloring all cells correctly.

The procedure for the solution of the intersection point between edge of boundary and edge of VC is defined as follow.

Firstly, all endpoints of the boundary line were sorted anticlockwise. The equation of each boundary line is derived from Table 1. The equation of a boundary line can be expressed as equation 3. According to the characteristic that all the points on the common edges have equal distances to their neighbor generating points, the common edges must be perpendicular to the line between two generating points which generate VC that passed through by boundary segment. As a result, we can get the function of common edges as equation 4. The intersection point can be obtained from two of these linear equations.

$$y = k_1 x + b_1 \quad (3)$$

$$y = k_2 x + b_2 \quad (4)$$

Thereafter, only points within airspace boundary of VC which is intersected by boundary segment and intersection point are retained as the new vertices set of these VC. After anti-clockwise sorting of these new

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 July 2019 doi:10.20944/preprints201907.0287.v1

vertices sets of these VC, every VC can be correctly colored by program. Figure 6 is an example that Voronoi cell diagram is reconstructed and colored according to airspace boundary. With this reconstruction and coloring, it is not only easy to distinguish which sector VCs belong to, but also easy to solve the intersection of route and boundary VCs.

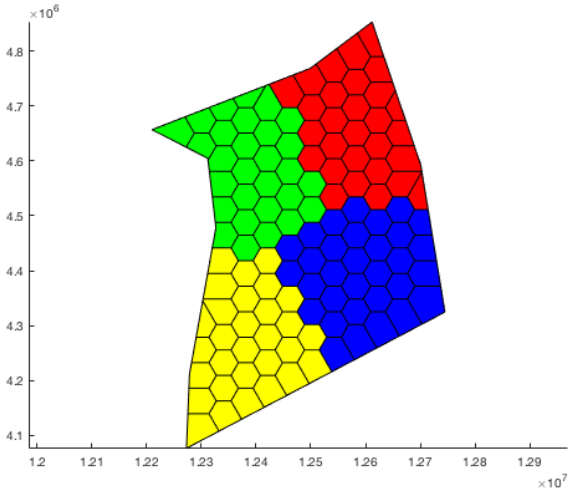

Figure 6. Voronoi cells are reconstructed and colored according to airspace boundary

2.4. Task load measurement and objective function

The research focusing on controller workload has been carried out by a desire to balance task load, understand occupational stress, eliminate operational errors, enhance safety, and improve throughput. Many workload calculation methods have been proposed in previous studies based on dependent variables: physical activity[45], physiological indicators [46-48], simulation models of the controller's tasks[34, 35, 49-52], and subjective ratings [53, 54]. Gianazza [55] have reviewed literature about workload prediction and compared several machine learning methods on the problem of learning workload prediction models from historical data. Surprisingly, there is no globally accepted definition for controller workload – "controller workload is a confusing term and with a multitude of definitions, its measurement is not uniform"[56]. From the perspective of technical feasibility, the task load calculation method based on statistical controller's tasks in field or simulated tests is easier than others.

Each sector is typically positioned with one or two controllers. A radar controller is responsible for radio communications with aircraft, monitoring the radar screen to maintain safe separation, and communicating with other controllers. When two controllers work a sector, the second is an associate controller, known as a data-controller. The data-controller typically receives flight-plan information and helps plan and organize the flow of traffic within the sector. During exceptionally busy periods, a third controller may be assigned to the team, although three-member teams are not typically planned for. Task load is the sum of the time spent by controllers performing tasks[57].

Task load calculation model designed by Oktal and Yaman [52] considered three kinds of task load, the monitoring task load, the conflict task load and the coordination task load. The monitoring and the conflict task loads occur inside the sector while the coordination task load between the sector and its adjacent sector. Due to the fact that the air traffic controllers in China usually build an aerial overpass bridge at the intersection of the route to insure that flights which may be conflict have been transferred to the specified flight levels before they arrive the intersection, it indicates that the monitoring task load includes the conflict task load. For this reason, our task load calculation model only considers monitoring and coordinating task loads. Formulas of our task load calculation model is similar to these designed by [35,

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 July 2019 doi:10.20944/preprints201907.0287.v1

37], while the tiny difference lies in that we consider the duration of the task, not just the number of flights in our task load model. We suppose that the monitoring task load of one aircraft is proportional to flight time. This ratio is the average time per unit of time that the aircraft needs to be monitored by the controller (including conflict resolution by telephone). The coordinated task load of an aircraft is equal to the average time for the controller to complete takeover or handover when this aircraft cross sector boundaries.

2.4.1. TASK LOAD MEASUREMENT

Given that the airspace studied has n VCs, each $VC_i, i \in \{1, \dots, n\}$, must be assigned to some sector $s_j, j \in \{1, \dots, k\}$ where k is the number of desired sectors to be opened. If VC_i is assigned to s_j sector, then $x_i^j = 1$, else 0. The relationship between task load of jth sector and task load of VC_i is expressed as follow:

$$wl^j = \sum_{i=1}^{n} x_i^j \times wl_i, \forall j \in \{1, \dots, k\}, \ \forall i \in \{1, \dots, n\} \quad (5)$$

wl_i includes two kind of task loads, namely monitoring and coordinating task loads, as mentioned before.

$$wl_i = wl_i^{mot} + wl_i^{cod} \quad (6)$$

2.4.1.1. *Monitoring task load measurement*

Assume that there are m routes passing through VC_i. Then monitoring task load was calculated as follow.

$$wl_i^{mot} = \alpha_{mot} \times \sum_{r=1}^{m} f_r \times (l_i^r / v_r) \quad (7)$$

l_i^r is length of route r in VC_i, km.

v_r is average speed of flights on route r, km/hour.

f_r is hourly traffic volume passing through route r, number/hour.

α_{mot} is the statistical monitoring time per hour per flight by field test, seconds/hour.

The intersection of all routes with the VC they pass through is computed and stored in a structured array before the task load is computed, as shown in Figure 7.

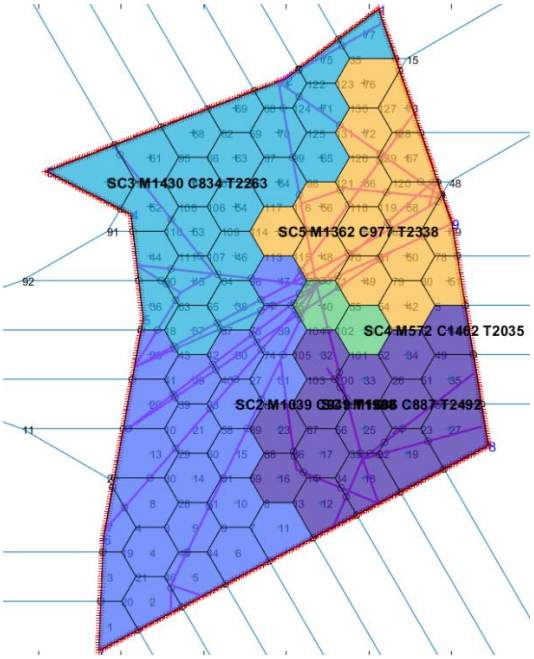

Figure 7. Intersection of all routes with the VC they pass through

2.4.1.2. *Coordinating task load measurement*

Coordinating task load calculated as follow:

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 July 2019 doi:10.20944/preprints201907.0287.v1

If $\forall r \in \{1, \dots, m\}$ passing through VC i and both of the boundaries of VC were the sector boundary, then

$$wl_i^{cod} = \alpha_{cod} \times \sum_{r=1}^{m} (\gamma_r^i \times f_r) \quad (8)$$

f_r is hourly traffic volume passing through route r, number/hour.

α_{cod} is the statistical coordinating time per hour per flight by field test, seconds/hour.

If $\forall r \in \{1, \dots, m\}$ passing through VC_i and two of the boundaries of VC_i was the sector boundary, then γ_r^i=2. This rarely happens unless a single VORONOI diagram is large. If $\forall r \in \{1, \dots, m\}$ passing through VC_i and just one of the boundaries of VC_i was the sector boundary, then γ_r^i=1. If $\forall r \in$ passing through VC i and none of the boundary of VC was the sector boundary, then γ_r^i=0.

2.4.2. OBJECTIVE FUNCTION

Four criteria have been included in our objective function for evaluation of a solution. The formula of objective function is:

$$F = (\alpha_1 * cost_{imb} + \alpha_2 * wl_{tot}^{cod} + \alpha_3 * cost_{tot}^{SDT} + \alpha_4 * cost_{tot}^{rin}) \quad (9)$$

$\alpha_1, \alpha_2, \alpha_3, \alpha_4$ representing the weight of different objective components.

2.4.2.1. Minimize task load imbalance

The first criterion minimizes the level of the task load imbalance among all controlled sectors in configuration. The task load imbalance of all controlled sectors is then computed with the following formula:

$$cost_{imb} = \sum_{j=1}^{k} |wl^j - \mu| \quad (10)$$

Where:

k: Number of desired sectors to be opened

j: jth sector

μ: Average task load, $\mu = \frac{\sum_{j=1}^{k} wl^j}{k}$

wl^j: Total task load in jth sector

2.4.2.2. Minimize total coordinating task load

Coordinating task load is the key factor of total task load. In the process of airspace sectorization , it is almost impossible to decrease monitoring task load. A good airspace configuration plan must be a plan with the minimum coordination task load. Minimizing coordination task load can not only reduce the total task load, but also keep the sector consistent with the main traffic flow. The total coordinating task load, wl_{tot}^{cod}, is equal to the sum of task load of all VCs as following formula.

$$wl_{tot}^{cod} = \sum_{i=1}^{n} wl_i^{cod}, \ i \in \{1, \dots, n\} \quad (11)$$

2.4.2.3. Minimize cost of short dwell time (SDT)

The sector capacity is calculate based on MAP from FAA [58], which is roughly 5/3 of average sector flight time (in minutes). This means that sector capacity increases with dwell time (average sector flight time). Because of the airspace studied is an en-route airspace, we arbitrarily set the minimum of dwell time as 240 seconds. The cost of short dwell time was measured as follow:

$$cost_{tot}^{SDT} = \sum_{j=1}^{k} cost_j^{SDT} \quad (12)$$

For any flight $a \in \{1, \dots, m\}$ in sector j,

$$cost_j^{SDT} = \sum_{p=1}^{m} \{[t_{dw}^{min} - t_{dw}(a,j)] \cdot \varepsilon_{dw}\} \quad (13)$$

$$t_{dw}(p,j) = t_{out}(a,j) - t_{in}(a,j) \quad (14)$$

$t_{dw}(a,j)$ is the pth flight's dwell time $t_{dw}(a,j)$ in the jth sector, t_{dw}^{min} was set as 240 seconds.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 July 2019 doi:10.20944/preprints201907.0287.v1

ε_{dw} is penalty coefficient of short dwell time,

$$\varepsilon_{dw}=\begin{cases}\sqrt[6]{exp\left[t_{dw}^{min}-t_{dw}(a,j)\right]}, & \left[t_{dw}^{min}-t_{dw}(a,j)\right]>0\\ 0, & \left[t_{dw}^{min}-t_{dw}(a,j)\right]\le 0\end{cases} \quad (15)$$

2.4.2.4. Minimize cost of reenter the same sector

We hope that every trajectory enters the same sector less than one time. To some extent, it means this cost will cause the optimization search process to discard those non-convex sectors and disconnected sectors.

$$cost_{tot}^{rin}=\sum_{r=1}^{m} cost_r^{rin} \quad (16)$$

$$cost_r^{rin}=\sum_{j=1}^{k}\beta\times t_{rin\_j} \quad (17)$$

If r route doesn't pass through j sector, then t_{rin_j}=0; If r route passes through jth sector more than once, cost of r route reenter in j sector will increase with reenter times. β is the reenter penalty factor. We arbitrarily set it as 100 seconds.

2.5. Constraints

Three constraints, task load, connectivity and compactness are considered.

2.5.1. Task load constraint

In theory, the highest value for R-side task load for a 15-minute period is 900 seconds (15 minutes times 60 seconds/minute), assuming (unrealistically) that a controller can effectively use all 900 seconds of available time and that a second controller is handling the D-side task load. When the task load rollup exceeds a certain threshold (around 600 seconds), it is assumed that two controllers are working the traffic[57]. The maximum acceptable task load in each sector is set as 3420 seconds for a 1-hour period. Assuming that there is only one sector in the studying airspace at first, the task load increases gradually with time. Assuming that there is only one sector in the research airspace at first, the task load will gradually increase with time. When the task load increases to its maximum, the partition of the airspace is triggered. In order to balance the workload, the best partitioning result is that the workload of each sector is half of the maximum workload. Over time, when the task load of both sectors reaches the maximum task load, the best division is that the task load of each sector is 2/3 of the maximum task load. It can be inferred that when airspace is divided into k sectors, the optimal average task load can be expressed as following.

$$\mu=\frac{k-1}{k}\times wl_{max} \quad (18)$$

When the airspace keeps k sectors unchanged, and the traffic flow of different sectors is not consistent with the growth of time, the sector structure also needs to be adjusted. Before changing the number of sectors from k to k+1, the variation range of average task load is as follows.

$$\frac{k-1}{k}\times wl_{max}\le\mu\le wl_{max} \quad (19)$$

Equation (18) and (19) are based on the trigger mechanism of sector division, which is when the traffic flow of all sectors increases to wl_{max}. The real environment may be that the traffic flow growth in each sector is not balanced, some sectors may not grow, and some sectors have reached the maximum. It also needs to be adjusted at this time. The more sectors there are, the more likely the growth imbalance will be. Then the trigger mechanism will fail. In the case of unbalanced growth of traffic flow in each sector, it is necessary to adjust the sector structure as long as a single sector reaches its maximum. In this case, the task load of the busiest sector can usually be reduced by adjusting the structure of airspace without increasing the number of departments. Assuming this happens, there are currently k sectors, the total workload is wl_{tot}, and the average workload is wl_{tot}/k. Then, the following equation is used to constrain the task load of each sector.

$$\beta_1\times wl_{tot}/k\le wl_i\le\beta_2\times wl_{max} \quad (20)$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 July 2019 doi:10.20944/preprints201907.0287.v1

β_1 is the minimum deviation from the average of total task load. The smaller the value of β_1 is, the more feasible solutions are. β_2 is the design redundancy parameter. If the design cycle is longer, the β_2 should be smaller, leaving enough room for traffic flow growth.

2.5.2. Connectivity constraint

The connectivity constraint ensures that the sectors are not fragmented[16]. In other word, a sector must be a contiguous portion of airspace. Connectivity of a sector was assured by checking whether every VC in this sector has at least one common edge with other VC or not. If each VC in certain sector has no common edge with other VC, then this sector with this VC is not connective.

Definition: Undirected graph $\boldsymbol{D} = (\boldsymbol{VC}, \boldsymbol{E})$, VC set $\boldsymbol{VC}$={VC_1, VC_2,..., VC_n} belong to sector s_i, if two Voronoi cell have common edge between each other, then the two cells are adjacent. Define the adjacency matrix $\boldsymbol{M}$=$(M_{ij})_{\mathrm{n\times n}}$ as follow:

$$M_{ij} = \begin{cases} 0\,, VC_i \ to \ VC_j \ is \ not \ \text{adjacent} \\ 1\,, \quad VC_i \ to \ VC_j \ is \ \text{adjacent} \end{cases}$$

Similarly, we define the reachable matrix $\boldsymbol{R}$=$(R_{ij})_{\mathrm{n\times n}}$ of graph $\boldsymbol{D}$ as follow:

$$R_{ij} = \begin{cases} 0\,, VC_i \ to \ VC_j \ is \ not \ \text{reachable} \\ 1\,, \quad VC_i \ to \ VC_j \ is \ \text{reachable} \end{cases}$$

$\boldsymbol{R}$ can be calculated by adjacency matrix $\boldsymbol{M}$ [59, 60].

If the sum of every column of R equals n, then all VCs in sector s_i are connected, otherwise, the conclusion is contrary.

Two methods are used to guarantee connectivity. One is to reject the new solution directly, keep the original solution and search on the basis of it. The other is to repair the disconnected sector into connected sector through the following methods. According to the adjacency matrix of a sector, numbers of connected areas were calculated in this sector. If there are more than two connected areas in any sector, only the area with the largest number of VCs are reserved, and other isolated area are allocated to some sector with the smallest task load and this sector must be adjacent to these isolated areas. When the simulated annealing temperature is higher, connectivity of sectors will be repaired with a larger probability. On the contrary, when the temperature is low, connectivity of sectors will be repaired with a small probability. While these disconnected solutions were rejected with a larger probability.

2.5.3. Compactness constraint

In the process of using neighborhood search method to generate new solutions, there are a few VCs very far from their assigned sector. This is called non-compactness. Compactness of a sector is defined as follows.

$$\delta_j = \sum_{i=1}^{n} D_i^j \cdot z_i^j / k \quad \forall i \in \{1, \dots, n\}, \forall j \in \{1, \dots, k\} \qquad (21)$$

D_i^j is the distance between generating point of VC_i, and the center of s_j. If VC_i located in sector s_j, z_i^j = 1,otherwise, $z_i^j = 0$. δ_j is the average distance of all VC to the center of s_j. The average distance is set as threshold to assure compactness of sectors.

Firstly, all VCs which distance between μ_j and themselves were larger than the threshold value are found. Then, the numbers of connected regions were found out in the undirected graph composing of these cells. Each of them is allocated unconditionally to the nearest sector.

2.5.4. Number constraint of flexible vertices

The vertices of a sector boundary can be classified according to their sharing by multiple sectors. All vertices of a sector boundary are divided into two categories: flexible vertices and fixed vertices. The classification of vertices is illustrated in Figure 8.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 July 2019 doi:10.20944/preprints201907.0287.v1

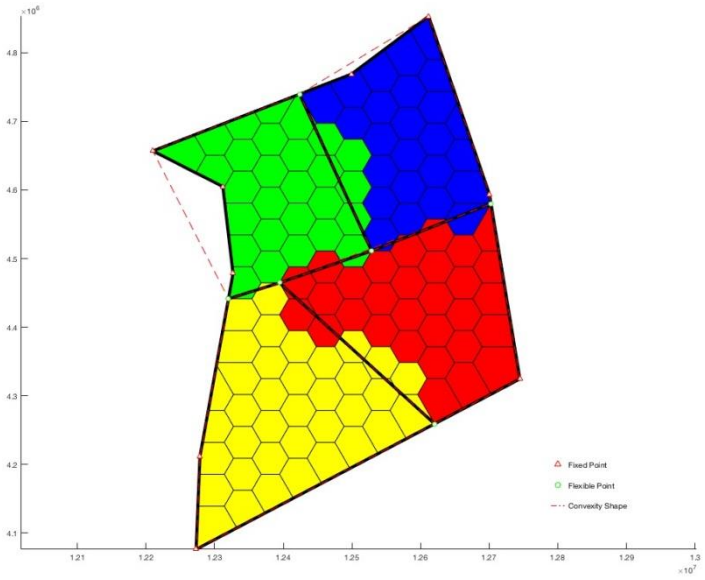

Figure 8. Flexible and fixed vertices

The blue-coated sector has a vertex shared (two-zone shared vertex) with the red-coated sector, as well as with the green-coated sector. There is a vertex shared by the blue-coated sector, the red-coated sector and the green -coated sector. Both of the two-zone shared vertex and three-zone shared vertex are called as flexible vertices. The other vertices marked with a red triangle in the blue coated sector in Figure 8 are fixed. We find that when the number of flexible vertices (NFV) in a sector is less than three, the sector shape generated by SAA will inevitably appear C shape. So we have to force the NFV to be greater than three. This constraint is not used for the SAA search process, but only for the sector evaluation and post-processing. If the number of flexible points in the sector generated by SAA is less than three, the SAA process is repeated.

3. Solution method

There are three common design questions related to meta-heuristics: the representation of solutions handled by the algorithm, new solution generation method and the definition of the objective function that will guide the search. The definition of objective function has been discussed in the previous section. In this section, we will introduce the representations of solutions and solution method of airspace reconstruction problem. The framework of the solution method is showed in Figure 9.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 July 2019 doi:10.20944/preprints201907.0287.v1

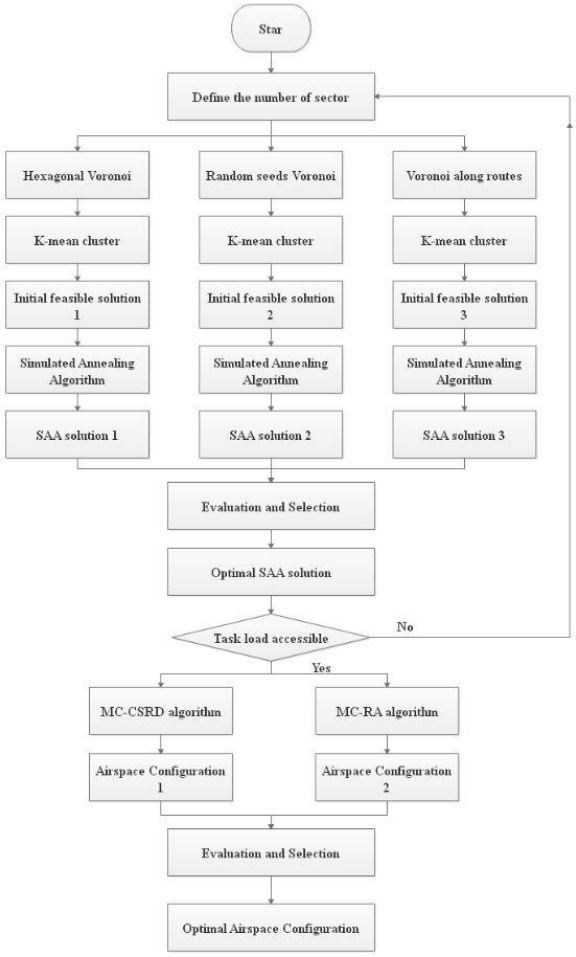

Figure 9. Framework of the solution method

3.1. Solutions expressions

The representation is based on the proposed reconfiguration problem modeling. When calculating task load, the representation of solution is similar to assignment problem. If VC_i, $\forall i \in \{1, \dots, n\}$, is assigned to sector $s_j, j \in \{1, \dots, k\}$, then $z_i^j = 1$, otherwise $z_i^j = 0$. Here, an example is given to show the representation of this solution. Suppose there are 10 cells allocated to 3 sectors.

Table 2.The first type of representation of solution z

VC	1	2	3	4	5	6	7	8	9	10
Sector 1	1	1	1	0	0	0	0	0	0	0
Sector 2	0	0	0	1	1	1	0	0	0	0
Sector 3	0	0	0	0	0	0	1	1	1	1

The first type of representation of solution $\boldsymbol{z} = \begin{Bmatrix} z_1^1, z_2^1, \dots z_n^1 \\ z_1^2, z_2^2, \dots z_n^2 \\ z_1^3, z_2^3, \dots z_n^3 \end{Bmatrix}$ was shown in Table 2. If VCs (1,2,3) are adjacent, as well (4,5,6) and (7,8,9,10), then the solution show in Table 2 is a feasible one. This type of representation of solution was used to calculate task load.

Table 3.The second type of representation of solution x

VC	1	2	3	4	5	6	7	8	9	10
Sector	1	1	1	2	2	2	3	3	3	3

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 July 2019 doi:10.20944/preprints201907.0287.v1

The second type of representation of solution $\boldsymbol{x} = \{x_1, x_2, \ldots, x_n\}$ was show in Table 3. This type of representation of solution was used to renew current solution.

Two kinds of these representations can be transformed into each other.

3.2. K-means clustering to generate initial feasible solution

Given a set of observations (p_i is coordinate of generation point of VC_i, i=1,2..n), k-means clustering [61] aims to partition the n observations into k ($\leq n$) sets $\boldsymbol{s} = \{s_1, s_2, \ldots, s_k\}$ so as to minimize the within-cluster sum of squares (i.e. variance). Formally, the objective is to find:

$$\arg\min_{\boldsymbol{s}} \sum_{j=1}^{k} \sum_{p \in s_j} ||p_i - \mu_j||^2 \qquad (22)$$

where μ_j is the centroid of points in s_j.

After k-means clustering, the sector of VC_i will become clear. The second type of representation of solution is used to represent k-means clustering results. For example, x= [1,1,1,2,2,3, 3], means VC 1,2,3 belong to sector 1, VC 4,5 belong to sector 2, VC 6,7 belong to sector 3.

As mentioned in Section 2.2 above, the solution space of different discretization methods (even the number of VCs) in the studied airspace will be different. Therefore, we design three parallel approaches to explore the optimal solution. Finally, the best solution is chosen by comparison. This process is shown in Figure 9.

3.3. Neighborhood search strategy

3.3.1. Variable neighborhood search

To prevent local optimization, the following two variable neighborhood search methods are used in the search process.

3.3.1.1. RC_PNB search algorithm

Firstly, we need to calculate the workload of all sectors and find out the sectors with the minimum task load. Assuming that the sector with the minimum task load is the sector s_j, and VC_{i1}, $VC_{i2}, VC_{i3} \ldots VC_{iA}$ (All Voronoi cells with crimson color of dark blue sector with the smallest task load as showed in Figure 10) adjacent to at least one of the VC in the sector s_j, we define set $\{VC_{i1}, VC_{i2}, VC_{i3} \ldots VC_{iA}\}$ as PNB_{s_j}, propagable neighborhood of s_j. These VC of PNB_{s_j} may be distributed in one sector or multiple sectors.

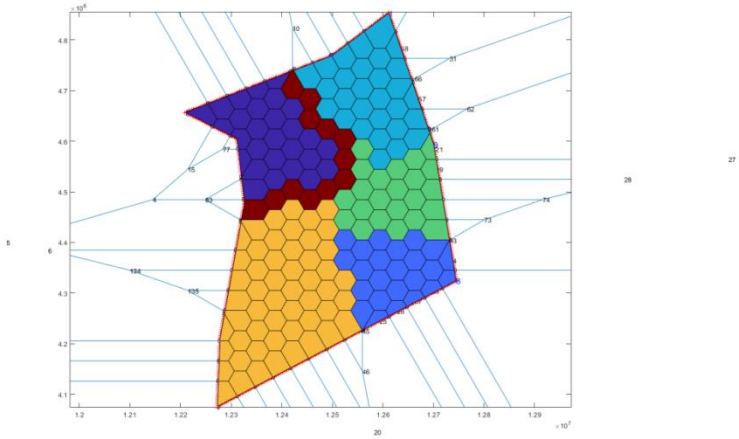

Figure 10. An example of propagable neighborhood (Crimson) of sector (Dark blue) with the smallest task load

A variable neighborhood search algorithm is used according to a certain probability as following.

RC_PNB search algorithm

$x = x_0$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 July 2019 doi:10.20944/preprints201907.0287.v1

$PNB_{s_j} = PBN\_find\_fun(x, M, sn_{tl\_min})$;
$A = length\left(PNB_{s_j}\right)$;
$B = randi(A)$;
If rand>0.5
$A_1 = A(1:B,:)$;
else
$A_1 = A(B:end,:)$;
end
$x(A_1) = sn_{tl\_min}$
Connectivity constraint check and assurance algorithm, $x \rightarrow x_1$
Compactness constraint assurance, $x_1 \rightarrow x_2$
If $F(x_2) < F(x_0)$
$x_0 = x_2$
else
$x_0 = x_0$
end

3.3.1.2. PN_VC search algorithm

All VCs adjacent to other sectors in the sector s_j with the lowest task load are called expandable boundaries points (*EBP*). A neighborhood search algorithm of expandable boundary points (*EBP_NB search algorithm*) in minimum task load sector s_j is developed as following.

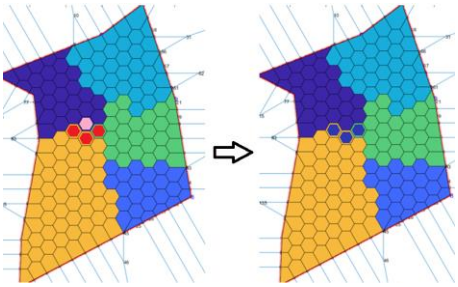

Figure 11. *PN_VC search*

As in Figure 11 left show, one VC (pink color) in *EBP* is randomly selected, all VCs around this VC but not belong to s_j (red color) are called as expandable boundary points of this VC (*EBP_VC*). A neighborhood search algorithm of propagable neighborhood of VC (*PN_VC search algorithm*) in minimum task load sector s_j is developed as following.

PN_VC search algorithm

$x = x_0$
Random select a VC in EBP_VC
Check out PN_VC of this VC
Allocate PN_VC to s_j *and update* x
Connectivity constraint check and assurance algorithm, $x \rightarrow x_1$
Compactness constraint assurance, $x_1 \rightarrow x_2$
If $F(x_2) < F(x_0)$
$x_0 = x_2$
else
$x_0 = x_0$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 July 2019 doi:10.20944/preprints201907.0287.v1

451 3.3.2. Connectivity constraint check and assurance

452 After calculating the reachable matrix R of the sector (using formula 20), if the sum of each column of R is
453 less than n, all connected components of the sector should be detected. If a sector is divided into several blocks
454 by using neighborhood search method, only the largest blocks in the sector are retained, other smaller blocks
455 are unconditionally allocated to sector connected to these components with smallest distance.

456 3.3.3. Compactness constraint assurance

457 In order to ensure the compactness of the sector, the average distance from the generating points of a VCs
458 to the center point of all VCs was set as threshold to assure compactness of sectors. If some VCs were far from
459 their assigned sector more than a threshold in equation (21), they would be allocated to the adjacent sector with
460 the smallest distance; otherwise, they would be kept in original sector.

461 **3.4. Simulated annealing algorithm**

462 Simulated annealing starts from a state x_0 and continues until temperature T cool down to T_{min} at a
463 cooling rate σ. In the process, the call neighbor ($\boldsymbol{x}$) should generate a randomly chosen neighbour of a given
464 state x; to increase search space, repeat call neighbor ($\boldsymbol{x}$) L times at a specific temperature. The call random (0,
465 1) should pick and return a value in the range [0, 1], uniformly at random. The algorithm determines whether
466 the new solution x_{new} is better or worse than the current solution x. If the new solution is better than the
467 current solution, it becomes the next solution. If the new solution is worse than the current solution, the
468 algorithm can still make it the next solution. The algorithm accepts a worse solution based on an acceptance
469 function $1/(1+ e^{\frac{\delta}{T}})$. Since both $\delta=E(\boldsymbol{x_{new}})-E(\boldsymbol{x})$ and T are positive, the probability of acceptance is between 0 and
470 1. Lower temperature leads to smaller acceptance probability. At the same time, larger δ leads to smaller
471 acceptance probability. The best solution and value of objective function were recorded at every specific
472 temperature. After the simulated annealing, the corresponding solution of the minimum value of all objective
473 function at different temperature was extracted as the optimal solution. The following pseudo code presents
474 the simulated annealing heuristic as described above:

Simulated annealing algorithm

Let $x= x_0$
n=length(x_0)
$T = T_0$
m=100, % m is the iteration times at different temperatures
Partition=[ ];
While $T> T_{min}$
    temp=zeros(m,n+1);
    for k=1:m
    If rand(1)>0.6
        RC_PNB search algorithm, $x_{new}\leftarrow$ *neighbour*(x)
    else
        PN_VC search algorithm, $x_{new}\leftarrow$ *neighbour*(x)
    end
    If $F(x_{new})< F(x_0)$
        $x\leftarrow x_{new}$
    Elseif $1/(1+ e^{\frac{E(x_{new})- E(x)}{T}})\geq$ *random(0, 1):*
        $x\leftarrow x_{new}$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 July 2019 doi:10.20944/preprints201907.0287.v1

End

temp(k,:)=[x, $F(x)$];

end

[E_{min},index]=min(temp(:,end)); %Find the optimal partition at current temperature.

Partition=[partition; temp(index,end)];

$T = T \times \sigma$; here σ is cooling rate

End

[F_{opt},index1]=min(Partition (:,end));

x_{opt}= Partition (index1,1:end-1); %the final state is x_{opt}, optimal value of objective function is E_{opt}

3.5. Post-processing

After solving the model by SAA, the boundaries of each sector are obtained. Even considering the compactness of the objective function, sector boundaries generated by SAA, like all bottom-up methods, still appear as zigzag, ladder or C-shaped. Drew [29] used Douglas-Peucker (DP) algorithm to smooth shared boundaries of the sectors generated by mixed integer programming, there are still some obvious problem, such as boundaries with large zigzag shape. Li, Wang, Hwang and Hwang [17] also developed a bisection method based on the shortest path to smooth boundaries of the sectors. How these methods ensure the balance of task load among sectors is questionable, because they adjust boundaries based on distance parameters rather than task load. However, the most critical criterion for sector division is task load balancing, because only task load balancing can give each sector the opportunity (buffer) to increase traffic throughput.

Generally, a convex hull can be constructed for each pair sectors generated by SAA (or any other methods). Then, there must be a line which can bisect this convex hull into two sectors with balance task work. We find that when all these lines are drawn, these lines cannot clearly divide the studied airspace into the required number and shape of sectors. Therefore, this method has been abandoned.

To illustrate the principle of our algorithm, we first observe the boundary characteristics of blue-coated sectors from Figure 8. Taking the blue-coated sector shown in Figure 8 as an example, we call the boundary of the studied airspace as the outer boundary, and the irregular boundary between the blue-coated sector and the adjacent (red or pink) sector as the shared boundary. Our algorithm is to smooth these irregular shared boundaries, and try to ensure convexity and task load balance of all sectors, as well as connectivity and compressibility. Next, all fixed and flexible vertices (described in 2.5.4) in each sector are found out by repeating this process. If all boundary vertices are connected by a straight line, the sector that coincides with the original sector can be obtained. But in this way, the task load of the sector will be different from that of the original sector, resulting in unbalanced task load of the sector. Therefore, our idea is to reposition these flexible vertices and rebuild sector boundaries to balance workload sectors.

In order to guarantee the near-convexity of sector shape in process of rebuilding sector boundaries, we add a near-convexity evaluation function ($cost_{tot}^{cvx}$).

$$cost_{tot}^{cvx} = \sum_{i=1}^{k} C_i^{cvx} \quad (23)$$

$$C_i^{cvx} = \frac{SA_i^{cvx} - SA_i}{SA_i^{cvx}} \quad (24)$$

SA_i^{cvx} is the area of the convex hull of the vertex of the ith sector, in square meters.

SA_i is the area of the ith sector, in square meters.

C_i^{cvx} is the concave coefficient of shape for the ith sector. The smaller C_i^{cvx} is, the less concave the polygon is and the closer it is to the convex shape.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 July 2019 doi:10.20944/preprints201907.0287.v1

$cost_{tot}^{cvx}$ is the concave coefficient of shape for all sectors. The smaller $cost_{tot}^{cvx}$ is, the better the overall convexity of all sectors is. In this way, this performance parameter can be added to equation (9) to form the following new evaluation function, which can be used to evaluate the new sector generated by the random change of the position of these flexible vertices.

$$F = \alpha_1 * cost_{imb} + \alpha_2 * wl_{tot}^{cod} + \alpha_3 * cost_{tot}^{SDT} + \alpha_4 * cost_{tot}^{rin} + \alpha_5 * cost_{tot}^{cvx} \quad (25)$$

To prevent concave shape, we added a constraint that the angle between all adjacent lines (counterclockwise) connected to the three-share point should not be greater than 180 degrees. In MC-CLFV algorithm, this kind of constrain is called as concave shape checking.

In order to prevent sector boundaries from getting too close to some important points (such as large traffic flow routes, intersections, airports, etc.), the minimum distance constraint between departmental boundaries and important points is added[17]. In MC-CLFV algorithm, this kind of constrain is called as minimum distance checking.

3.5.1. MC-CLFV algorithm

As mentioned earlier, the key to smooth the boundary is to locate these flexible vertices. Therefore, the problem of sector boundary smoothing becomes the problem of locating these flexible vertices. Monte Carlo techniques can be used to ensure high quality, robust designs [62, 63]. In order to improve the search efficiency, we developed a dynamic Monte Carlo method by changing location of flexible vertices (MC-CLFV).

Before describing the algorithm, we give the following definitions.

Given all flexible vertices $(VF_1^{s_i}, VF_2^{s_i}, .. VF_{n1}^{s_i})$ belong to set $\boldsymbol{VF}$, and all fixed vertices $(VD_1^{s_i}, \ldots VD_{n2}^{s_i})$ belong to set $\boldsymbol{VD}$. $\boldsymbol{B}$ is boundary of airspace. $\boldsymbol{L}$ are air routes passing by airspace. Then the shape of airspace, $\boldsymbol{G}$, is a function of the above vectors, which can be defined as follows.

$$\boldsymbol{G}=g(\boldsymbol{VF},\boldsymbol{VD},\boldsymbol{B},\boldsymbol{L}) \quad (26)$$

$\boldsymbol{f}$ is vector of flow on air routes $\boldsymbol{L}$. Then the objective function (25) under the specific sector shape is the function of traffic flow and shape, which can be expressed as following.

$$F=F(\boldsymbol{G},\boldsymbol{L}) \quad (27)$$

Γ is the standard deviation of task load for all sectors. F^0 is the value of evaluation formula (27) .

r_0 is base search radius which is equal to half of the average distance between all flexible vertices. *In* search process of MC-CLFV algorithm, a random location point around each flexible vertex within the radius is used to replace the location of these flexible vertexes in $\boldsymbol{VD}$. Then $\boldsymbol{G}$ and F change with locations of these flexible vertexes.

The search process of MC-CLFV algorithm is as following.

MC-CLFV algorithm

Check out all flexible vertices $VF_1^{s_i}, VF_2^{s_i}, .. VF_{n1}^{s_i}$ *and all fixed vertices* $VD_1^{s_i}, \ldots VD_{n2}^{s_i}$ *in* $s_i, i = 1..k$, *sorted them counter-clockwise*

Setting base search radius, r_0, *and its decrease rate in every iteration*, σ^1=*0.96*

Calculate the value of the evaluation function, F^0;*Calculate initial the standard deviation* Γ_0

While $\Gamma_0 > \tau$ (τ *is maximum acceptable standard deviation*)

$r = r_0 \times \sigma^1$

For i=1:length(VF)

θ=2π× rand()

x($NVF_i^{s_i}$) =x($NVF_i^{s_i}$)+ r×rand()×cos θ;

y($NVF_i^{s_i}$) =y($NVF_i^{s_i}$)+ r×rand()×sin θ;

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 July 2019 doi:10.20944/preprints201907.0287.v1

end

VD *and renewed* ***VF****, constitute a new shape of sector.*

If concave shape checking or minimum distance checking is not true

Continue;

end

$\boldsymbol{G}=g(VF,VD,\boldsymbol{B},\boldsymbol{L})$

Calculate $F=F(\boldsymbol{G},\boldsymbol{L})$

$\Delta F=F^0-F_{opt}$

$$\Gamma=\sqrt{\frac{1}{k}\sum_{j=1}^{k}(F_j-\overline{F})^2}$$

If $\Delta F\le 0$ *and* $\Gamma\le\Gamma_0$ *and* $\beta_1\times mean(F_j,j=1\dots k)\le min(F_j,j=1\dots k)$ *and* $max(F_j,j=1\dots k)\le \beta_2\times wl_{max}$

$F_{opt}=F^0$

$\Gamma_0=\Gamma$

$\boldsymbol{G_{opt}}=\boldsymbol{G}$

end

end

3.5.2. MC-RC algorithm

When some sector produced by SAA is surrounded by other sectors (hub-sector), MC-CLFV algorithm cannot partition airspace properly. Especially, when traffic flow is crossed in the center of the airspace, or when an important airport is located in the airspace Center. Basu, Mitchell and Sabhnani [31] also developed a 'Pie Cutting ' method for more flexibility during sectorisation. But they did not specify what airspace the pie-cut method would apply to. Traffic flows and routes in different airspace have different characteristics, so the shape or partition method of sector should be different. The concept of generic sectors proposed by the Federal Aviation Administration (FAA) supports this view. Generic sectors are defined as segments of airspace in Air Route Traffic Control Centers (ARTCCs) that controllers could manage without significant specialized training or experience, beyond what they would normally acquire to become certified[64, 65]. The following Monte Carlo method by radius changing (MC-RC) is developed for this type of situation.

The sector whose geometric center is closest to the geometric center of the airspace is chosen as the hub-sector. A circle, which center is the geometric center of hub-sector and radius is the average distance between center and points consisting of convex hull of hub-sector, is constructed. By adjusting the radius, the task load of the circular region can be equal to that of the original sector.

There is an overlapping region between two convex hulls of adjacent sectors. A line, which passes through center of hub-sector and center of the overlapping region, is called as spoke. The spoke will divide the overlapping areas into two adjacent sectors. In theory, near the overlapping area of convex hull, there must be a spoke that can divide task load of two sectors equally. Such a spoke can be found by keeping the starting point of spoke (center of hub-sector) unchanged and only changing the bearing of spoke from center. Figure 12 show initializing hub and spoke.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 July 2019 doi:10.20944/preprints201907.0287.v1

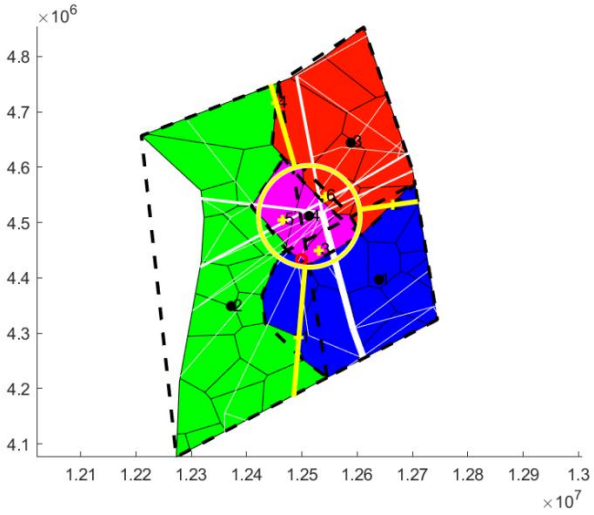

Figure 12. Initializing hub and spoke

The search process of MC-RC algorithm is as following.

Configuration $\boldsymbol{G}$ is determined by dynamic variables r_h, $\boldsymbol{\theta} = (\theta_1, \theta_2 .. \theta_{k-1})$ and static variables $\boldsymbol{B}$, $\boldsymbol{L}$. k is the number of sectors. Graph can be expressed as $\boldsymbol{G} = g(r_h, \boldsymbol{\theta}, \boldsymbol{B}, \boldsymbol{L})$. r_h is radius of hub sector. $\boldsymbol{\theta}$ is vector consisting of three bearing of spoke. $\boldsymbol{B}$ is boundary of airspace. $\boldsymbol{L}$ are air routes passing by airspace. Then value of objective function in formula (9) can be expressed as $\boldsymbol{F} = F(\boldsymbol{G}, \boldsymbol{f})$. $\boldsymbol{f}$ is vector of flow on air routes $\boldsymbol{L}$. Given that the value of objective function of every sector is F_j, $j = 1,2,..k$, under configuration $\boldsymbol{G}$ and traffic flow $\boldsymbol{f}$. Then $F^{ave}$is the average of value of task load of all sectors. Given that F^h is the value of task load of hub sector, then the task load difference between F^h and F^{ave} can be expressed as following.

$$\delta_h = F^{ave} - F^h \quad (26)$$

If $\delta_h > 0$, r_h will be increase by $\frac{r_h * \delta_h}{F^{ave}}$ to make F^h equal to F^{ave}, vice versa. Therefore, r_h^i at $i\,th$ iteration can be expressed as following.

$$r_h^i = r_h^{(i-1)}\left(1 + \frac{\delta_h}{F^{ave}}\right) \quad (27)$$

Γ is used to represent standard deviation of objective function. The pseudo-codes for solving r_h^i and $\boldsymbol{\theta^i}$ by Monte Carlo method (MC-RC algorithm) is showed as following.

MC-RC algorithm

Initializing r_h^0, θ^0, δ_h,*i=0*

Calculating configuration parameters, $G^0 = g(r_h^0, \theta^0, B, L)$

Calculating of value of objective function, $F^0 = F(G, f)$

While δ_h>=*50 or* $\Gamma >= \tau$

$r_h^i = r_h^{(i-1)}\left(1 + \frac{\delta_h}{F^{ave}}\right)$

Generate three random integers between 0 and 360 and assign them to θ^i, *n mean at* $i\,th$ *iteration.*

$G^i = g(r_h^i, \theta^i, B, L)$.

$F^i = F(G^i, f)$

$\delta_h = F^{ave} - F^h$

$$\Gamma = \sqrt{\frac{1}{k}\sum_{j=1}^{k}(F_j - \overline{F})^2}$$

i=i+1;

If F^i<F^0 and δ_h<50 and $\Gamma < \tau$ *and* $\beta_1 \times mean(F_j, j = 1 \ldots k) \leq min(F_j, j = 1 \ldots k)$ *and* $max(F_j, j = 1 \ldots k) \leq$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 July 2019 doi:10.20944/preprints201907.0287.v1

$\beta_2 \times wl_{max}$

$F^0 = F^i$

$G^{opt} = G^i$

end

End

Figure 13 is a case of hub-spoke partition by optimizing the radius of hub sector and bearing of spoke using MC-RC algorithm.

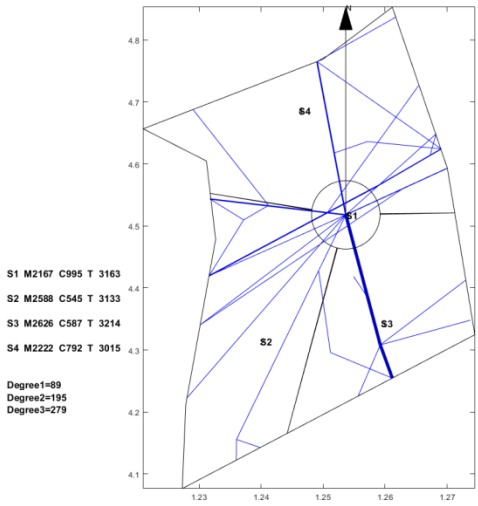

Figure 13. Hub-spoke optimizing partition using MC-RC algorithm

4. Results

This section may be divided by subheadings. It should provide a concise and precise description of the experimental results, their interpretation as well as the experimental conclusions that can be drawn.

We extracted one week's data from Shanxi regional airspace. Since there are few flights between 20:00 p.m. and 7:00 a.m. in Shanxi regional airspace, we only count the traffic flow data of each segment from 7:00 a.m. to 20:00 a.m. and take the average value to calculate the task load during the reconfiguration process. The parameters used in different algorithms are showed in Table 4.

Table 4. parameters used in algorithm

Parameters	value	SAA	MC-CLFV	MC-RC
Task load imbalance	α_1=4000	√	√	√
Total coordinating task load	α_2=200	√	√	√
Cost of short dwell time	α_3=5	√	√	√
Cost of reenter the same sector	α_4=1500	√	√	√
Initial temperature	T_0=1e+9	√	√	√
Termination temperature	T_{min}=1e-3	√	√	√
Cooling rate	σ=0.8	√	√	√
Radius decreasing rate in every iteration	σ^1=0.9		√	√
Monitoring task load per 10 minutes	α_{mot}=22/600(s/s)	√	√	√
Coordination task load required for each handover	α_{cod} =9(s/flight)	√	√	√

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 July 2019 doi:10.20944/preprints201907.0287.v1

Permissible minimum of dwell time	t_{dw}^{min}=240(s)	√	√	√
The minimum deviation from the average of total task load	β_1=0.5	√	√	√
The design redundancy parameter	β_2=0.95	√	√	√
The maximum acceptable standard deviation	τ=100		√	√
The maximum acceptable task load in each sector	wl_{max}=3420(s)	√	√	√

4.1. SAA test result

Based on the three Voronoi diagrams, we use the SAA described in Section 3 to execute the automatic sector optimization calculation for the studying airspace. The results comparing are shown in Table 5.

Table 5. Comparing SAA results with three kind of seeds generating method

Seeds generating method	TL in Sector1 (s)	TL in Sector2 (s)	TL in Sector3 (s)	TL in Sector4 (s)	*Std*	TTL in Airspace (s)	Figure
Based on randomly	3316	2409	2443	2023	546.4	10191	Figure 14
Based on every 50 kilometers	2618	2459	2688	2890	178.8	10655	Figure 15
Based on hexagonal	2383	3144	2493	2265	393.0	10285	Figure 16
Current configuration	3018	2503	4025	3031	636.6	12577	Figure 17

*note：TL is the abbreviation of task load. TTL is the abbreviation of total task load. The (s) is the abbreviation of seconds. *Std* is the abbreviation of Standard deviation.

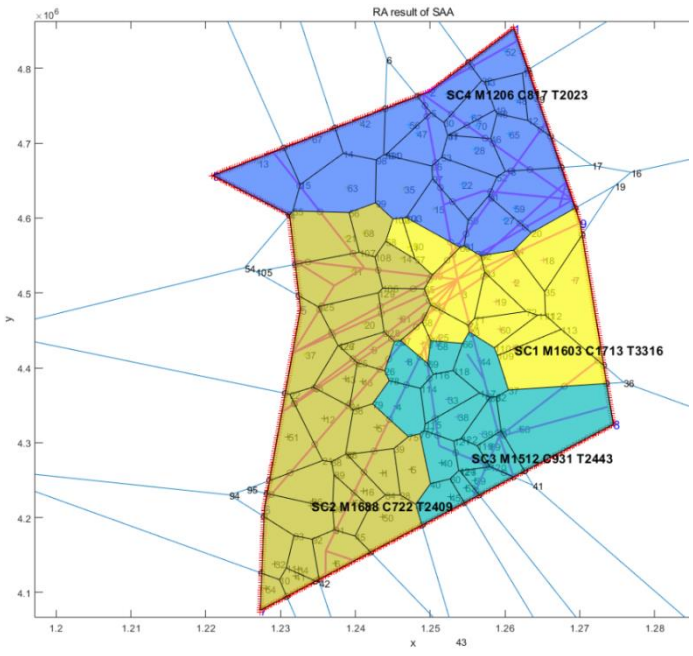

Figure 14. Reconfiguration result based on random seeds with SAA

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 July 2019 doi:10.20944/preprints201907.0287.v1

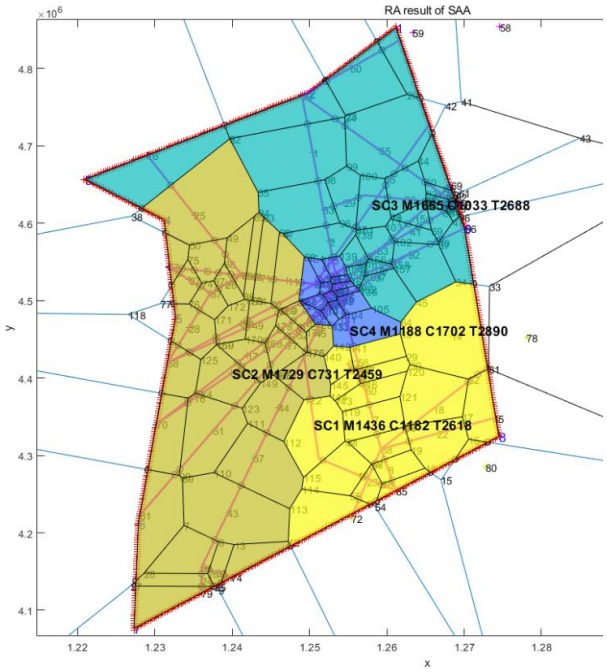

Figure 15. Reconfiguration result based on seeds every 50 kilometers with SAA

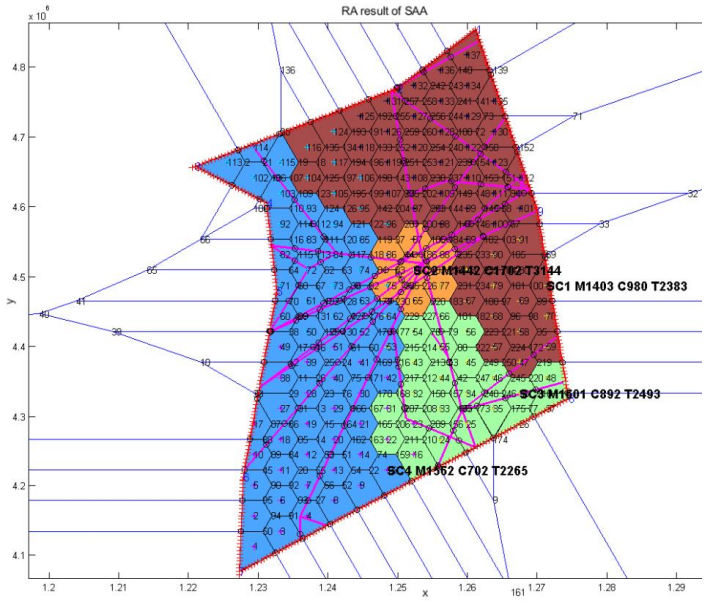

Figure 16. Reconfiguration result based on hexagonal seeds with SAA

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 July 2019 doi:10.20944/preprints201907.0287.v1

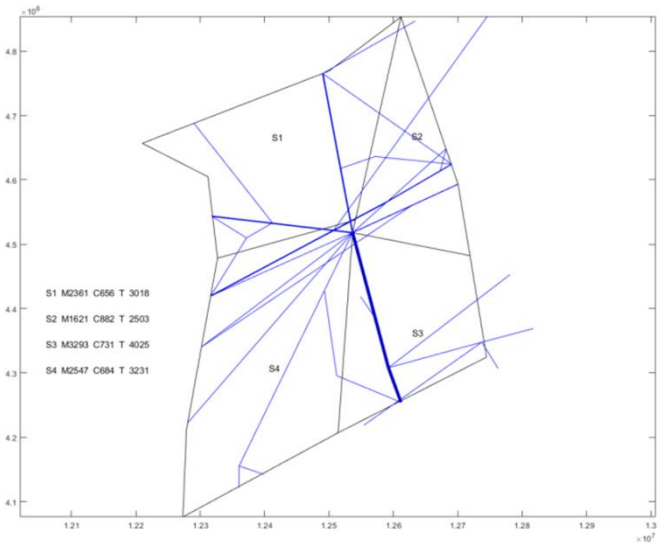

Figure 17. Current configuration and task load

4.2. Post-processing result

The first purpose of post-processing is to smooth the boundary while keep task load balance and other constrains. The second purpose is to compare the results of post-processing with MC-CLFV and MC-RC. The results of post-processing graph are showed in Figure 18-20.

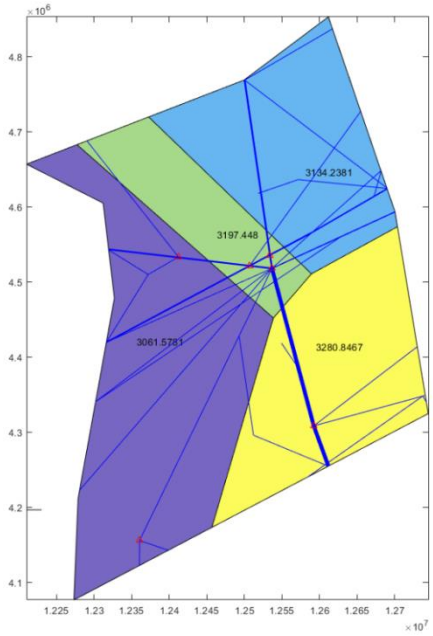

Figure 18. Post-processing result of figure 14 with MC-CLFV

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 July 2019 doi:10.20944/preprints201907.0287.v1

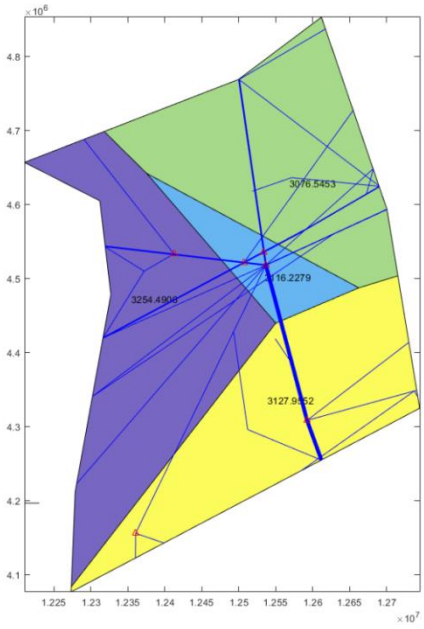

Figure 19. Post-processing result of figure 15 with MC-CLFV

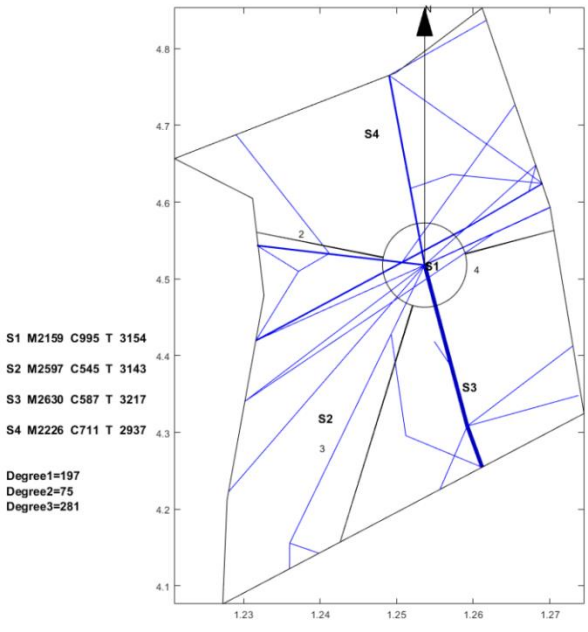

Figure 20. Post-processing result of figure 16 with MC-RC

Task loads after post-processing based on 3 kinds of seeds and task load evaluated based on current configuration show in table 6.

Table 6. Comparison of results between MC-CLFV algorithm and MC-RC algorithm

Algorithm	Original configuration	After post-processing	TL of S1 (s)	TL of S2 (s)	TL of S3 (s)	TL of S4 (s)	*Std*	TTL in Airspace (s)
MC-CLFV	Figure 14	Figure 18	3061.6	3197.5	3134.2	3280.9	93.2	12674
MC-CLFV	Figure 15	Figure 19	3254.5	3076.6	3116.2	3128.0	77.0	12575
MC-RC	Figure 16	Figure 20	3154	3143	3217	2937	121	12451
Current	Figure 17		3018	2503	4025	3031	636.6	12577

 doi:10.20944/preprints201907.0287.v1

configuration

Note: The abbreviations in this table have the same meaning as in Table 5.

5. Discussion

From the result of SAA in table 5, the sector generated by SAA method is more balanced than the current sector workload, and the total workload is smaller. We also found that the solution space of SAA varies with the seeds generating method and number of seeds. Three Voronoi diagram generation methods, combined with the number of variable seed points, improve the reliability of airspace design meeting the required performance standards. This reveals that there are defects in some previous literatures which only depend on one method to discretize the airspace. Although simulated annealing is a very mature algorithm, it must be pointed out that adding re-entry sector penalty and short dwell time penalty to the objective function will have an impact on the optimization search. If the weight of these two penalties is too large, the algorithm will be limited to local optimization, and if the settings are too small, there will be re-entry sector and short dwell. However, because the simulated annealing algorithm accepts infeasible solutions by itself with a certain probability, it allows the re-entry sector and short residence to occur in the exploration process, which makes it possible to explore the global optimization solution, and makes it possible for the region growth method to produce sectors that meet the performance requirements. Using SAA to find a good initial solution lays a foundation for warm start in post-processing.

However, irregular sector boundaries in figure 14-16 are not satisfactory. Any method based on discrete airspace block reorganization will have such problems as zigzag, c-shaped, trapezoidal and so on. Therefore, effective post-processing is very important. The result in table 6 show that the two proposed post-processing methods can achieve a more balanced partitioning scheme than the existing sector. MC-CLFV algorithm can get more balance configuration than MC-RC, but MC-RC algorithm can get smaller total task load. The choice between the two performance criteria, the minimum total workload and the balanced task load of each sector, seems to be contradictory. Of course, we expect the minimum total task load and the balanced task load of each sector at the same time. But when we can't have both of them, how do we choose? Answering this question needs to be traced back to the purpose of AS. The fundamental purpose of AS is to increase traffic throughput. Especially when the current throughput of airspace is mainly limited by the task load of controllers, the purpose of AS is to reduce the average task load of controllers, regardless of whether the number of sectors remains unchanged or by increasing the number of sectors. When the number of sectors and the demand of controllers remain unchanged, a more balanced scheme is preferred for the fundamental purpose of AS. Because only in this way, each sector has more buffer to cope with fluctuating traffic flow and traffic growth. Therefore, we conclude that MC-CLFV algorithm is superior to MC-RC algorithm in Shanxi airspace for increasing traffic throughput. However, compared with Figure 20 and Figure 18-19, it is found that there are no narrow blocks in the sector generated by MC-RC, and MC-RC seems to show some advantages when it is necessary to utilize the space of narrow blocks. We believe that any algorithm has its use value in an appropriate environment.

Comparing the results of tables 5 and 6, we can find that although the sector generated by SAA algorithm has irregular boundaries, the total task load is smaller than that after irregular boundaries are smoothed, which indicates that appropriately allowing some non-convex boundaries is a measure to reduce the total workload, and blindly pursuing convex boundaries will increase the total workload.

6. Conclusions

This section is not mandatory, but can be added to the manuscript if the discussion is unusually long or complex.

Partitioning the airspace into optimal sectors, even 2D, is still a leading or forefront research issue. It is a multi-objective optimization problem, and the contradiction between multiple objectives

or constraints makes the problem complicated. Many technologies and methods, such as Voronoi diagrams, k-mean clustering, graphic reshaping technology, graph coding technology, visualization, task load calculation, the simulated annealing algorithm (SAA) and variable neighborhood search strategies, post-processing of sector boundary based on Monte Carlo method and morphology, have been synthetically used to realize automatic AS In this paper. The modeling approach and algorithm solution presented in this paper have been tested and compared to existing sectorisation. The provided results demonstrate that the proposed sector design method framework is able to provide very satisfying sectorisation with regards to sector load balancing, as well as to total task load minimization, minimum distance to boundary, the number of re-entering and short-dwell criteria.

The graph cutting algorithms proposed by us have two theoretical contributions. The first is to eliminate the inherent problem of irregular sector boundaries generated by region growth algorithm. Second, because of the results of the previous region growing algorithm as a hint, how many sectors and the number of flexible vertices in the graph cutting algorithm become known parameters. Therefore, AS problem becomes the problem of determining the location of flexible vertices. This greatly simplifies the complexity of graph cutting algorithm.

The proposed method framework of AS and software can provide reliable assistant design and analysis tools for airspace planners to design airspace, improve the reliability and efficiency of design, and reduce the burden of airspace planners. In addition, this lays the foundation for reconstructing airspace with more intelligent method.

We do not claim that we have completed the full-automatic design of the airspace. Because airspace design is a complicated process, we have only solved the problem that the initial airspace design was almost completed manually in china in the past. This scheme has to be further simulated and evaluated by SIMMOD, AIRTOP and other tools[66]. Finally, it has to be evaluated by human in the loop before it can be implemented[4]. Some modifications will even be made based on feedback in field test. This reflects the fact that there is a huge gap for improvement in current task load model and airspace sectorization model. It is also a great opportunity and challenge for researchers to automate the whole process of airspace design.

In general, the proposed method framework not only satisfies a variety of key performance indicators, but also eliminates the irregular sector shape generated by the regional growth method, which lays the foundation for improving the reliability and acceptability of the airspace design and enriches the automatic airspace reconstruction method.

Author Contributions: conceptualization, Z.Y. and J.M.; methodology, Z.Y.; software, Z.Y. and F.K.; validation, Z.Y. and F.K; formal analysis, Z.Y.; investigation, Z.Y.; resources, W.G.; data curation, Z.Y.; writing—original draft preparation, Z.Y.; writing—review and editing, Z.Y. and B.Z.; visualization, F.K.; supervision, J.M.; project administration, Z.Y.; funding acquisition, B.Z

Funding: This work was funded by the National Natural Science Foundation of China (grant number 71571182), the National Natural Science Youth Foundation of China (grant number 61603396), the National Natural Science Foundation of China and the Civil Aviation Grant (grant number U1633124)

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Cui, Q.; Wei, Y. M.; Li, Y.; Li, W. X., Exploring the differences in the airport competitiveness formation mechanism: evidence from 45 Chinese airports during 2010-2014. *Transportmetrica B* **2017,** 5, (3), 330-346.
2. Chen, Y. X.; Yu, J.; Tsai, S. B.; Zhu, J. F., An Empirical Study on the Indirect Impact of Flight Delay on China's Economy. *Sustainability-Basel* **2018,** 10, (2).
3. Ehrmanntraut, R.; McMillan, S., Airspace design process for dynamic sectorisation. *Ieeeaaia Digit Avion* **2007**, 584-+.

4. Conker, R. S.; Moch-Mooney, D. A.; Niedringhaus, W. P.; Simmons, B. T. In *New process for "clean sheet" airspace design and evaluation*, 7th US/Europe ATM Seminar, 2007; 2007.
5. Delahaye, D.; Alliot, J.; Schoenauer, M.; Farges, J. In *Genetic algorithms for partitioning air space*, Proceedings of the Tenth Conference on Artificial Intelligence for Applications, 1-4 March 1994, 1994; 1994; pp 291-297.
6. John, R. M.; Robert, G. R.; David, B. F., Genetic Algorithms for Automatic Regrouping of Air Traffic Control Sectors. In *Evolutionary Programming IV: Proceedings of the Fourth Annual Conference on Evolutionary Programming*, MITP: 1995; p 1.
7. Babić, O.; Krstić, T., Airspace daily operational sectorization by fuzzy logic. *Fuzzy Set Syst* **2000,** 116, (1), 49-64.
8. Xue, M., Airspace Sector Redesign Based on Voronoi Diagrams. *J Aeros Comp Inf Com* **2009,** 6, (12), 624-634.
9. Chen, Y. Z.; Zhang, D. F., Dynamic airspace configuration method based on a weighted graph model. *Chinese J Aeronaut* **2014,** 27, (4), 903-912.
10. Kulkarni, S.; Ganesan, R.; Sherry, L., Dynamic Airspace Configuration Using Approximate Dynamic Programming Intelligence-Based Paradigm. *Transport Res Rec* **2012**, (2266), 31-37.
11. Gerdes, I.; Temme, A.; Schultz, M., Dynamic airspace sectorisation for flight-centric operations. *Transport Res C-Emer* **2018,** 95, 460-480.
12. Michalek, D.; Balakrishnan, H., Dynamic Reconfiguration of Terminal Airspace During Convective Weather. *Ieee Decis Contr P* **2010**, 4875-4881.
13. Zou, X.; Cheng, P.; An, B.; Song, J. Y., Sectorization and Configuration Transition in Airspace Design. *Math Probl Eng* **2016**.
14. Flener, P.; Pearson, J., Automatic Airspace Sectorisation: A Survey. *Computer Science* **2013**.
15. Bichot, C.-E.; Durand, N. In *A tool to design functional airspace blocks*, ATM 2007, 7th USA/Europe Air Traffic Management Research and Developpment Seminar, 2007; 2007; pp pp 169-177.
16. Trandac, H.; Baptiste, P.; Duong, V., Airspace sectorization with constraints. *RAIRO-Operations Research* **2005,** 39, (2), 105-122.
17. Li, J.; Wang, T.; Hwang, I.; Hwang, I. In *A spectral clustering based algorithm for dynamic airspace configuration*, 9th AIAA Aviation Technology, Integration, and Operations Conference (ATIO) and Aircraft Noise and Emissions Reduction Symposium (ANERS), 2009; 2009; p 7056.
18. Chen, Y.; Bi, H.; Zhang, D.; Song, Z., Dynamic airspace sectorization via improved genetic algorithm. *Journal of Modern Transportation* **2013,** 21, (2), 117-124.
19. Zhang, D. F.; Chen, Y. Z., Airspace sectorisation via a weighted graph model. *Aeronaut J* **2014,** 118, (1201), 267-274.
20. Yousefi, A.; Donohue, G. In *Temporal and spatial distribution of airspace complexity for air traffic controller workload-based sectorization*, AIAA 4th aviation technology, integration and operations (ATIO) forum, 2004; 2004; p 6455.
21. Martinez, S.; Chatterji, G.; Sun, D.; Bayen, A. In *A weighted-graph approach for dynamic airspace configuration*, Aiaa guidance, navigation and control conference and exhibit, 2007; 2007; p 6448.
22. Drew, M. In *A method of optimally combining sectors*, 9th AIAA Aviation Technology, Integration, and Operations Conference (ATIO) and Aircraft Noise and Emissions Reduction Symposium (ANERS), 2009; 2009; p 7057.

23. Kulkarni, S.; Ganesan, R.; Sherry, L. In *Static sectorization approach to dynamic airspace configuration using approximate dynamic programming*, Integrated Communications, Navigation and Surveilance Conference (ICNS), 2011, 2011; IEEE: 2011; pp J2-1-J2-9.
24. Bloem, M.; Gupta, P., Configuring airspace sectors with approximate dynamic programming. **2010**.
25. Jägare, P., Airspace sectorisation using constraint programming. In 2011.
26. Leiden, K.; Peters, S.; Quesada, S. In *Flight level-based dynamic airspace configuration*, 9th AIAA Aviation Technology, Integration, and Operations Conference (ATIO) and Aircraft Noise and Emissions Reduction Symposium (ANERS), 2009; 2009; p 7104.
27. Gianazza, D., Forecasting workload and airspace configuration with neural networks and tree search methods. *Artif Intell* **2010,** 174, (7-8), pp 530-549.
28. Kicinger, R.; Yousefi, A. In *Heuristic method for 3D airspace partitioning: Genetic algorithm and agent-based approach*, 9th AIAA Aviation Technology, Integration, and Operations Conference (ATIO) and Aircraft Noise and Emissions Reduction Symposium (ANERS), 2009; 2009; p 7058.
29. Drew, M., Analysis of an Optimal Sector Design Method. *Digit Avion Syst Con* **2008**, 540-549.
30. Sabhnani, G.; Yousefi, A.; Mitchell, J. S. In *Flow conforming operational airspace sector design*, 10th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference, 2010; 2010; p 9377.
31. Basu, A.; Mitchell, J. S.; Sabhnani, G. K., Geometric algorithms for optimal airspace design and air traffic controller workload balancing. *Journal of Experimental Algorithmics (JEA)* **2009,** 14, 3.
32. Sherali, H. D.; Hill, J. M., Configuration of airspace sectors for balancing air traffic controller workload. *Ann Oper Res* **2013,** 203, (1), 3-31.
33. Xue, M., Three-dimensional sector design with optimal number of sectors. *Journal of Guidance, Control, and Dynamics* **2012,** 35, (2), 609-618.
34. Gerdes, I.; Temme, A.; Schultz, M., Dynamic airspace sectorization using controller task load. *Sixth SESAR Innovation Days* **2016**.
35. Delahaye, D.; Schoenauer, M.; Alliot, J.-M. In *Airspace sectoring by evolutionary computation*, Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on, 1998; IEEE: 1998; pp 218-223.
36. Pawlak, W.; Goel, V.; Rothenberg, D.; Brinton, C. In *Comparison of algorithms for the dynamic resectorization of airspace*, Guidance, Navigation, and Control Conference and Exhibit, 1998; 1998; p 4106.
37. Sergeeva, M.; Delahaye, D.; Mancel, C.; Zerrouki, L.; Schede, N. In *3d sectors design by genetic algorithm towards automated sectorisation*, 5th SESAR Innovation days, 2015; 2015.
38. Ito, T.; Demaine, E. D., Approximability of the subset sum reconfiguration problem. *Journal of Combinatorial Optimization* **2014,** 28, (3), 639-654.
39. Cheng, C. K., The optimal partitioning of networks. *Networks* **1992,** 22, (3), 297-315.
40. Gendreau, M.; Potvin, J.-Y., *Handbook of metaheuristics*. Springer: 2010; Vol. 2.
41. Johnson, D. S.; Aragon, C. R.; McGeoch, L. A.; Schevon, C., Optimization by simulated annealing: An experimental evaluation; part I, graph partitioning. *Oper Res* **1989,** 37, (6), 865-892.
42. Johnson, D. S.; Aragon, C. R.; McGeoch, L. A.; Schevon, C., Optimization by simulated annealing: an experimental evaluation; part II, graph coloring and number partitioning. *Oper Res* **1991,** 39, (3), 378-406.
43. Rahimian, F.; Payberah, A. H.; Girdzijauskas, S.; Jelasity, M.; Haridi, S., A Distributed Algorithm for Large-Scale Graph Partitioning. *Acm T Auton Adap Sys* **2015,** 10, (2).

44. Dobrin, A., A REVIEW OF PROPERTIES AND VARIATIONS OF VORONOI DIAGRAMS. *Whitman College* **2005**.
45. Sridhar, B.; Sheth, K. S.; Grabbe, S. In *Airspace complexity and its application in air traffic management*, 2nd USA/Europe Air Traffic Management R&D Seminar, 1998; 1998; pp 1-6.
46. Crump, J. H., Review of stress in air traffic control: Its measurement and effects. *Aviation, space, and environmental medicine* **1979**.
47. Averty, P.; Athenes, S.; Collet, C.; Dittmar, A. In *Evaluating a new index of mental workload in real ATC situation using psychophysiological measures*, Digital Avionics Systems Conference, 2002. Proceedings. The 21st, 2002; IEEE: 2002; pp 7A4-7A4.
48. Martin, C.; Cegarra, J.; Averty, P. In *Analysis of mental workload during en-route air traffic control task execution based on eye-tracking technique*, International Conference on Engineering Psychology and Cognitive Ergonomics, 2011; Springer: 2011; pp 592-597.
49. Baart, D., An evaluation of dynamic density metrics using RAMS. *FAA Technical Center Report* **2001**.
50. Yousefi, A.; Donohue, G. L.; Qureshi, K. In *Investigation of en route metrics for model validation and airspace design using the total airport and airspace modeler (TAAM)*, Proceedings of the fifth USA/Europe Air Traffic Management R&D Seminar, 2003; 2003.
51. Majumdar, A.; Ochieng, W.; McAuley, G.; Lenzi, J.; Lepadetu, C. In *The factors affecting airspace capacity in europe: A framework methodology based on cross sectional time-series analysis using simulated controller workload data*, Proceedings of the 6th USA/Europe Air Traffic Management R & D Seminar, 2005; 2005.
52. Oktal, H.; Yaman, K., A new approach to air traffic controller workload measurement and modelling. *Aircr Eng Aerosp Tec* **2011,** 83, (1), 35-42.
53. Chatterji, G.; Sridhar, B. In *Measures for air traffic controller workload prediction*, 1st AIAA, Aircraft, Technology Integration, and Operations Forum, 2001; 2001; p 5242.
54. Kopardekar, P.; Magyarits, S. In *Measurement and prediction of dynamic density*, Proceedings of the 5th USA/Europe Air Traffic Management R & D Seminar, 2003; 2003.
55. Gianazza, D. In *Learning air traffic controller workload from past sector operations*, ATM Seminar, 12th USA/Europe Air Traffic Management R&D Seminar, 2017; 2017.
56. Jorna, P., Operator workload as a limiting factor in complex systems. In *Automation and systems issues in air traffic control*, Springer: 1991; pp 281-292.
57. Council, N. R., *Air Traffic Controller Staffing in the en Route Domain: A Review of the Federal Aviation Administration's Task Load Model*. Transportation Research Board: 2010; Vol. 301.
58. FAA, Order 7210.3, Facility Operation and Administration. In FAA: 2007.
59. Palla, N., Graph connectivity by the adjacency matrix. *Far East J.appl.math* **2003,** 12, (3), 189-225.
60. Kanazawa, N.; Sasahira, T.; Kaneda, S.; Haga, H. In *Automatic generation method of relation matrix for ISM*, Conference on Knowledge-based Software Engineering, 2006; 2006.
61. Likas, A.; Vlassis, N.; Verbeek, J. J., The global k-means clustering algorithm. *Pattern Recognition* **2003,** 36, (2), 451-461.
62. De, L. A.; Blanco, S.; Clergent, Y.; Dufresne, J. L.; El, H. A. F. R., Monte Carlo method and sensitivity estimations. *Journal of Quantitative Spectroscopy & Radiative Transfer* **2018,** 75, (5), 529-538.
63. Wakefield, A.; Miller, S. In *Improving System Models by Using Monte Carlo Techniques on Plant Models*, AIAA Modeling and Simulation Technologies Conference and Exhibit, 2008; 2008; p 7094.
64. Mogford, R. H.; Bridges, W.; Gujarl, V.; Lee, P. U.; Preston, W., Generic Airspace Survey. **2013**.

 doi:10.20944/preprints201907.0287.v1

65. Guttman, J. A.; Stein, E.; Gromelski, S. *The Influence of Generic Airspace on Air Traffic Controller Performance*; FEDERAL AVIATION ADMINISTRATION TECHNICAL CENTER ATLANTIC CITY NJ: 1995.

66. Schier, S.; Papenfuss, A.; Lorenz, S.; Walther, J.; Moehlenbrink, C., An approach to support controller work-place design in a multi-airport environment using fast and real-time simulations. *CEAS Aeronautical Journal* **2011,** 2, (1-4), 185-193.