32nd Congress of the International Council of the Aeronautical Sciences
September 6-10, 2021
Pudong Shangri-La, Shanghai, China

ARTIFICIAL INTELLIGENCE AND HUMAN-MACHINE INTERACTIONS FOR STREAM-BASED AIR TRAFFIC FLOW MANAGEMENT

Pannawat Lertworawanich[1], Nichakorn Pongsakornsathien[1], Yibing Xie[1], Alessandro Gardi[1,*], Roberto Sabatini[2]

[1]RMIT University – School of Engineering, Bundoora, VIC 3083, Australia

[2]Dept. of Aerospace Engineering, Khalifa University of Science and Technology, PO Box 127788, Abu Dhabi, UAE

Abstract

Considerable growth in air traffic has led to airspace congestion in certain regions, with the consequent need of introducing new decision support systems and flexible schemes to optimally manage the available resources, towards maximising efficiency and safety of air operations. This evolution has elicited the introduction of higher levels of automation, which can support en-route Air Traffic Flow Management (ATFM) systems to deliver a more efficient route planning and balancing demand and capacity of airspace sectors. The stream-based management paradigm has been proposed as a promising strategy to improve the efficiency of ATFM, which is selected for this study as it can also enhance the intuitiveness and interpretability of system resolutions. A clustering algorithm is proposed in this paper to automatically identify the traffic streams, addressing the need for an optimal method in stream identification. In addition, a hybrid Artificial Intelligence (AI) approach is implemented for the autonomous determination of Traffic Flow Management Initiatives (TFMI) for each stream, and thus to demonstrate the potential use of the stream-based traffic. Lastly, custom Human-Machine Interactions (HMI) are designed and prototyped to improve the ATFM operator's situational awareness and overall human-machine teaming.

Keywords: Air Traffic Flow Management, Cognitive Human Machine Interfaces and Interactions, Stream-based management, Human-Machine Teaming

1. Introduction

The progressive growth in air traffic density in various regions is posing challenges to the efficiency of Demand-Capacity Balancing (DCB) processes in en-route airspace sectors, a time-critical task fulfilled by Air Traffic Flow Management (ATFM) services. This trend is eliciting an evolution of ATFM systems planning and demand forecast models, which are increasingly relying on sophisticate Artificial Intelligence (AI). However, the complexity of overload situations and of associated AI resolutions may exceed the cognitive capabilities of the human operators, thus compromising their situation awareness and potentially leading to undesirable effects such as distrust, cognitive overload, etc. [1]. For this reason, the design of AI-based ATFM systems with a "human-on-the-loop" philosophy shall to the extent possible resort to more intuitive concepts and integrate explainable AI (XAI) functions to maintain optimal human-machine teaming.

As part of DCB processes, suitable Traffic Flow Management Initiatives (TFMI) are to be determined to resolve overload instances, and these include various operational measures for every phase of flight such as take-off, approach, and cruise. Speed control, path-stretching and re-routing measures have been commonly implemented, while Dynamic Airspace Management (DAM) concepts are also explored for the future [2]. Evolving operational conditions and weather patterns can sometimes generate complex and unpredicted overload situations which could require intense workload especially during the peak traffic periods [3]. In line with SESAR and NextGen, Communication Navigation and Surveillance (CNS) and ATM (CNS+A) systems both onboard and on the ground are evolving in three steps: Time-based Operations, Trajectory-based Operations (TBO), and Performance-based Operation (PBO) [4]. This evolutionary pathway supports the progressive merge of strategic DCB and deconfliction duties, which is made possible by the introduction of 4-Dimensional Trajectories (4DT). For instance, the optimisation of 4DT by CNS+A systems was proposed as a way to integrate high levels of automation in 4D-TBO [2]. Various works addressed specific aspects of ATFM including its methods, objectives, and operational phases such as strategic, pre-tactical, and tactical phase, which inform the scenario design [5, 6]. ATFM has an important role of planning the flow of traffic in advance, and balancing demand and capacity of each sector. The mathematical model to estimate the en-route capacity consider various operational and weather factors. In conventional Monitor Alert Parameter (MAP)-based approaches, the maximum number of aircraft inside any sector shall not exceed 18 [7, 8]. Performance enhancements of en-route ATFM services are essential due to a huge growth in traffic, and in order to effectively handle a large amount of information. Several ATFM strategies have been proposed to improve DCB problem. One study proposed a new operational concept for ATFM, which is the stream-based management [9]. The concept groups traffic with similar routes into a stream; hence, the complexity of the sector and information is reduced. This paper adopts the stream-based traffic to be a platform for TFMI determination models solving DCB issues. The study investigates an efficient and robust clustering approach to determine the streams; hence, these streams are input to the TFMI determination process, which in this study is based on a genetic algorithm. The analysis includes the comparison of TFMI determination efficiency between adopting the stream-based and individual aircarft. Lastly, the HMI design of the proposed stream-based management is presented in Section 5.

1.1 Stream-based Management

Stream-based management was introduced by Wei et al. to improve ATFM efficiency [9]. It is an operational ATFM concept for which a stream is defined as a group of aircraft with a similar route. Within one stream, all aircraft are managed in the same manner and comply with the same TFMI. The concept also supports the relinquishment of airspace sectors, assuming that an individual ATCO is responsible for managing traffic streams rather than individual sectors. The responsibility for separation assurance and the management of crossing traffic would be shifted to automation [10]. A comparison between traditional sector-based management and stream-based operations showed potential to enable both ATCO and ATFM system to handle more aircraft with less complexity and workload than in conventional sector-based management. The measures used in this study were dynamic density and complexity. The stream was defined based on aircraft types, destination airports, and arrival gates, which is consistent with FAA' definition of a stream in the current Traffic Management Advisor (TMA) system [9]. This new concept is not yet implemented in the current ATFM system, and the mathematical framework underpinning this concept was only partially developed. This paper partially adopts the stream-based management concept because the airspace sectors are still considered in this study. This paper focusses on the development of a suitable clustering algorithm to automatically identify the streams based on multiple criteria, which are detailed in Section 2. The AI-based generation of TFMI for individual streams and an HMI concept are also presented and discussed.

2. Traffic Stream Identification

Clustering methods are frequently adopted for traffic flow identification [11, 12]. Unsupervised Machine Learning (ML) algorithms are generally chosen for such clustering, such as K-means [13], Gaussian Mixture Model (GMM) [14], Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [15], and Hierarchical DBSCAN (HDBSCAN) [16]. Considering that the stream is not defined upfront automatically, the clustering algorithm needs to be fast and reliable. Hence, the analysis in this paper adopts K-means clustering algorithm to identify the traffic stream due to its simplicity, computational efficiency, and comparably high transparency. The process comprises of four steps: constructing feature vectors, standardizing data, clustering data, and evaluating the performance. The metric to evaluate the clustering efficiency is a silhouette method ($s(o)$) which determines how close a point is to its cluster relative to other clusters [17].

2.1 Feature Vectors

Since the traffic stream concept is different from the traditional traffic flow, parameters that should be considered as the feature vectors have to be defined upfront. The feature vectors from the literature are applied, i.e., engine types, unit vector of origin, and destination. However, the unit vector of the origin-destination pair only tells that two lines are parallel but cannot distinguish how far apart they are. Therefore, the midpoint of each route is also used. The last parameter is the estimated operation time of each aircraft; this parameter is important for the TFMI determination algorithm that is detailed in Section 3. The following list presents all feature vectors adopted in this study:

1. Engine types: jet = 1, turboprop = 2 and propeller =3;
2. Route unit vector, defined as:

$$\hat{v} = \frac{v}{\|\vec{v}\|} = \frac{((lat, lon)_{destination} - (lat, lon)_{origin})}{\|\vec{v}\|} \tag{1}$$

3. Route midpoint (latitude, longitude), defined as:

$$Centroid(lat, lon) = \frac{((lat, lon)_{destination} - (lat, lon)_{origin})}{2} \tag{2}$$

4. Estimated operation time (s), defined as:

$$t = \frac{Route\ distance}{v_{ac}} \tag{3}$$

where v_{ac} is the average aircraft cruise speed (m/s).

2.2 Data Standardization

The z-score index is adopted in this paper to differentiate the relative contribution of individual features to the clustering, meaning that when the standard deviation σ is high, the z-score of the relative feature will be low. Thus, the feature that has a higher σ leads to a lower contribution to the clustering.

$$zscore_{iv} = \frac{x_{iv} - \bar{x}_v}{\sigma_v} \tag{4}$$

where x_{iv} is defined as the raw data point of the feature
$\bar{x}_v$ is defined as the mean of the feature
σ_v is defined as the standard deviation of the feature.

2.3 Clustering Algorithm

The K-means clustering algorithm is adopted in this study to preliminarily demonstrate traffic stream classification. The K-means algorithm is a partitioning method, using the nearest mean of the distance metric to assign n observations into k clusters [18]. The distance metric commonly implemented in aviation literature is the Euclidean distance due to its low computational requirements [19]. The objective function is defined as:

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \|x_i - c_j\|^2 \tag{5}$$

where: J is the sum of squared errors;
c_j is the selected data cluster center point;
x_i is the data point.

The required input for K-means clustering is the number of clusters (k) that needs to be specified upfront. The optimal number of k is then determined by the elbow method. The elbow method identifies clusters utilizing the minimum value of intra-cluster variation, most often measured by variance. The ratio used is inter-group variance to total variance. The clustering performance is evaluated by the silhouette method.

3. TFMI Determination

This section details the TFMI determination function adopted for solving DCB overload issues, adopting the Evolutionary Algorithm (EA). The EA has the characteristics of randomness, high efficiency and global search optimization [20]. In traffic management, and specifically optimization forecasting and planning problems, EA has shown outstanding versatility [21-23], which can effectively allow to determine optimal and near-optimal solutions for non-deterministic polynomial hard (NP hard) problems. EA has a variety of derivative algorithms, including Genetic Algorithm (GA), evolutionary strategy (ES) and evolutionary programming (EP) [24]. As a classic EA, GA algorithm is easier to modify and adapt to the environment and tasks while ensuring better performance. Therefore, the algorithm used in this study is GA.

3.1 Genetic Algorithm

GA is essentially a parallel global search method based on the evolutionary law of "survival of the fittest". The new population is generated by crossover and mutation operations on the initial population through modularization and coding [25]. In each iteration of the algorithm, a new set of solution approximations are selected through fitness evaluation. A better population is chosen according to the fitness evaluation function for the next generation of operations [26]. The GA steps are described as follows:

1. Initialisation

   Three types of data are initialized: aircraft states data, airspace and environmental states data, and clustering data. The initial operation regards the group by clustering as the individual of the initial population and randomly generates a plural population. Generally, the range is 50 to 200. A larger population will increase the complexity of the calculation, while too small a population size will affect the performance of the algorithm optimization.

2. Fitness

   The fitness F of the population is calculated based on the individual fitness f_g （Clustering group/Traffic stream. The f_g value is obtained based on the fitness f_i of the aircraft in the group. Equation 6, 7 and 8 show the calculation method of f_i, f_g and F.

$$f_i = \mathrm{t}_e^A - \mathrm{t}_o^A \tag{6}$$

$$f_g = \sum_{i=1}^{NG} f_i \ (i = 1, 2, 3, \dots, NG) \tag{7}$$

$$\mathrm{F} = \sum_{g=1}^{NP} f_g \ (g = 1, 2, 3, \dots, NP) \tag{8}$$

   where the definition of t_e^A is estimate arrival time; and t_o^A is planned arrival time. NG means the number of aircrafts in a stream.

3. Selection

   The selection is made according to the fitness value of the stream. The selection is based on the results of the double ranking, the result of airspace overload and the value of population fitness. The airspace overload is arranged in a positive order. The result with the lowest overload situation is first obtained; the population fitness value is based on the result of the airspace overload in reverse order, and the individual with the highest group fitness value is obtained. Then one-third of the optimal solution is retained as the parent of the next generation of chromosomes.

4. Crossover

   The crossover operation is divided into random crossover and probability crossover. Probabilistic crossover is based on P_c, which controls the frequency of crossover operations. Paired individuals exchange their gens (actions) at the crossover position to generate a new population. The value range of P_c is 0.25 to 1.00; the low value may make the algorithm dull and unable to obtain the target solution.

5. Mutation

   The mutation operation is an auxiliary operation in GA, and its purpose is to maintain the diversity of the population. This operation randomly changes the gens (actions) of an individual (traffic stream) based on the mutation probability P_m. The value of $P_m$ranges from 0.001 to 0.1; too high a mutation frequency will cause the algorithm to tend to search randomly.

6. Determine

   The last step calculates the fitness of the newly generated population and checks whether the output results meet the conditions. The outputs are listed below:

a) The state data set of each aircraft;
b) The state data set of the airspace;
c) The corresponding decision-making actions of each stream;
d) The fitness of population F, individualf_g, aircraft f_i;
e) The airspace overload data

4. Verification Case Study

The verification case study is performed on synthetic air traffic data in Australia, with the number of simulated aircraft set to 60. The synthetic traffic is generated on the routes from the top ten airports by traffic in Australia, which include: Sydney, Melbourne, Brisbane, Perth, Adelaide, Gold Coast, Canberra, Cairns, Hobart and Darwin. This section presents the verification case study for both stream identification and TFMI determination.

4.1 Traffic Stream Identification

The synthetic data is generated as illustrated in Figure 1, showing each trajectory of each aircraft. The feature variables are constructed from all 60 aircraft where the engine types are randomly assigned to each aircraft. Then, all feature variables (engine types, unit vector, centroid and estimated operation time) are then standardised by z-score index and inputted to the elbow method.

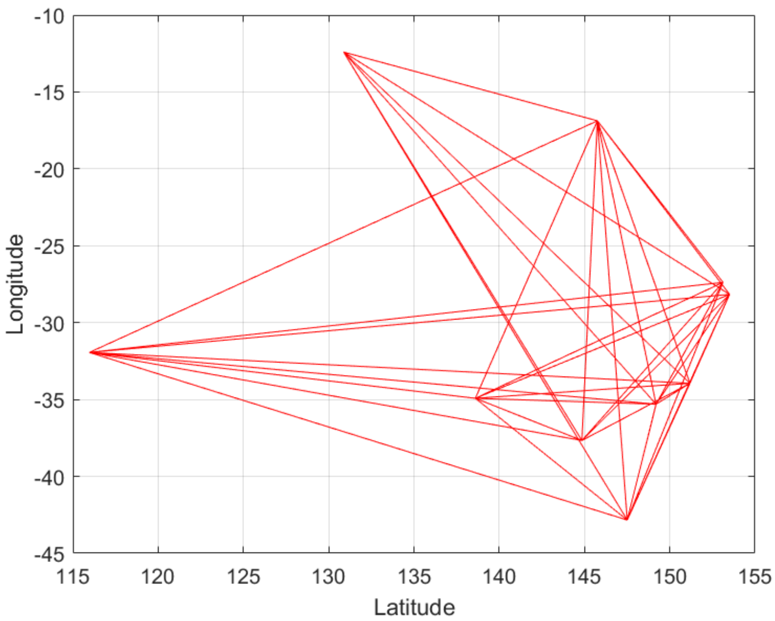

Figure 1 – Synthetic traffic routes in Australia from the top ten airports.

The result of the elbow method is lying between the ratio of more than 0.8 and a cutoff point, which leads to a k value between 10 and 13. Therefore, K-means clustering is performed with four k values. Each k value provides different silhouette values. The performance evaluation of each k is then to compute the average silhouette values. Such k that has the highest silhouette values is chosen to be the optimal numbers of cluster. The average of silhouette values is calculated. When k = 10, s(o) is 0.439, k = 11, s(o) is 0.396, k = 12, s(o) is 0.559, k = 13, s(o) is 0.497 Hence, the best s(o) is when k = 12. The twelve traffic streams are illustrated in Figure 2. The dashed lines represent the original routes with color-coding of each stream, while the thick lines represent the traffic streams. These streams are then inputted to the TFMI determination algorithm solving DCB issues.

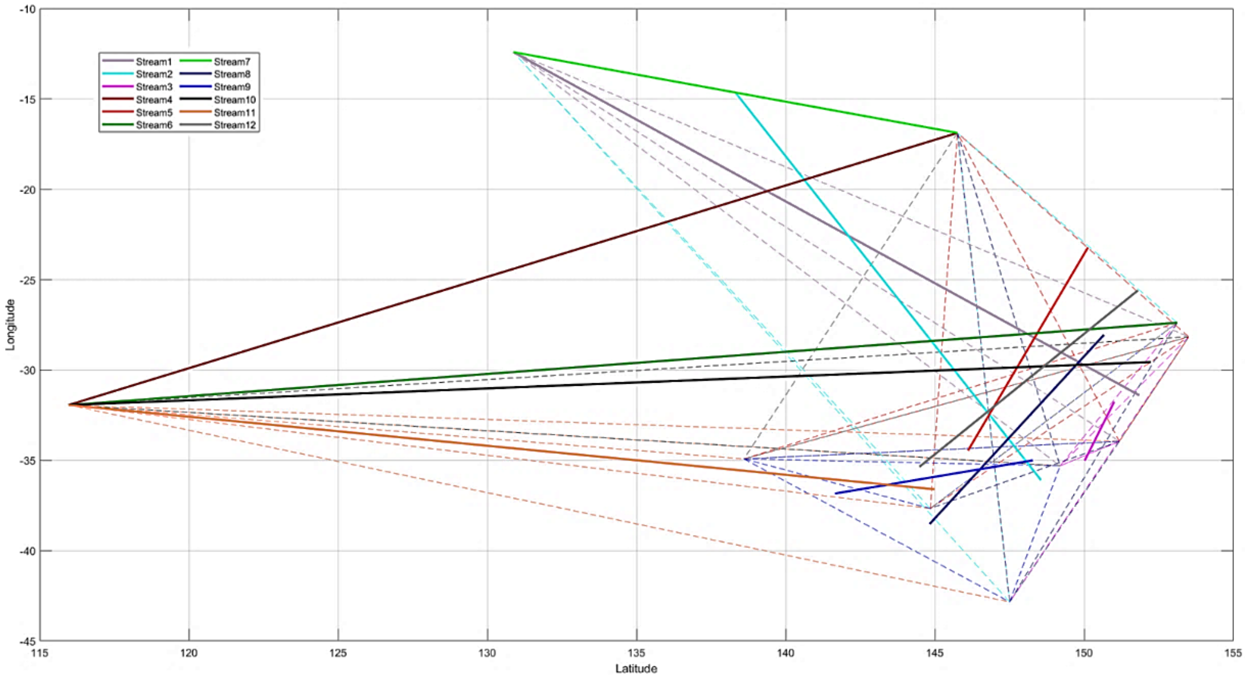

Figure 2 – Automatically determined traffic streams from K-means clustering.

4.2 TFMI determination

In this case study, the TFMI determination function is designed to choose among seven strategic decision actions (Table 1), which are based solely on speed control for this initial verification activity. Four parameters need to be firstly set for the GA, including the population size of NP, number of iterations, crossover probability, and mutation probability also shown in Table 1. In our simulation, the GA method generated a total of 523 solutions, of which 68 effective solutions to the DCB overload are achieved, while the rest of the solutions only partly solve the capacity overload. In the initial state, the number of aircraft in the overloaded airspace is 32, and the total computation time takes 1685 seconds. Figure 3a and 3b show the results comparing TFMI determination with the streams and TFMI determination with individual aircraft, both based on the number of overload instances and the number of iterations.

Table 1 - Detail of Strategic Actions and Parameter setting of GA.

Strategic Actions	
1. Normal	Remain unchanged
2. Speed Up 1	the flight speed increases by 2.5 meters per second
3. Speed Up 2	the flight speed increases by 3.0 meters per second
4. Speed Up 3	the flight speed increases by 3.5 meters per second
5. Speed Down 1	the flight speed drops by 1.5 meters per second
6. Speed Down 2	the flight speed drops by 1.0 meters per second
7. Speed Down 3	the flight speed drops by 0.5 meters per second

GA Parameters Setting	
The population size of NP	50
Number of iterations	500
Crossover probability P_c	0.7
Mutation probability P_m	0.1

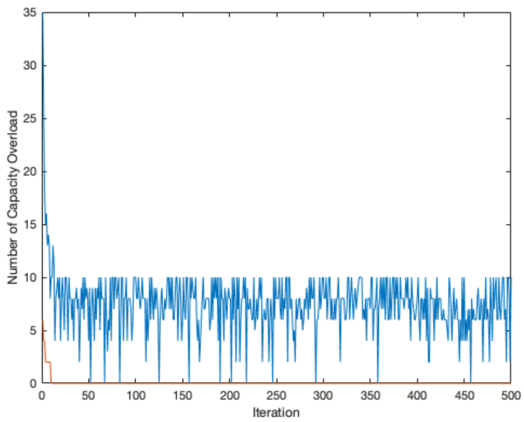

Figure 3a. Number of iterations and capacity

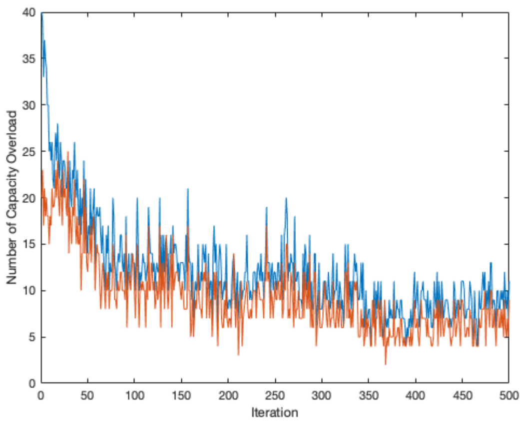

Figure 3b. Number of iterations and capacity

overload instances in the streams. overload instances in individual aircraft.

As specified in Table 1, each iteration provides 50 solutions; the blue line in the figure represents the maximum value of the number of overload instances while the orange line represents the minimum value of it. Figure 3a shows the results of the traffic streams, and the system calculation took 97.18 seconds. In the initial state, the overload instance number is the highest. The algorithm obtains the first effective DCB solution at the 10th iteration, as highlighted by a drop of overload instances down to zero (orange line). Then, the solution of the algorithm stabilizes and starts to converge. Figure 3b shows the case of non-stream-based. The calculation time of the system is 91.8 seconds, and the calculation speed is 5.5% faster than the streams. However, the algorithm does not obtain any effective DCB solution in a limited number of iterations; all generated solutions can only partly solve capacity overload.

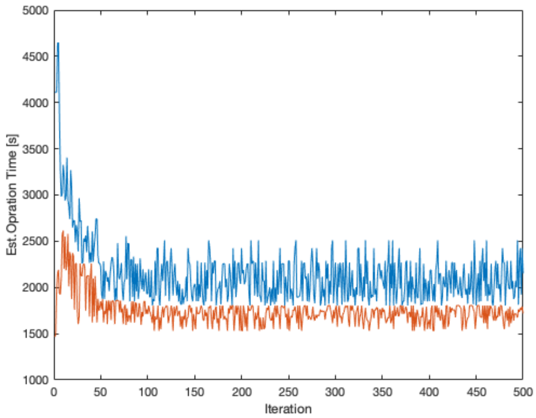

Figure 4a - Number of iterations and optimization results from the streams.

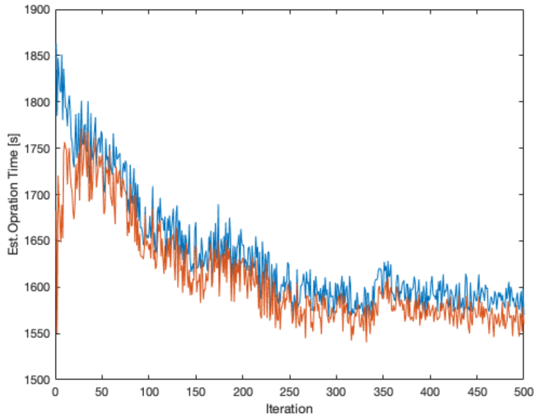

Figure 4b Number of iterations and optimization results from individual aircraft.

Figure 4a shows the results of pre-processing data after clustering. In the initial iteration of the system, the flight time of the aircraft in the airspace was as high as 4642.5 seconds, and then it began to descend rapidly, and after 50 iterations, it was smoothly controlled within 1600 to 2500 seconds. In time, the algorithm began to converge. As can be seen in Figure 4b, the system without clustering pre-processing has not converged in a limited number of iterations. And since no DCB solutions are obtained, these are all invalid solutions.

In this case study, the total operation time of the DCB optimal solution is 1804.9 seconds. Among the 12 traffic streams, the number of clustering groups selected for Action 2 is 4; the number of each selected Actions 3, 4, 5, and 6 are 2; Action 1 and Action 7 are not selected for execution.

5. HMI for Stream-based Management

In ATFM operation, the graphical interface has considerable influence on the performance of DCB. Therefore, the graphical interface for ATFM should be well designed to improve ATCO's interpretability and human-machine teaming. Even though the verification case study only presents one TFMI, which is speed control, the HMI design concept presented here is not limited to only speed control but also includes other TFMI supporting future work development. The graphical interface is implemented in CesiumJS, an open-source JavaScript library for 3D geospatial visualisation. The ATCO is able to customise the visualization as they prefer. The traffic density and DCB can be represented in three ways: through the stream, through the sector, and bar chart. The first visualization style, which represents traffic density via a stream, is illustrated in Figure 5. The colour is used to represent the density of each stream: red means very dense, yellow is medium and green is low. Furthermore, the DCB of a sector is displayed as a pop-up window when the ATCO selects the sector. The bottom of the display is the timeframe slide bar where the ATCO can choose the specific time period of interest. In addition, the weather cell is also presented in the graphical interface.

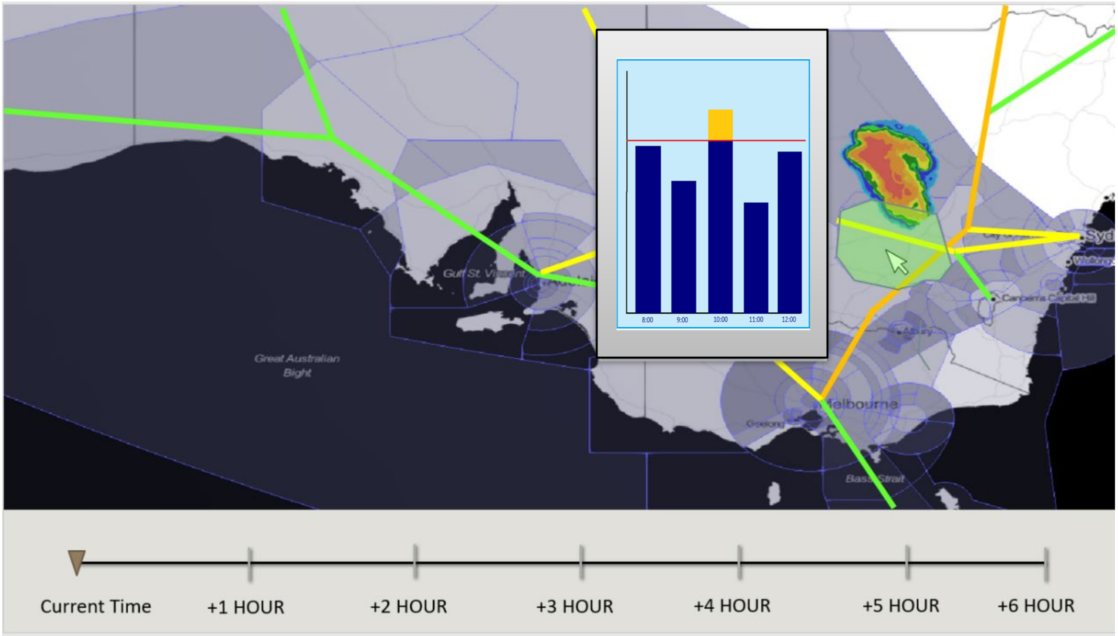

Figure 5 - ATFM display design for stream-based traffic density visualization.

For the second approach, as shown in Figure 6, the colour of each sector represents the demand and capacity of that sector instead of the stream. Therefore, the operators do not have to use the mouse button to view the balance in each sector, which means that the level of interpretability and situational awareness is enhanced. In addition, the following menus are provided for TFMI review and amendment: re-routing, speed control, and sector morphing along with submenus for each technique is also accessible by right-clicking on a stream. The ATCO can implement various tasks in this graphical interface, including choosing the speed and time to apply for speed control and using the text box to choose the number of aircraft for re-routing. At the same time, sector morphing would be performed by moving the vertex represented by circles on the boundary of sectors, as shown in Figure 7. The interactions for the ATCO, such as opening the menus and calling for a pop-up window, are triggered by the mouse button and hotkeys. Right-click is used to open menus which are re-routing, speed control, and sector morphing, while pop-up window would be triggered by pressing shift and right-click simultaneously. Left-click would be used for most of the actions in the interface, the select the menus of re-routing, speed control, and sector morphing would be achieved by left click.

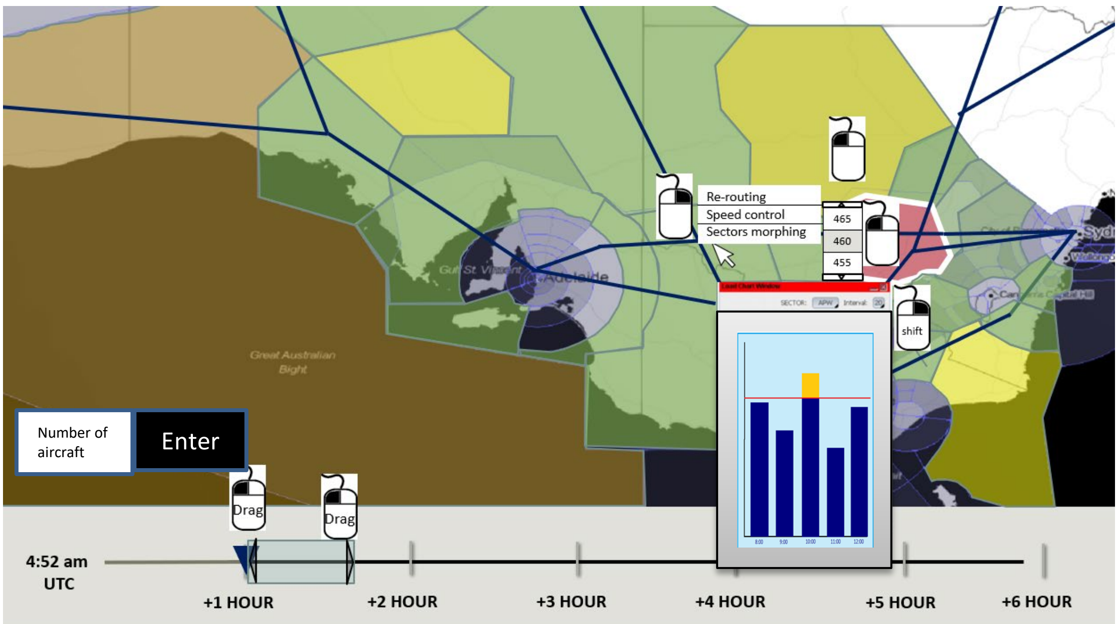

Figure 6 - ATFM display design for stream-based DCB visualization.

Moreover, each technique needs further actions in which speed control is performed by changing the speed and choosing the duration. Speed could be selected by left-clicking, while the duration could be selected by left-clicking and drag to cover the desired duration. Re-routing requires a number of aircraft to deviate from the original routes; it could be done by typing the number of aircraft and then press enter button.

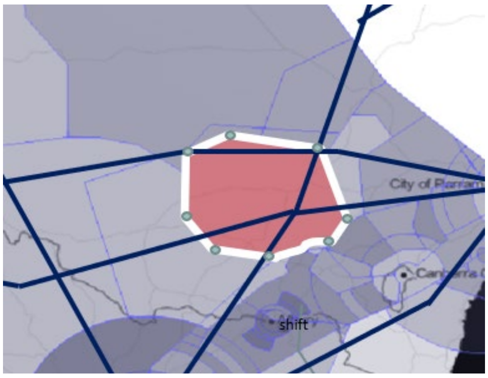

Figure 7 - Airspace morphing function.

The last visualization approach is illustrated in Figure 8. There is a total of six features added into the graphical interface, the number one represents the traffic data including ten airports in Australia, and international flights and airspace sectors which were imported to display all the routes, and flights could move in the real-time as they were planned in the traffic data file. After the traffic and sectors are displayed in the graphical interface, menu for sector morphing is shown when the sector is selected as shown in number two. Menus for routes which are re-routing and speed control, as represented in number three, could be open when the route is selected. Additional functions to cooperate with speed control and re-routing have been emerged in the graphical interface. When speed control is chosen to implement, the speed below speed control menu could be adjusted, while there is a text box of “Enter number of aircraft” as shown in number four, to select number of aircraft to deviate from overloaded sectors. For the number five, it is histogram that was imported from JavaScript D3 library and adjusted for the application of showing demand and capacity in the sector from the current time to the next three hours with 30 minutes interval. The function for the histogram was successfully capable of plotting demand and capacity of selected sector and update the traffic after balancing by using function of aircraft counter. The timeline imported as represented by number six was added for the purpose of future time visualisation, so the operators could notice problem of imbalance of demand and capacity in sectors and apply ATFM techniques to solve the problem in advance.

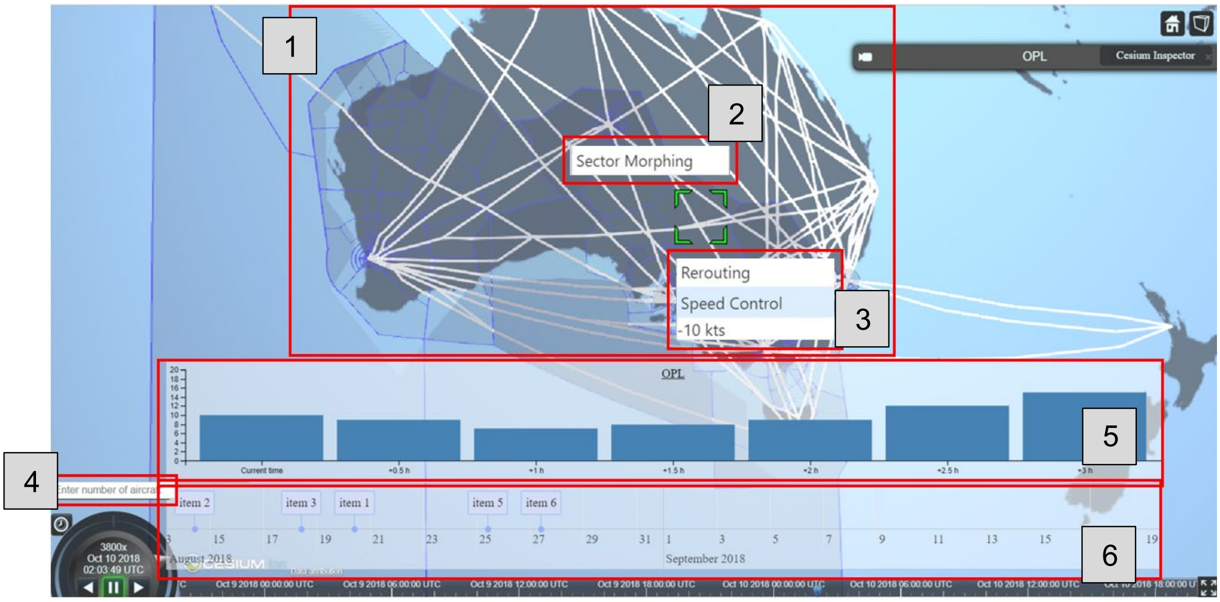

Figure 8 - ATFM display design for stream-based approach.

6. Conclusion

This paper has investigated the adoption of the stream-based management paradigm in Air Traffic Flow Management (ATFM), which holds great promise to enhance the intuitiveness, interpretability and efficiency of Demand-Capacity Balancing (DCB) solution processes. An unsupervised clustering algorithm, k-means, was proposed and evaluated to autonomously identify the traffic streams, based on four feature vectors: origin-destination unit vector, origin-destination centroid, engine type and estimated operation time. The output of such clustering algorithm, traffic streams, is then input to the Traffic Flow Management Initiative (TFMI) determination process, which in this study exploits a genetic algorithm to resolve DCB issues only by speed control measures. The verification case study highlights the efficiency gains achievable when dealing with traffic streams instead of individual aircraft, by obtaining effective solutions in early iterations. Moreover, a custom Human-Machine Interactions (HMI) design concept was proposed to enhance human-machine teaming by introducing stream-based management and TFMI solutions formats and functions. The proposed stream-based management reduces the complexity for both the human operator (via the proposed HMI) and the system by lowering the number of individual entities shown and handled by the operator to determine the most appropriate traffic flow management initiatives.

Contact Author Email Address

mailto: alessandro.gardi@rmit.edu.au

7. Copyright Statement

References

[1] ICAO, "Human Factors training manual", ed. Montreal, Canada, 1998.

[2] T. Kistan, A. Gardi, R. Sabatini, S. Ramasamy, and E. Batuwangala, "An evolutionary outlook of air traffic flow management techniques", *Progress in Aerospace Sciences,* vol. 88, pp. 15-42, 2017. DOI: 10.1016/j.paerosci.2016.10.001

[3] S. Athènes, P. Averty, S. Puechmorel, D. Delahaye, and C. Collet, "ATC complexity and controller workload: Trying to bridge the gap", *International Conference on Human-Computer Interaction in Aeronautics, HCI-02*, Cambridge, MA, USA, 2002

[4] SESAR Joint Undertaking, "Roadmap for Sustainable Air Traffic Management, European ATM Master Plan", ed: October, 2012.

[5] A. P. Timoszczuk, W. N. Pizzo, G. F. Staniscia, and E. Siewerdt, "The SYNCROMAX solution for air traffic flow management in Brazil", in *Computational models, software engineering, and advanced technologies in air transportation: Next generation applications*, pp. 23-37, IGI Global, 2010.

[6] ICAO, "Asia Pacific Framework for Collaborative Air Traffic Flow Management", ed: Asia/Pacific Air Traffic Flow Management Steering Group (ATFM/SG), 2017.

[7] J. Y. Cho, J. D. Welch, and N. K. Underhill, "Analytical workload model for estimating en route sector capacity in convective weather", *9th USA/Europe ATM R&D Seminar, Berlin, Germany*, 2011

[8] J. D. Welch, J. W. Andrews, B. D. Martin, B. Sridhar, and M. Field, "Macroscopic workload model for estimating en route sector capacity", *Proc. of 7th USA/Europe ATM Research and Development Seminar, Barcelona, Spain*, 2007, p. 138

[9] P. Wei, C. Surakitbanharn, S. Landry, and D. Sun, "Workload evaluation of sectorized air traffic control and stream management", *2012 Integrated Communications, Navigation and Surveillance Conference*, 2012, pp. N2-1-N2-8

[10] P. Kopardekar, K. D. Bilimoria, and B. Sridhar, "Airspace configuration concepts for the next generation air transportation system", *Air Traffic Control Quarterly,* vol. 16, pp. 313-336, 2008

[11] A. Eckstein, "Automated flight track taxonomy for measuring benefits from performance based navigation", *2009 Integrated Communications, Navigation and Surveillance Conference*, 2009, pp. 1-12. DOI: 10.1109/ICNSURV.2009.5172835

[12] X. Olive, L. Basora, B. Viry, and R. Alligier, "Deep Trajectory Clustering with Autoencoders", *Proceedings of the International Conference on Research in Air Transportation*, 2020

[13] A. S. Torne, "Route clustering for strategic planning in air traffic management", PhD Thesis, University of California - Irvine, Irvine, CA, USA, 2015.

[14] X. Olive and L. Basora, "A Python Toolbox for Processing Air Traffic Data: A Use Case with Trajectory Clustering", *7th OpenSky Workshop 2019*, 2019, pp. 73-60

[15] X. Olive and J. Morio, "Trajectory clustering of air traffic flows around airports", *Aerospace Science and Technology,* vol. 84, pp. 776-781, 2019/01/01/ 2019. DOI: https://doi.org/10.1016/j.ast.2018.11.031

[16] L. Basora, J. Morio, and C. Mailhot, "A trajectory clustering framework to analyse air traffic flows", *7th SESAR Innovation Days*, Vojvode Stepe 305, Belgrade , Serbia, 2017, pp. 1-8

[17] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis", *Journal of Computational and Applied Mathematics,* vol. 20, pp. 53-65, 1987/11/01/ 1987. DOI: https://doi.org/10.1016/0377-0427(87)90125-7

[18] Y. Li and H. Wu, "A Clustering Method Based on K-Means Algorithm", *Physics Procedia,* vol. 25, pp. 1104-1109, 2012/01/01/ 2012. DOI: https://doi.org/10.1016/j.phpro.2012.03.206

[19] S. J. Corrado, T. G. Puranik, O. J. Pinon, and D. N. Mavris, "Trajectory Clustering within the Terminal Airspace Utilizing a Weighted Distance Function", *Multidisciplinary Digital Publishing Institute Proceedings*, 2020, p. 7

[20] S. Mirjalili, *Evolutionary Algorithms and Neural Networks Theory and Applications*, Studies in Computational Intelligence, 780, 1st ed. 2019. ed., Springer International Publishing, Cham, 2019.

[21] A. Pellegrini, P. D. Sanzo, B. Bevilacqua, G. Duca, D. Pascarella, R. Palumbo*, et al.*, "Simulation-Based Evolutionary Optimization of Air Traffic Management", *IEEE access,* vol. 8, pp. 161551-161570, 2020. DOI: 10.1109/ACCESS.2020.3021192

[22] J. G. Herrero, A. Berlanga, and J. M. M. Lopez, "Effective Evolutionary Algorithms for Many-Specifications Attainment: Application to Air Traffic Control Tracking Filters", *IEEE transactions on evolutionary computation,* vol. 13, pp. 151-168, 2009. DOI: 10.1109/TEVC.2008.920677

[23] W.-C. Hong, Y. Dong, F. Zheng, and S. Y. Wei, "Hybrid evolutionary algorithms in a SVR traffic flow forecasting model", *Applied mathematics and computation,* vol. 217, pp. 6733-6747, 2011. DOI: 10.1016/j.amc.2011.01.073

[24] T. Bäck, *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*, Oxford University Press, Inc., 1996.

[25] H. M. Zubi, R. W. Dunn, F. V. P. Robinson, and M. H. El-Werfelli, "Passive filter design using genetic algorthims for adjustable speed drives", ed: IEEE, 2010, pp. 1-7.

[26] A. Obuchowicz, *Stable Mutations for Evolutionary Algorithms*, Studies in Computational Intelligence, 797, 1st ed. 2019. ed., Springer International Publishing, Cham, 2019.