STATIC SECTORIZATION APPROACH TO DYNAMIC AIRSPACE CONFIGURATION USING APPROXIMATE DYNAMIC PROGRAMMING

Sameer Kulkarni, Rajesh Ganesan, Lance Sherry,
Center for Air Transportation Systems Research, Fairfax, VA 22030

Abstract

The National Airspace System (NAS) is an important and a vast resource. Efficient management of airspace capacity is important to ensure safe and systematic operation of the NAS eventually resulting in maximum benefit to the stakeholders. Dynamic Airspace Configuration (DAC) is one of the NextGen Concept of Operations (ConOps) that aims at efficient allocation of airspace as a capacity management technique.

This paper is a proof of concept for the Approximate Dynamic Programming (ADP) approach to Dynamic Airspace Configuration (DAC) by static sectorization. The objective of this paper is to address the issue of static sectorization by partitioning airspace based on controller workload i.e. airspace is partitioned such that the controller workload is balanced between adjacent sectors. Several algorithms exist that address the issue of static restructuring of the airspace to meet capacity requirements on a daily basis. The intent of this paper is to benchmark the results of our methodology with the state-of-the-art algorithms and lay a foundation for future work in dynamic resectorization.

Introduction

The Joint Planning and Development Office (JPDO) has developed ConOps for the Next Generation Air Transportation System (NGATS). The evolution of these ConOps is envisioned to improve the air transportation industry to maximize the benefits of the air transportation system to the stakeholders. The US airspace is divided into 20 Air Route Traffic Control Centers (ARTCCs) (See Figure 1, blue lines). Each ARTCC is again divided into sectors (Figure 2, red lines). A sector can be considered as a traffic unit, which is usually controlled by one or more controllers. The sectors must be designed in way such that the controller workload across all the sectors is minimum to ensure high operational safety. The current NAS is a rigid structure composed of sectors with a pre-defined geometry. DAC aims at migrating from the rigid airspace structure to a more flexible airspace thus enabling controllers to efficiently manage airspace in heavy traffic and/or severe weather conditions without compromising on safety and workload.

DAC can be achieved by static sectorization and dynamic resectorization. Static sectorization refers to one time redrawing of the sector boundaries to meet today’s average traffic patterns. The limitation with static sectorization is that it is a one-time procedure, usually done at the beginning of a day/month/year based on current traffic data, which often tends to change sector geometry. A change in sector geometry is unfavorable owing to the Human In The Loop (HITL) factor, which is the time taken by the controllers to adjust to the new sector pattern. This calls for dynamic resectorization, which refers to adjusting sector boundaries on a need base criteria, which can be achieved by splitting/consolidating adjacent sectors, thus leaving the sector geometry untouched.

Limited resectorization is currently being practiced based on a certain set of FAA playbook rules, but the real challenge lies in building a decision support tool that will ultimately aid the controllers in sectorization related decisions in conditions of severe weather and/or traffic.

This paper is organized as follows. Section I gives a brief overview of the state-of-the-art static sectorization algorithms Section II Approximate Dynamic Programming (ADP) literature review, Section III describes the ADP methodology, Section IV discusses simulation data and experimental setup, Section V discusses the results of the sectorization algorithm, Section VI Conclusions and Section VII Future Work.
I. State-of-the-art algorithms

There are certain design constraints that an airspace sector should satisfy, namely:

1. The workload constraint, which basically states that the capacity of a sector should be below a maximum threshold i.e. the Monitor Alert Parameter (MAP). This value specifies the maximum number of allowable aircraft in any sector at a given time.

2. The convexity constraint, which is imposed to ensure that an aircraft stays in the sector for considerable amount of time and does not re-enter the sector again. The sector shape should avoid irregularities like jagged edges & acute angles.

3. The boundary constraint, which basically states that the sector boundary should be at a minimum distance from any possible point of conflict viz. waypoints, jet ways intersection points so that the controller has sufficient time to resolve the conflicts.

The literature shows that there are four distinct modeling techniques for designing DAC algorithms,

1. Region-based, which involves decomposing the airspace into small polygonal cells and applying relevant clustering algorithms to cluster these cells to form sectors.

2. Graph-based, which involves graphical methods to partition airspace (graph) into sectors (sub graphs) based on a set of termination criterion, which usually is the number of sub graphs (sectors) to be obtained and the weight (workload after balancing) of each sub graph.

3. Trajectory- based, which involves forming sector boundaries around user-preferred routes while balancing dynamic density levels.

4. Hybrid method, which involves combining two different models mentioned above.

Several algorithms have been developed to address the issue of static sectorization. Yousefi et al [1] designed an integer programming approach to partition airspace based on optimal controller workload. The airspace was discretized into hexagonal cells and the workload value for each cell was determined via simulation. An integer programming-based clustering algorithm was formulated to cluster these hexagonal cells such that the resulting sectors had similar workload values. Brinton et al [2] developed an approach that formed sector boundaries around user-preferred flight routes that were clustered using a clustering algorithm while maintaining acceptable Dynamic Density levels. Xue, M [3] proposed an approach to redesign airspace sector based on Voronoi diagrams (a graph-partition method) and a genetic algorithm that optimizes the sectorization. Martinez et al [4] proposed a weighted-graph approach for partitioning airspace based on peak traffic-counts. The approach involved creating a network flow graph, which was further partitioned into sub-
graphs till a set termination criterion was met. The termination criterion is the desired number of sub-graphs (sectors) and desired weight (peak traffic-count) of each sub-graph. Trandac et al [5] proposed a constraint programming approach to partition airspace into sectors such that the controller workload across all the sectors was balanced while at the same time the coordination workload was minimized. Mitchell et al [6] developed a computational geometry technique to achieve dynamic resectorization. A recursive, top-down partitioning algorithm was used to subdivide a 2D polygonal region of the airspace into sectors. They studied three different techniques of partitioning viz. straight-line cuts, pie-cuts and wheel cuts. The partitioning balanced workload among the resulting sub regions, while maintaining the sector shape parameters within acceptable limits. In all these partitioning methods, while the sector size was acceptable and the workload among the sectors was balanced, there were some corners in these sectors that made them operationally undesirable. Delahaye et al [7][8] developed the genetic algorithm approach to regroup airspace sectors. Klein, A [9] developed a fast algorithm to partition airspace into sectors based on the principle of equalized traffic mass.

The approaches mentioned above suggest that the new sectorizations will be implemented seasonally and up to multiple times a day. The Air Traffic Control system cannot implement these changes with this frequency in the near future (within five years). Literature suggests that implementing changes in sectors takes 6-18 months and training a controller on a new set of sectors requires six months to two years. There is a possibility though to achieve this with significant improvements in automation [10].

II. ADP Literature Review

Decisions made in real life can be classified as static decisions (one-time decisions) and dynamic decisions (sequential decisions). Basically, problems that typically involve making decisions, then observing information derived as a result of the above decisions, after which some more decisions are made and so on are termed as sequential decision problems. A simple example can be that of driving a car to work, wherein decisions related to speed and route are never static.

Linear programming is an effective modeling tool for cases where the decisions made are mainly static. Dynamic Programming can be used to model problems that involve sequential decision-making. Dynamic Airspace Configuration is one such problem that falls under this case. This problem can be solved as a Dynamic Resource Allocation problem. Typically, control theory is classified into optimal, robust, and adaptive control. The literature reviewed here pertains to the model-based and model-free classification of control theory and provides a historical motivation for pursuing the model-free approach.

The Model-Based Controllers

These controllers use two main types of models: differential-algebraic equations and difference equations. The differential-algebraic equation approach has been used for both linear and linear-quadratic optimal control [11], and in control of non-linear systems [12]. Robust control for non-linear systems has been addressed by [13], which in turn reduces to finding a solution to the Hamilton-Jacobi-Bellman equation. In recent years, linear and non-linear hybrid dynamical system has been the focus of research [14][15]. The most popular form of control using difference equations is the Run-by-Run (RbR) controller in which the control laws are obtained from designed experiments and/or regression models. Some of the RbR methods are given in [16]. Model-based simulation techniques have been used for the control of discrete-event dynamic systems but lack closed form solutions [17].

Some primary limitations of above model-based controllers are that 1) they depend on good process models, 2) control actions are based on the parameters of the model, which are often fixed, 3) they cannot handle large perturbations of the system because the system is not intelligent, 4) they need multiple filtering steps to compensate for drifts and autocorrelation, and 5) they lack scalability. One of the ways to handle the above limitations is through a process model-free (data-driven) learning-based control, which is a simulation-based optimization technique.
Process Model-Free (Dynamic Data Driven) Control

Model-free control systems use some form of artificial intelligence such as neural networks, fuzzy-logic rules, and machine learning. They have strong mathematical foundations underlying their construction. Some of these systems, particularly neural networks and fuzzy-logic rules, though are claimed to be model-free, do contain certain hidden or implicit models and make certain strong modeling assumptions when it comes to proving the stability of the controller [18].

Optimal Control And Approximate Dynamic Programming (ADP)

Stochastic approximation methods have been proved effective for control of non-linear dynamic systems. In this method the controller is constructed using a function approximator (FA). However, it is not possible for a model-free framework to obtain the derivatives necessary to implement standard gradient-based search techniques (such as backpropagation) for estimating the unknown parameters of the FA. Usually such algorithms for control applications rely on well-known finite-difference stochastic approximations (FDSA) to the gradient [19]. The FDSA approach, however, can be very costly in terms of the number of system measurements required, especially in high-dimensional problems for estimating the parameters of the FA vector. This led to the development of Simultaneous Perturbation Stochastic Approximation (SPSA) algorithms for FA, which is based only on measurements of the system that operates in closed-loop [20]. Among the several variants and applications of SPSA, the implementation of SPSA in simulation-based optimization using Reinforcement Learning (RL) offers several advantages in solving many stochastic dynamic sequential decision-making problems of which the stochastic control problem is a subset [21].

RL (a strand of ADP) is a method for solving Markov Decision Processes (MDP), which is rooted in the Bellman [22] equation, and uses the principle of stochastic approximation (e.g. Robbins-Monro method [23]. Howard [24] first showed how the optimal policy for a MDP may be obtained by iteratively solving the linear system of Bellman equations. Textbook treatment of this topic and convergent average reward RL algorithms can be found in [25] [26] [27]. The connection between various control theories and ADP is available in [28].

III. ADP Methodology

We design a region-based approach by decomposing the entire airspace into hexagonal cells. Following is the stepwise breakdown of the methodology:

Step 1: Discretize the NAS with hexagonal cells. The current minimum lateral separation in en route airspace is 5nm. Considering this, the side of each hexagonal cell was calculated to be 0.4 deg latitude and longitude, which approximately translates to 24nm. The entire US airspace is bound in a rectangle. The Cartesian coordinates of the top-left corner correspond to 50° latitude and -130° longitude, and the bottom-right corner correspond to 24° latitude and -64° longitude [1]. Figure 3 below shows the decomposition of the NAS into hexagonal cells. In our case we have a 24 x 175 matrix of hexagonal cells (4200 cells).

Figure 3. Hexagonal mesh over the NAS

Step 2: Identify nodes i.e. NAVAIDS and jet ways intersection points. These nodes basically represent the seeds that act as cluster nucleation points. Typically the IFR charts contain information pertaining to jet ways and waypoints (fixes or NAVAIDS). The position of these nodes can also be obtained from today’s flight plan. Figure 4 below describes the same.
**Step 3**: A knapsack algorithm is developed to cluster these hexagonal cells around the nodes identified in Step 2, such that the workload of the clusters is maximized.

The objective of this algorithm is to maximize the cluster workload subject to the total distance constraint for a cluster (i.e. sum of all distances from the node to each grid element of the cluster). This algorithm results in say ‘m’ number of clusters in an ARTCC and a workload value for each cluster. Figure 5 below shows the clusters that are formed around the nodes. It is important to note that the clusters formed in this step can be of any shape. The algorithm is designed such that it forms clusters along heavy traffic routes.

**Step 4**: We then formulate a set-covering algorithm to find the minimum number of seeds, for a given ARTCC, which will basically act as sector nucleation points. The primary purpose of this step is to ensure that the sector is appropriately sized.

**Step 5**: A set-covering algorithm is formulated to combine the clusters obtained in Step 3 to form sectors such that the MAP between adjacent sectors is balanced.

**IV. Simulation data & experimental set up.**

1. Traffic date: 08/19/2007
2. Traffic condition: Light
3. No weather
4. Simulation run for only high altitude sectors (FL240 and above)
5. Simulation run time: 24 hrs.
6. NAS Simulation software: Probabilistic NAS Platform (PNP), provided by Sensis Corporation.

For this project, we employed the PNP software to simulate a day’s worth of traffic data. Sensis Corporation is developing this software. PNP is both a research tool and a decision support platform to analyze air traffic and traffic flow management (TFM) issues. Real-time PNP provides real time meteorological data and traffic advisories in which an operator or Air Traffic Manager can make more informed and accurate decisions. Playback PNP evaluates current or future TFM concepts using projected or actual traffic demands sets with recorded historical meteorological data.

The software calculates the MAP value of each cell based on the size of the cell. These values are the baseline MAP values in our case.

The software, as an output, generates a cell congestion report based on the traffic data used for simulation. The report consists details of cell loading over the entire day for every 15-minute time bin. The maximum load value, for each cell, across all the time bins is considered to be the capacity of the cell. These capacity values for each cell form an input to the clustering algorithm described in Step 3 of the previous section.

**V. Results**

This section discusses the results of the algorithm. The codes were written in MATLAB.

---

Figure 6. MAP matrix for the entire NAS

Figure 7. Algorithm generated clusters in the NAS

Figure 8. ZFW ARTCC clusters
Figure 6 above represents the MAP values for each hexagonal cell in the 24 x 175 matrix. The color scheme on the right is an indication of the cell load or MAP. The white patches, regions of high traffic, above position 140 marks the Indiana, Illinois, Minneapolis and Kansas regions and above position 160 marks the Washington D.C, New York, Boston, Atlanta and Florida regions. Yellow patches above positions 100 and 40 represent the Dallas Fort Worth and the Colorado area.

Figure 7 above shows the results of the clustering algorithm for the entire NAS. The algorithm generated a total of 420 clusters. In this case the clusters were square/rectangle shaped. The area on the figure outlined in blue is the ZFW ARTCC, which will be the main focus of this paper.

Figure 8 above shows an exploded view of the ZFW ARTCC clusters. There are 50 clusters that approximately cover the ZFW area. These clusters are later combined to form sectors.

![Figure 9. ZFW high altitude sectors](image)

Figure 9 above is the sectorization result for the high altitude ZFW ARTCC. We compared the results of our sectorization with the current day sectors, Figure 10 [29], and we found that the location and the approximate size of our sectors match the actual ZFW sectors. We now need to do a mapping between the sectors and the hexagonal cells, which will define the true boundary of the sector. The design of appropriate boundary smoothing algorithms will help in achieving better sector shapes.

VI. Conclusions

This paper determined that DP could be a viable tool to achieve static sectorization. The results of ZFW sectorization were encouraging. Further analysis will be needed to measure any reduction in NAS wide delays after resectorizing other ARTCCs.

VII. Future work

The main challenge of DAC is to address the issue of making dynamic adjustments in the airspace to meet daily fluctuations in traffic triggered by factors such as weather and high demand. Dynamic Resectorization is being practiced by controllers even today but in a very limited manner. While the primary goal would be to build a Decision Support Tool (DST) that would assist ARTCC managers to effectively manage airspace capacity, there are two main factors that need to be addressed. Firstly, the static sectorization methods discussed above suggest one time redrawing of sectors to increase airspace capacity while meeting traffic demand in case of a trigger in the NAS. This is not favorable considering the Human-in-the-Loop (HITL) factor i.e. time taken by the Air Traffic Controllers (ATC) to adjust to the new sector geometries. Secondly, the issue of “when to reconfigure the airspace and how often” has not been addressed in great detail. Some algorithms for dynamic resectorization have been developed recently [10].
The solution to the dynamic resectorization problem can be solved using the Learning-based Stochastic Dynamic Programming principles a.k.a Reinforcement learning or Approximate Dynamic Programming. The objective of this algorithm would be, to basically study the optimal decision to be taken for a given value of Dynamic Density \( D \) workload of a sector. The challenge of when and how often to resectorize, in conditions of inclement weather and/or high traffic, will be answered by studying the results of multiple simulations with different values of look-ahead times, trigger notification times and time taken to adapt to new sector boundaries.

Reconfiguration of airspace has certain benefits associated with it but again it has its associated costs. This necessitates the study and understanding of the mechanisms that trigger reconfiguration and the extent under which, in effect of these triggers, the airspace would have to be reconfigured. Yousefi et al [31] present a detailed study of the trigger metrics for DAC.

While DAC deals with capacity management, Traffic Flow Management (TFM) deals with demand management and it would be important and interesting to study the interaction between DAC and TFM in detail. While both the initiatives have their own benefits, they do have certain implementation costs associated with them. It would be of prime importance to seek a break-even point that would minimize costs thus maximizing the benefits of the combined system. An interesting study by Hunter [32] addresses this issue.

References


Acknowledgements

I would like to thank Dr. Rajesh Ganesan, Dr. Lance Sherry and all other faculty and student members at the Center for Air Transportation Systems Research for their guidance and continued support in my ongoing research.

I would also like to thank Dr. George Hunter, Mr. Benjamin Boisvert and their team at Sensis Corporation for the PNP software. I would also like to thank them for their timely cooperation and professional inputs in my ongoing research.

I would also like to thank my dear friend, Aditya Mugali, for his help in all possible programming aspects.

2011 Integrated Communications Navigation and Surveillance (ICNS) Conference

May 10-12, 2011