Dynamic Sectorization based on a Column Generation Approach

Tambet Treimuth, Daniel Delahaye, Sandra Ulrich Ngueveu

ENRI Int. Workshop on ATM/CNS. Tokyo, Japan (EIWAC2015)

[EN-A-055] Dynamic Sectorization Based on a Column Generation Approach

+ T. Treimuth * D. Delahaye * S. Ullrich Ngueveu **

* French Civil Aviation University (ENAC)
Toulouse, France
[treimuth | daniel]@recherche.enac.fr

** LAAS/INP Toulouse
Toulouse, France
sungueve@laas.fr

Abstract: This paper presents a method for solving a dynamic sectorization problem. It is important because the airspace is limited and the number of aircraft in the airspace increases. The number of aircraft in the airspace cannot be increased due to unbalanced workload of air traffic controllers. It is unbalanced because fixed sectors are used. Airspace capacity can be increased if the sectorization is more flexible. The sectorization must be changed according to the traffic. We use column generation algorithm to solve dynamic sectorization problem. Column generation refers to linear programming (LP) algorithm designed to solve problems in which there are a huge number of variables compared to the number of constraints and the simplex algorithm step of determining whether the current basic solution is optimal or finding a variable to enter the basis is done by solving an optimization problem rather than by enumeration Column generation problem consists of the master problem and sub-problem. The master problem focuses on choosing the best configuration for each period, among a restricted list of configurations available, taking into consideration the static and dynamic costs.

Keywords: Airspace Sectorization, Optmization, Column Generation,Resectorization

1 INTRODUCTION

Fixed airspace structure is one obstacle why more aircraft cannot fly in the airspace. The sector geometry has not changed even if routes have changed over the past decade. Dynamic Airspace Sectorization is a principle where the airspace is redesigned dynamically to take into account changing traffic demands. Some solutions have been proposed to dynamically partition the airspace to accommodate traffic growth. The problem is that their complexity increases as the airspace size and traffic volume increase. Yousefi proposed method for resectorization but it was too slow for implementing it in real time [1]. Tien proposed method to avoid sector boundary changes during busy periods [3] A mixed-integer programming (MIP) model was developed [2]. In this paper column generation algorithm to solve dynamic sectorization problem. Column generation algorithm is used to solve problems in which there are a huge number of variables.

2 MATHEMATICAL MODEL

2.1 Principle

The master problem focuses on choosing the best configuration for each period, among a restricted list of configurations available, taking into consideration the static and dynamic costs. Each master problem is solved with a linear solver. The pricing (also called subproblem) uses dual variables values to generate configurations of negative reduced cost that need to be added to the restricted list of configurations for the master problem. In this implementation we propose to solve the pricing heuristically with graph partitioning algorithm designed for a fast and efficient 1 period partitioning.

2.2 Model

Let C be the set of feasible configurations applicable to any time period, let c_i^t be the static cost of a configuration $i \in C$ at time t and let:

- X_i^t be equal to 1 if configuration i is used at time t and 0 otherwise,
- Y_e^t be equal to 1 if edge e is frontier at time t and 0 otherwise,
- Z_e^t be equal to 1 if edge e was not frontier at time $t-1$ but became frontier at time t, or if edge e was frontier at time $t-1$ and is not frontier at time t. The variable is $= 0$ if the edge status (frontier or not) has not changed from $t-1$ to t.

The resectorization problem can be reformulated as follows:

$$(\mathrm{P_{CFB}})\ \min f = \sum_{t=1}^{T}\sum_{i\in C} c_i^t X_i^t + \gamma \sum_{t=1}^{T}\sum_{e\in E} Z_e^t \quad (1)$$

s.t.

$$\sum_{i\in C} X_i^t = 1, \qquad \forall t \in \{1..T\} \quad (2)$$

$$Y_e^t - \sum_{i\in C(e)} X_i^t = 0, \qquad \forall t \in \{1..T\}, \forall e \in E \quad (3)$$

$$Z_e^t - Y_e^t + Y_e^{t-1} >= 0, \qquad \forall t \in \{2..T\} \quad (4)$$

$$Z_e^t - Y_e^{t-1} + Y_e^t >= 0, \qquad \forall t \in \{2..T\} \quad (5)$$

$$Y_e^t \in \{0,1\}, \qquad \forall t \in \{1..T\}, \forall e \in E \quad (6)$$

$$X_i^t \in \{0,1\}, \qquad \forall t \in \{1..T\}, \forall i \in C \quad (7)$$

$$Z_e^t \in \{0,1\}, \qquad \forall t \in \{2..T\}, \forall e \in E \quad (8)$$

where γ is a predefined weighting factor and $C(e) \subset C$ is the subset of configurations that use edge e as a frontier.

The objective-function (1) minimizes the sum of the static and dynamic costs. The static cost is given by the configurations in use at each instant (for which α_i^t will quantify the weighted sum of the total load, load difference, ...) whereas the dynamic cost is given by the number of edges that change status normalized by β. Constraints (2) ensure that one configuration is chosen per time period. Constraints (3) link the status of each edge (frontier or not) with the set of configurations ($C(e)$) in which the edge is a frontier. Constraints (4) and (5) ensure that the status changes between consecutive instants are correctly computed for each edge.

The dynamic cost is defined in terms of frontier edge in this model, instead of sectors as in the CB model, to avoid computing the transition cost between all pairs of existing (and also newly added) configurations.

It can be proven that only the binarity of variables Y_e^t is mandatory in CFB and that the other two variables can be considered continuous between 0 and 1. Therefore constraints (7) and (8) can be replaced with (9) and (10). Such modification reduces the number of binary variables and therefore the number of potential branches if a branch-and-price is applied. Also, branching on variables Y_e^t is compatible with the pricing method of the column generation, which means that there is no need to develop a second pricing method.

$$0 \le X_i^t \le 1, \qquad \forall t \in \{1..T\}, \forall i \in C \quad (9)$$

$$0 \le Z_e^t \le 1, \qquad \forall t \in \{2..T\}, \forall e \in E \quad (10)$$

2.3 Resolution: Column generation / Branch-and-Price

Model P_{CFB} has a polynomial number of constraints but an exponential number of variables, therefore a column-generation resolution method is appropriate. This method iterates between two phases: a master problem resolution and a pricing resolution (or also known as subproblem resolution or generator). The master problem solves the linear relaxation of the problem restricted to a predefined set of variables. The pricing uses the optimal dual variables values from the master problem to identify/generate new variables of negative reduced cost to add to the master problem, before the latter is solved again. The algorithm stops when no variable of negative reduced cost was generated.

The restricted master problem can be expressed as follows:

$$(\mathrm{RMP_{CFB}})\ \min f = \sum_{t=1}^{T}\sum_{i\in \overline{C}} c_i^t X_i^t + \gamma \sum_{t=1}^{T}\sum_{e\in E} Z_e^t \quad (11)$$

s.t.

$$\sum_{i\in \overline{C}} X_i^t = 1, \qquad \forall t \in \{1..T\} \quad (12)$$

$$Y_e^t - \sum_{i\in \overline{C}(e)} X_i^t = 0, \qquad \forall t \in \{1..T\}, \forall e \in E \quad (13)$$

$$Z_e^t - Y_e^t + Y_e^{t-1} >= 0, \qquad \forall t \in \{2..T\} \quad (14)$$

$$Z_e^t - Y_e^{t-1} + Y_e^t >= 0, \qquad \forall t \in \{2..T\} \quad (15)$$

$$0 \le Y_e^t \le 1, \qquad \forall t \in \{1..T\}, \forall e \in E \quad (16)$$

$$0 \le X_i^t \le 1, \qquad \forall t \in \{1..T\}, \forall i \in \overline{C} \quad (17)$$

$$0 \le Z_e^t \le 1, \qquad \forall t \in \{2..T\}, \forall e \in E \quad (18)$$

where $\overline{C}$ is the restricted list of configurations considered. Each restricted master problem can therefore be solved with a linear programming solver.

Let $u^t, \forall t \in \{1..T\}$ be the dual variables associated with constraints (12) and let $v_e^t, \forall t \in \{1..T\}, \forall e \in E$ be the dual variables associated with constraints (13). The pricing aims at generating new configurations of negative reduced costs, which can be added to $\overline{C}$.

The resulting column generation procedure is summarized in the pseudo-code 1.

Algo 1 Column Generation-based resolution method for the CFB-model

Require: graph G, dynamic coefficient β
1: *ImportData*(G, c, Q)
2: $\forall t \in \{1..T\}, u^t := 0$
3: $\forall t \in \{1..T\}, \forall e \in E, v_e^t := 0$
4: $f := 0$
5: **for** each time period $t \in \{1..T\}$ **do**
6: *GenerateBestConfigurations*(G, c, Q)
7: **end for**
8: *InitializeMasterProblem*$(\overline{C})$
9: stop := 0
10: **repeat**
11: *SolveMasterProblem*$(\overline{C}, f, u, v)$
12: **for** each time period $t \in \{1..T\}$ **do**
13: *UpdateEdgesReducedCosts*$(G, \tilde{c}, u, v)$
14: *GenerateBestConfigurations*$(G, \tilde{c}, Q)$
15: **end for**
16: **if** $\exists$ Configurations of Negative Reduced Cost **then**
17: $\overline{C} = \overline{C} \cup$ {a few configurations}
18: **else**
19: stop := 1
20: **end if**
21: **until** stop == 1
22: **return** f

2.3.1 Generate Best Configurations

Given the time period t, the graph G and the edge costs (c or $\tilde{c}$) as well as the node weights and group capacity Q, the goal is to design/identify the best configurations. It corresponds to the pricing phase of the column generation. If the pricing is solved optimally, then the optimality of the final solution is guaranteed once no configuration of negative reduced cost can be found.

We propose to use a graph partitioning to solve the pricing. As a consequence, the optimality of the final solution is not guaranteed, but valid lower bounds and feasible solutions of good quality can be obtained.

2.3.2 Initialize Master Problem

Using the set of configurations generated $\overline{C}$, create the first restricted master problem (11)-(18).

2.3.3 Solve Master Problem

Solve the restricted master problem with a linear solver such as Cplex or Gurobi (or Scip). Get the dual variables values u^t and v_e^t associated with constraints (12) and (13).

2.3.4 Update Edges reduced costs

The graph weights must be updated to ensure that the application of graph partitioning on the modified graphs generates the configurations with the best reduced costs. Let $\tilde{c}_i^t$ denote the reduced cost of a configuration i for period t and let $\mathcal{F}(i)$ denote the list of edges that are frontier in configuration i. By definition, $\tilde{c}_i^t = c_i^t - u^t - \sum_{e \in \mathcal{F}(i)} v_e^t$ where u^t and v_e^t are the dual variables values associated with constraints (2) and (3) respectively.

For a given time period t, u^t is constant and has the same value for all configurations, therefore finding the configuration that has the best reduced cost is equivalent to finding the configuration i with the best $c_i^t - \sum_{e \in \mathcal{F}(i)} v_e^t$ value, and then substracting u^t. Since $c_i^t = \alpha loaddifference + \beta \sum_{e \in \mathcal{F}(i)} c_e^t$, then $c_i^t - \sum_{e \in \mathcal{F}(i)} v_e^t = \alpha loaddifference + \beta \sum_{e \in \mathcal{F}(i)} c_e^t - \sum_{e \in \mathcal{F}(i)} v_e^t = \alpha loaddifference + \beta \sum_{e \in \mathcal{F}(i)} (c_e^t - \frac{v_e^t}{\beta})$. As a consequence, finding the configuration that has the best reduced cost on the graph G with edge costs c_e is equivalent to finding the configuration that has the minimum total cost on graph G with modified edge costs $\tilde{c}_e^t = c_e^t - \frac{v_e^t}{\beta}$ and then substracting u^t from that total cost.

Therefore, it suffice to apply graph partitioning on the graph with modified edge costs $\tilde{c}_e^t$ and then substracting u^t from the total cost (on the modified graph) of the configuration obtained.

3 PRICING

3.1 Graph Partitioning

Consider a simple weighted graph $G = (V, E)$, where $V = [1, n]$ is the set of vertices and E is the set of edges. Denote by w_{ij}, the non-negative weight of the undirected edge $ij \in E$; if $ij \notin E$, then $w_{ij} = 0$. Let $v_i$be a positive weight of vertex $i \in V$ and $v(A) = \sum_{i \in A} v_i$, where $A \subseteq V$ The goal of the k-partititoning problem is to find a partition of V into a family of k disjoint nonempty subsets $p(\pi)_{1 \leq k \leq p}$ while enforcing the following [4]:
1) w_{ij} is minimized (edgecut)

$$\sum_{i \in \pi_p \Longrightarrow j \notin \pi_p}^{n} w_{ij}$$

2) is minimize (balanced objective)

$$\max_{p \in [1,k]} |v(\pi_p) - \frac{v(V)}{k}|$$

3.2 Multi Level Algorithm

The multi-level graph partitioning algorithm is used for reducing the size of the graph by merging vertices and edges at different levels and then partitions the smallest graph and then reconstructs initial graph structure See figure 1. At each level the partition is

Table 1 Computation Time

Test	Nodes	Parts	Computation time [s]
1	64	4	3.02
2	128	4	8.2
3	256	4	10.5
4	64	8	2.7
5	128	8	7.4
6	256	8	8.8
7	64	16	2.5
8	128	16	7.3
9	256	16	8.0

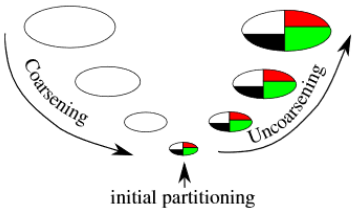

Figure 1 Example of multilevel framework for a 4-partitioning problem [4]

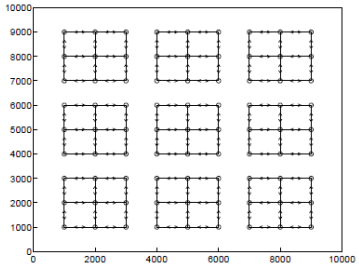

Figure 2 Test Network [6]

refined. The the multi-level graph partitioning algorithm is used for partitioning and the Fiduccia Mattheyses algorithm for refining the partition at each level.The coarse graph is a replica of the original graph.

4 RESULTS

4.1 Input Data

To evaluate proposed algorithm a test network for which an evident solution is known is used. See Fig. 2

4.2 Experiments

Proposed were tested with different network sizes and different number of parts. See Table 1

5 CONCLUSION

This paper presented a method for solving a dynamic sectorization problem. Column generation method is useful for large scale optimization problem. Column generation problem consisted of the master problem and sub-problem. The master problem focuses on choosing the best configuration for each period, among a restricted list of configurations available, taking into consideration the static and dynamic costs.

6 FUTURE WORK

This method should be tested with real traffic and sectors that are used today in the airspace.It can solve more quickly than other MIP solvers. More precise model with additional variable could be developed that takes into account more constraints.

References

[1] Yousefi, A., Donohue, G. L. Temporal and spatial distribution of airspace complexity for air traffic controller workload-based sectorization. In AIAA 4th aviation technology, integration and operations forum, Chicago, 2004

[2] Kopardekar, P., Bilimoria, K., Sridhar, B Initial concepts for dynamic airspace configuration 7th AIAA aviation technology, integration and operations conference (ATIO), Belfast, Northern Ireland,2007

[3] Tien, A., Hoffmann, R. Optimizing airspace sectors for varying demand patterns using multicontroller staffing. 8th USA/Europe air traffic management research and development seminar,2009

[4] Chevalier C, Safro I Comparison of coarsening schemes for multilevelgraph partitioning Learning and Intelligent Optimization: Third International Conference, Berlin, Germany,2009

[5] Desaulniers G,Desrosiers J, Solomon M. Column Generation Springer Science and Business Media, 20. 3. 2006

[6] Modeling and Optimization of Air Traffic Delahaye D., Puechmorel S,2013