Optimal routing for dynamic ATFM

Olivier RICHARD, Rémy FONDACCI

Abstract— In this paper we study the problem of optimal routing in the global context of dynamic ATFM for the European Airspace. The modelling and the algorithms described below have been developed in order to feed with trajectories a global optimization problem in accordance with the column generation technique. However, the method can be used independently to quickly build feasible 4-dimensional trajectories that take into account not only costs of flight (fuel consumption, arrival time…) but also dynamic costs of using control resources. In the first part we describe the general context of dynamic ATFM, the method of column generation and the resulting constraints on the shortest path problem we try to solve. The modelling of the network and of the sectors is detailed in the second section and the algorithms used to build set of feasible trajectories are introduced in the third one. Finally the first experimental results conducted on a modelling of the whole European network with real data from the CFMU are given with the conclusion.

Index Terms— Dynamic ATFM, column generation, dynamical and constrained shortest path problem

I. OPTIMAL ROUTING IN THE CONTEXT OF DYNAMIC ATFM

As the European provider of ATFM (Air Traffic Flow Management), the CFMU (Central Flow Management Unit) is an interface between air traffic services and aircraft operators, trying to share limited airspace control resources between flights while keeping in mind interests of both stakeholders: smoothing of the traffic flow, protection against overload and minimization of congestion consequences. The main regulation tools are ground delays considered as safer and less expensive than air delays. However, tactical ATFM, plagued by uncertainty on the air system, fails to achieve completely its objectives with overloads that still occur and unnecessary allocated delays, especially when demand is close to capacity. Dynamic ATFM is intended to improve regulation results. It does not come instead of tactical ATFM but in addition to it and it consists in taking regulation actions shortly before detected saturations. The envisioned regulation actions on airborne flights (and not only on on-ground flights) are: speed control, air delay (temporal actions), rerouting and flight level changing (spatial actions). Dynamic ATFM raises numerous questions both theoretical and organizational as for example: who takes the decisions; which decisions for which goals; which timing of regulation etc. In this paper we focus on a part of an optimization process designed to find dynamic regulation actions in a global approach at a European scale which is the pertinent level of regulation as showed by the history of air traffic flow management. Further investigations of other questions or on the whole optimization process can be found in previous works of the LICIT laboratory on dynamic ATFM [1].

In the hypothesis of centralised dynamic ATFM, regulators would have to determine for each airborne flight a feasible 4-dimensional trajectory at each regulation step in order to avoid any predicted sector overload. This is a complex and global optimization problem of huge size. We proposed [2] a solving method based on the column generation technique [3]. The problem consists in choosing for each flight a 4-dimensional trajectory (equivalent with choosing a set of regulation actions). With many possible trajectories and many flights, the task of enumerating all trajectories and choosing one among them is impossible to perform in decent time. The idea of column generation is to limit the trajectories for each flight (for example only two different roads or no delay combined with rerouting etc.) in order to enable standard optimization software to allocate a regulated trajectory to each flight. Then the set of ignored trajectories is explored in the light of results of the first allocation and new trajectories can be suggested to the optimization software which decides a new allocation. This is an iterative process that stops when no new interesting trajectories are found. The point of this paper is the determination of these trajectories for one flight and at one given regulation step.

They are numerous constraints that trajectories must satisfy: they have to respect usual navigation rules such as air routes and flight level rules or opening patterns. Aircraft must be able to follow the trajectory within their own performance limits (maximal climbing rates, speed…) and load. In the context of dynamic ATFM, trajectory modelling has to be as accurate as possible in order to get precise entry time in each crossed sector. Third and fourth dimensions (flight level change and temporal dimension) are then particularly important in the modelling.

The cost structure that allows evaluating a trajectory consists of two parts: the cost for the aircraft operator and the cost of overload for the air control services. Fuel consumption, cost of delay for the passengers of the flight and consequences of delay on connected flights are assessed in the first part. The use of air sector control resources are assessed...
in the second part. The dynamical cost of control resource depends on the sector and on the entry time into the sector: the more saturated a sector, the higher the price to cross it. This cost is given by the global optimisation program that assigns trajectory to flights through so called “reduced costs”. In the next section we will develop the idea behind control resources costs. To conclude this part, the problem we discuss is a constrained dynamical shortest paths problem on a continuous graph. There is one instance per flight, which means numerous instances, but once the reduced cost of using airspace control resources are known, each instance can be handled separately and at the same time in a parallel system. Next part is dedicated to the network and air sectors modelling and to the costs of control resources.

II. MODELLING OF THE NETWORK AND OF THE AIR SECTORS

A. Network modelling

The reasons that shaped the network modelling are the following. First, the problem couldn’t fit into a classical graph representation mainly because of the flight level changes which are continuous, so the representation of the third dimension (altitude) had to be continuous. Second, we worked with the context of air routes (contrary to free flight) and in this context route changing occurs above beacons. Third, arrival time in each sector had to be precisely estimated during the computation of a trajectory in order to have a good estimation of the resource capacity needed by each trajectory. Finally, the modelling had to be able to support the dynamic nature of the air network (opening scheme, conditional routes etc.). Thus a space representation using what we will call “slices” has been developed, this representation respect all condition we have just listed. A slice is a two dimensional part of airspace (see Fig. 1) between two consecutive beacons of an air route. It contains the following data: the authorized flight level series (including the maximum and minimum flight levels), the sector frontiers, the opening scheme (directly linked to the opening scheme of the route the slice belongs to) and the next slices that can be used by plane exiting the slice. Each component of the slice has a temporal validity. The set of all slices represents the global air routes network. Each slice is also the node of a classical graph which will be used in the computation of trajectories. One slide represents only one direction of a route, a bidirectional route will be modelled with two distinct slides. However two different routes that share the same portion between two beacons as for example route X and route UX (Upper X) will be modelled on the same slice.

B. Costs of control resources

When assessing the cost of a trajectory, we do not try to add the real and money cost of air traffic control (including air traffic control staff wages or computer equipment costs for example) but costs caused to other flights by the utilisation of air resources. In fact a sector overload forces other flights to use this sector to wait or to reroute with consequences on fuel consumption, arrival time and finally on global cost (i.e. the sum of all chosen trajectories costs). So for each flight that really crosses this sector, driving it to its capacity limit, there is an external cost on other flights and this is precisely what we take into account with the control resources costs. As stated in the first part these costs are calculated through the global optimization problem which is a linear programming instance: they are the reduced costs associated to the constraints. We can give some characteristics of these costs. First a sector-period which has not reached its capacity limit, there is an external cost on other flights and this is precisely what we take into account with the control resources costs. As stated in the first part these costs are calculated through the global optimization problem which is a linear programming instance: they are the reduced costs associated to the constraints. We can give some characteristics of these costs. First a sector-period which has not reached its capacity limit in the solution given by the solving of the global optimization problem will have a resource control cost of zero. Next the resource control cost will depend on the number of flights scheduled to use the sector period but also on the number of alternatives. The algorithm we wrote does not depend on the origin of the control resources costs, so empirical costs based on experience of air regulators could be used outside of a global optimization context to get alternative routes to propose to aircraft operators for example. In our application control resources costs are directly linked to capacity constraints that will then influence the cost of a trajectory, the capacity constraints we chose are then explained next.

For the CFMU a sector capacity is defined as the maximal number of flights allowed to enter the sector in an hour. We took into account this definition in our modelling but the smoothing of the traffic flow, which is a main goal of the CFMU, cannot be insured with this sole constraint so we added capacity constraints on shorter time intervals. The value of these new constraints is derived from the value of the hourly capacity. (see Fig.2). For example if the hourly capacity of a sector is 40 flights, we force the traffic to respect this limit but we add a limit of 12 flights per

---

Fig. 1. slice of airspace between LISTO and HON
quarter. We can note that CFMU smooths the traffic in its regulation algorithm by dividing a period into slots to allocate [4], the number of slots deriving from the capacity, but we have not use this approach in our solving process because of the complexity raised by the rerouting and flight level changes.

In summary of this part, we built a model of the network by defining a graph with complex nodes (slices of space) and a cost structure, shaped by the constraints of our global optimization process but that can be used independently. We are going to use this model to compute alternatives trajectories for flights as described in the next section.

III. SOLVING OF THE OPTIMAL ROUTING PROBLEM AND RESULTS OF COMPUTATIONAL EXPERIMENTS

The computation of 4-dimensional trajectories in our method is a two step process. First we handle the spatial dimension with the construction of a set of feasible 3-dimensional trajectories thanks to the slices model of the network. Second we add the temporal dimension by exploring consequences of speed control on selected trajectories, with the goal of determining optimal entry time in crossed sectors within the flight and network context (due arrival time, connected flights, fuel consumption and dynamical cost of air sector control resources).

A. Determining the spatial route

We will call a partial trajectory a trajectory from the initial position of the flight to a given beacon and a complete trajectory a trajectory from the initial position to the destination airport. The heart of the algorithm is the extension of partial trajectories with the crossing of a slice: for a set of partial trajectories arriving at the first beacon of a slice we can compute a new set of trajectories at the exit of the slice using static and dynamic data on the slice and on the aircraft (weight, type…) by considering different flight level strategies during the crossing of the slice (see fig. 3). Each partial trajectory is coupled with a time interval, the lower bound linked to maximal speed and the upper to minimal speed.

![Fig. 2. Capacity constraints for the sector LFRZS and smoothing of the traffic flow](image)

![Fig. 3. extending trajectory with the crossing of a slice](image)
The algorithm that determines the set of interesting 3-dimensional trajectories follows this pattern: At first there is one flight inside a slice (determined by the current position of the plane), then a set of partial trajectories at the exit of the slice is computed with the different flight level possibilities. A slice is chosen among successors of the first slice to compute a new set of extended trajectories. Then, the classical step of the algorithm is: a slice is chosen, a new set of extended trajectories is computed from the trajectories at the entry of the slice. Bad trajectories are discarded, that means trajectories that do not respect constraints or that are too expensive. This is based on labelling algorithm for shortest path problem, with a sophisticated labelling process (extension of partial trajectories). A lowest bound of the distance from each slice to each airport has been pre-processed with a classical Dijkstra algorithm on a simplified network (with each existing route no matter the flight level or the opening pattern) which allows a better estimation of the lower bound of the arrival time during the algorithm and, following, a faster pruning of the exploration tree. The choice of the next slice at the pruning step of the algorithm is a powerful tuning device with big influence on computational time. For this choice we could use rules from labelling algorithms for shortest path problem (Dijkstra, A*...) but we chose our own rules because the goal of the algorithm is not to determine the best trajectory but to build a set of trajectories, including the best. Any rule of selection can easily be implemented, should the objectives change; besides the complexity of the algorithm does not depend on this rule. During the determination of a three dimensional trajectory, all data needed to determine the optimal speed on this trajectory (minimal and maximal travel time from sector to sector, associated fuel consumption...) are stored and then used in the second part of the algorithm.

B. Computing the optimal speed control

Slices, which are defined from beacon to beacon, are the good tool to determine a three dimensional route which has to follow predefined air routes. However, the important point when dealing with temporal aspect is not flying over beacon but entering a new sector, because the capacities are defined for sector. So all the data collected on the first step of the algorithm are then from sector to sector (easy to obtain as slides contain frontier sectors). The second step of the process consists in determining the best speed control policy for each 3-dimensional trajectory. We perform this task with dynamical programming method. The characteristics of dynamic programming problems are: the problem can be divided into stages with a decision required at each stage, each stage has a number of states associated with it and the decision at one stage transforms one state into a state in the next stage. Given the current state, the optimal decision for each of the remaining states does not depend on the previous states or decisions. In our application the stages of the problem are entry into a new sector, the state associated to a stage is the arrival time at this sector and the decision is the speed to reach the next sector (or similarly the time to reach it). This problem can easily be solved by a classical Bellman algorithm adapted to dynamic programming (see Fig. 4). It is important to note that the cost of a decision take into account the fuel consumption related to speed but also the dynamic cost of using air sector control resources and the arrival time: this explains why the solution is not trivial and the dynamic programming is an efficient tool for this type of problem.

Fig. 4. optimal speed control for a chosen 3-D trajectory by dynamic programming

The two steps of the solving method are independent (as long as the first step store the data needed in the second step). We could envision other algorithms for the second part (dynamic shortest path algorithm with time discretisation for example). In case of lack of computing time the second step can be performed only on a subset of trajectory based on the lowest bound for the total cost of the trajectory computed during the first step. When both steps of the method are performed we get a set of trajectories on the network including the optimal one. It combines aircraft performances with the structure of efficient shortest path algorithms. In the next part we shortly describe first computational experiments and give conclusion on this work.

IV. CONCLUSION

First experiments have been conducted on a network of 21 434 slices representing the whole European airspace during one day and built from real data and for the real planes that flew this day. More than only one trajectory, the algorithm gives all non redundant trajectories (with a limited flight level policy) with cost below a given limit. (see Fig. 5). This limit
will be determined by the global optimization problem: the more attractive the already known routes, the lower the limit used to find new routes. Obviously computing time depends on this limit and on fuel reserve of the plane, but first results are very encouraging with dozens of feasible 4-D trajectories computed in a few seconds. This rough estimate is compatible with the planned application of this algorithm inside a global optimization program which will call it many times during the solving process. The next step of this work, after complementary testing of the algorithm, will be to embed it into the iterative column generation method which we intend to use to solve efficiently the whole optimization problem raised by dynamic ATFM.

REFERENCES


Fig. 5. set of spatial trajectories obtained from slice LATAN-DONIS to airport LGRP