ELSEVIER

Journal of Air Transport Management 13 (2007) 323–328

Journal of
AIR TRANSPORT
MANAGEMENT

www.elsevier.com/locate/jairtraman

Air-traffic complexity resolution in multi-sector planning

Pierre Flener[a,*,1], Justin Pearson[a], Magnus Ågren[a], Carlos Garcia-Avello[b], Mete Çeliktin[c], Søren Dissing[d]

[a]Department of Information Technology, Uppsala University, 751 05 Uppsala, Sweden
[b]Directorate of ATM Programmes, ATC Applications and Systems Division, EuroControl, 1130 Brussels, Belgium
[c]Directorate of ATM Strategies, SESAR and ATM Strategy Division, EuroControl, 1130 Brussels, Belgium
[d]Directorate of ATM Programmes, Communications Systems and Programmes, EuroControl, 1130 Brussels, Belgium

Abstract

This paper considers the problem of minimizing the traffic complexities in an airspace of adjacent sectors. The traffic complexity of a sector is determined by the numbers of flights within it, near its border, and on non-level segments within it. The dimensions of complexity resolution involve changing the take-off times of non-airborne flights, changing the approach times into the chosen airspace of airborne flights by slowing and accelerating within the two layers of feeder sectors around that airspace, as well as changing the altitude at way-points in that airspace. Experiments with European flight profiles from the Central Flow Management Unit show that these forms of resolution can lead to significant complexity reductions and rebalancing.

Keywords: Air traffic complexity; Complexity resolution; Multi-sector planning; Constraint programming

1. Introduction

Air-traffic control operations within any control center rest on a division of airspace into sectors. Because of this fragmentation, the capacity of a control center is limited by the sector with the smallest capacity. Control capacity may be increased by an early identification of traffic complexity bottleneck areas and a reorganization of the traffic patterns so that traffic can be more evenly balanced between sectors. This possibility has stimulated the development of concepts dealing with multi-sector planning (MSP) based on traffic complexity management (TCM), where tools are being developed to predict the complexity of traffic over several sectors and to manage overall complexity by anticipating peaks and developing alternate plans. Traffic complexity in a sector indicates the air-traffic controller (ATC) workload of that sector and is defined here in terms of the numbers of flights within it, near its border, and on non-level segments within it. Each of these positions of a flight requires special attention and procedures to be followed by the ATC. The initial MSP TCM research and development (R&D) activities at EuroControl have focused on complexity measurement and complexity prediction (EuroControl, Directorate of ATM Strategies, Air Traffic Services Division, 2004).

This paper presents some EuroControl R&D results on complexity resolution dealing with the dynamic modification of flight profiles to reduce predicted complexities over a given time interval for sectors. The aim is to avoid excessive peaks in ATC workload as well as to balance the complexities of several adjacent sectors and thereby avoid unacceptable dips of ATC workload and discrepancies between ATC workloads in the various sectors. The concern here is only with en-route flights in the upper airspace that follow standard routes and not free flight.

The tactical rolling-horizon scenario considered has several features. At a given moment, *now*, the complexity manager queries the predicted complexities for an airspace involving several adjacent sectors over a future time interval of 20–90 min. If the look-ahead was shorter than 20 min, there would not be enough time for the computation and implementation of any complexity-resolved flight

*Corresponding author.
E-mail address: Pierre.Flener@it.uu.se (P. Flener).
[1]On leave of absence at Sabanci University, Turkey, during 2006/07.

0969-6997/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jairtraman.2007.05.001

profiles and beyond 90 min, there is too much uncertainty in trajectory prediction. If there are ATC workload peaks, dips, or discrepancies over a time sub-interval $[m_1, \ldots, m_2]$ that warrant interference, then the complexity manager launches a complexity resolution process that changes the current flight profiles over that sub-interval, whose length is 5–10 min. (The average flight time through a European upper airspace sector is about 8 min, hence the proposed resolution window is a trade-off between the time required to find a reasonable resolution and the variations in complexity.) The implementation of a resolution strategy will also have an impact on the evolution of the predicted complexity for later intervals. A minimum fraction, *ff*, of the flights planned in the chosen multi-sector airspace within $[m_1, \ldots, m_2]$ has also to be there under the flight profile settled on. Complexity resolution would otherwise just replan a maximum of flights to be outside all chosen sectors, at the expense of increased complexity in the adjacent sectors and intervals. This process is to be repeated approximately every 10 min. For this to work, the time spent on computing and implementing the complexity resolutions should not exceed these 10 min, and the implementation effort should be offset by the reduction in complexity and in rebalancing among sectors.

Here, there are three possible forms of complexity resolution.[2]

- First, change the take-off time of a not yet airborne flight by an integer amount of minutes within the range $[-5, \ldots, +10]$.[3]
- Second, change the remaining approach time into the chosen airspace of an already airborne flight by an integer amount of minutes, but only within the two layers of feeder sectors around that airspace, at a speed-up rate of maximum 1 min per 20 min of approach time, and at a slow-down rate of maximum 2 min per 20 min of approach time.[4]
- Third, change the altitude of passage over a point in the chosen airspace by an integer amount of flight levels (hundreds of feet), within the range $[-30, \ldots, +10]$, so that the flight climbs no more than 10 levels per minute, or descends no more than 30 levels per minute if a jet, and 10 levels per minute if a turbo-prop.[5]

The objective of the exercise is, given a set S of adjacent sectors, moments $now < m_1 < m_2$, and a fraction *ff*, to modify the profiles of the N flights that are planned at *now* to be inside S within the time interval $[m_1, \ldots, m_2]$ such that minimum $ff \cdot N$ of these flights are still planned to be inside S within $[m_1, \ldots, m_2]$ and such that the complexities of the sectors in S are minimized and, ideally, better balanced. In practice, an allocated amount *timeOut* of computation time is given, and the aim is to get the best flight profile changes within *timeOut* seconds.

The initial assumption is that times can be controlled within an accuracy of 1 min. Indeed, the resolved flight profiles may have new take-off times for some of the flights originally planned to take off after *now*, or new approach times into the airspace for flights already airborne, and the rest of their profiles are shifted accordingly, but the computed optimal complexity holds only if these resolved flight profiles are adhered to by the minute. Semantically, the two kinds of time change amount to a modification of the entry time of a flight into the airspace, in whose sectors the traffic complexities are balanced and minimized. Outside that airspace, the flight profiles have to be updated according to the kind of time change, but we only measure the impacts of time changes within that airspace.

The second assumption is that the flight time along a segment does not change if we restrict flight-level changes over end points to be 'small', as realistically constrained in the third form of resolution. Otherwise, we cannot shift the flight profile by the new entry time, and more time variables would be needed leading to a combinatorial explosion.

2. Air-traffic complexity

Much of the basic work on complexity is found in EuroControl, Directorate of ATM Strategies, Air Traffic Services Division (2004). The complexity of sector s at a moment m is based on: *Traffic volume* (N_{sec} is the number of flights in s at m); *Vertical state* (N_{cd} is the number of non-level, climbing or descending, flights in s at m); and *Proximity to sector boundary* (N_{nsb} is the number of flights that are at most $h_{\mathrm{nsb}} = 15$ nautical miles (nm) horizontally or $v_{\mathrm{nsb}} = 40$ flight levels (FL) vertically beyond their entry to, or their exit from, s, at m). The complexity of sector s at

[2]Many other forms of complexity resolution are possible, such as the horizontal reprofiling along alternative routes (from a list of fixed or dynamically calculated routes) and the introduction of even more time variables than just on the entry into the relevant multi-sector airspace.

[3]Currently, if the air-traffic-management (ATM) system is overloaded, the Central Flow Management Unit (CFMU) imposes slots on aircrafts of 15 min duration and with a $[-5, \ldots, +10]$ min distribution along the slot time. However, a finer definition of the departure time within that slot allows a reduction in the predicted complexity peaks within the system.

[4]The present system lets an aircraft fly its preferred speeds during the cruise phase and most of the descent phase. Typically, the first speed restriction for inbound flights happens below 10,000 feet. There is, however, room on long flights to change the speed of some aircrafts to attain a different future traffic distribution within sectors that reduces traffic complexity. However, due to aircraft aerodynamics and airline cost index management, the speed control range during the cruise phase may be limited but remains significant during the descent phase and interesting during the climb phase.

[5]Aircrafts on crossing routes at the same level or in an overtake situation on the same route contribute significantly to traffic complexity. Separating the aircrafts vertically at an early stage can reduce the traffic complexity perception of the controller by reducing the amount of time needed to monitor a given pair of aircrafts. This technique, level capping, is already used to increase sector throughput. There are cost implications when changing the cruise level of a flight although this can be taken into account by reducing the levels of inbound flights before affecting outbound flights and overflights.

moment m is a normalized weighted sum of these terms:

$$C(s,\ m) = (a_{\mathrm{sec}}N_{\mathrm{sec}} + a_{\mathrm{cd}}N_{\mathrm{cd}} + a_{\mathrm{nsb}}N_{\mathrm{nsb}})S_{\mathrm{norm}},$$

where the sector normalization constant S_{norm} characterizes the airspace structure, equipment used, procedures followed, etc., of s, ensuring that complexity values have a relatively consistent meaning across a wide range of sectors.

The other parameters identified by EuroControl called 'data-link equipage' (indicating whether the ground–air data-link is digital), 'time adjustment' (necessary if the specific flight has a time constraint requiring controller action), 'temporary restriction' (the proportion of the normal capacity of the sector predicted to be available, considering the weather, equipment malfunction, military use of shared airspace, etc.), 'potentially interacting pairs' (the number of flights that will violate horizontal or vertical separation constraints within the sector), and 'aircraft type diversity' (the diversity in aircraft types has an impact on the speeds and heights they fly, and the rates at which they change altitude, thus requiring more monitoring), are not used. 'Data-link equipage', 'time adjustment', and 'temporary restriction' are not included because of data limitations. The 'potentially interacting pairs' parameter is not in strong correlation with the complexity value estimated by the COCA metric (EuroControl Experimental Centre, 2003) probably because 'traffic volume' and 'vertical state' parameters already capture this impact. Finally, 'aircraft type diversity' also showed a weak correlation with the COCA complexity value, though this may be due to the limited amount of data used in the determination of the weights.

The resulting air-traffic complexity measure is maybe simple, but correlates with a model of ATC workload and has more parameters than other metrics actually deployed for complexity resolution (Bertsimas and Stock Patterson, 2000; Sherali et al., 2003).

Moment complexity, however, has a large variance, with steep rises and falls in just seconds. To reduce the probability that the complexity-resolved flight profile just falls into a dip of such a curve, complexity is redefined so that its curve follows a smoother pattern. This is done by deploying a windows averaging technique defining complexity over a given time interval rather than for a specific moment. The interval complexity of a given sector s over a given time interval $[m,\ \ldots,\ m+kL]$ is the average of the moment complexities of s at the $k+1$ sampled moments $m+iL$, for $0 \leqslant i \leqslant k$:

$$C(s,\ m,\ k,\ L) = \frac{\sum_{i=0}^{k} C(s,\ m+iL)}{k+1},$$

where k is the smoothing degree and L the time step between the sampled moments. The interval complexity reduces to moment complexity when k is zero, with the value of L being irrelevant then.

For complexity resolution, the experimentally determined values of the parameters are $k = 2$ with $L = 210\,\mathrm{s}$, indicating that there are three sampled moments—m, $m+210$, and $m+420$—spanning an interval of 7 min. With such values, interval complexity follows a much smoother curve than moment complexity. Values of $k>2$ do not lead to any further significant smoothening and would involve impractically long computation times.

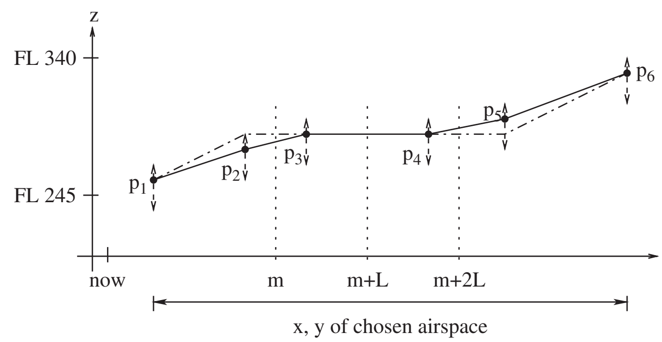

Fig. 1. Planned profile (plain line) and resolved profile (dot–dashed line) that minimizes the number of climbing segments for a flight at m, $m+L$, and $m+2L$.

Any value of $k \cdot L$ differing significantly from 420 s is not in line with average time flights spent in a given segment. How the third form of complexity resolution works, in isolation from the other two forms, can be seen by referring to Fig. 1. For every point, p_i on the planned route (shown as a plain line) for a given flight f within the chosen airspace, there is a maximal range $[-30,\ \ldots,\ +10]$, denoted by a dashed vertical double arrow, of flight levels by which the altitude of passage of f over p_i can be changed, depending on engine type of f and the distances to the previous point p_{i-1} and next point p_{i+1}. Suppose, for $k = 2$, that the moments m, $m+L$, and $m+2L$ are as indicated on the horizontal time axis, namely, respectively on a climbing segment $[p_2,\ p_3]$, on a level segment $[p_3,\ p_4]$, and on another climbing segment $[p_4,\ p_5]$. Complexity resolution tries to level off the two climbing segments, by making $[p_1,\ p_2]$ reach the level h of $[p_2,\ p_3]$, or by making $[p_5,\ p_6]$ start only from h, or by changing h for as many as possible points p_2 to p_5, depending on the lengths of $[p_1,\ p_2]$ and $[p_5,\ p_6]$. Indeed, the sum of the N_{cd} terms for the involved sectors would then decrease by the number of these steeper climbs that are compatible with the climbing performance of f. The dot-dashed alternative route in the figure assumes that both sampled climbing segments could be leveled off. If L is too small, then several moments might fall into the same non-level segment, thereby producing a lot of complexity reduction for a single change. Conversely, if L is too large, then the sampled complexity values concern segments that are too far apart for their average to be useful.

3. Feeder sectors and approach times

The second form of complexity resolution is the slowing down or speeding up of flights in the air, within the two layers of sectors around the chosen multi-sector airspace, to delay or advance their entries into that space. The surrounding sectors can be treated as feeder sectors with a feeder sector s of a flight f being a neighbor of a sector of the chosen airspace, or a neighbor of a neighbor of such a sector, so that f is planned to fly through s before entering the subject airspace.

The time still to be spent in the feeder sectors is the approach time. The approach time of a flight f that has taken off at or before *now* but that has not entered the subject airspace is the duration that f is planned to fly within the feeder sectors between *now* and its entry into that airspace. The restriction to two layers of feeder sectors around the subject airspace aims at generating a reasonable number of updates to the upstream air-traffic control centers for each updated flight profile, as well as at leaving sufficient time to implement the updates. Furthermore, there is too much uncertainty to propagate changes more than two sectors upstream and hope that the new entry time into the airspace will materialize. Finally there is no practical way for an ATC center to modify a flight much in advance of the time it reaches its own airspace.

4. Pareto optimization

We are dealing with a multi-objective minimization problem involving a number of sectors $s_1, s_2, \ldots, s_n$ with the aim of minimizing their complexity with respect to a resolution R: $\langle C_R(s_1, m, k, L), \ldots, C_R(s_n, m, k, L) \rangle$.

Pareto efficiency is a concept that originates in economics and sociology but is now a widely used concept for multi-objective optimization problems within engineering. A vector of complexities is Pareto optimal if no element can be made less complex without making some other element more complex. One of the standard techniques for solving Pareto optimization problems is the combination of multiple objectives into a single one using a weighted summation:

$$\sum_{i=1}^{n} \alpha_i \cdot C_R(s_i, m, k, L)$$

for some weights $\alpha_i > 0$. Different weights α_i produce Pareto-optimal solutions with different tradeoffs; $\alpha_i = 1$ is often used in practice. Although the weighted sum only guarantees to minimize convex parts of the set of Pareto-optimal points, non-convex sets are seldom found. Further, since we are only interested in a resolution that reduces complexity, but not in the structure of the set of Pareto-optimal points, any Pareto-minimal resolution is sufficient.

5. Model

The model is fully parameterized and constituted to allow experimentation with various values of the following parameters: *maxEarly* (resp. *maxLate*)—the maximum amount of minutes that a flight can take off before or after its planned time, a typical value being 5 (10); *maxSlowDown* (resp. *maxSpeedUp*)—the maximum amount of minutes that a flight can be slowed down (or sped up) per 20 min, a typical value being 2 (1); *maxDown* (resp. *maxUp*)—the maximum amount of flight levels by which the altitude of a flight over a point can be decreased (or increased), a typical value being 30 (10); *maxDownJet* (resp. *maxUpJet*, *maxDownTurbo*, and *maxUpTurbo*)—the maximum amount of flight levels that a jet (or turbo-prop) can descend (or climb) per minute, a typical value being 30 (10, 10, 10); *lookahead*—an integer amount of minutes, a typical value being a multiple of 10 in the range [20, …, 90]; *now*—the time at which a resolved scenario is needed with a forecast of *lookahead* minutes; k—the smoothing degree, a good value being 2; L—the time step, a good value being 210; *ff*—the minimum fraction of the sum of the numbers of flights planned to be in the chosen multi-sector airspace at the sampled moments $m + i \cdot L$, for all $0 \leqslant i \leqslant k$, that have to be there in the resolved flight profile as well; and *timeOut*—the maximum number of seconds that should be spent on computations before returning the current best feasible solution.

The model constraints are, for each flight f, for each pair (f, p) of a flight f and one of its waypoints p within the chosen airspace, for each sector s, and for each index i of a moment, the values of the decision variables $\Delta T[f]$ (denoting the entry-time change of f into the chosen airspace), $\Delta H[f, p]$ (denoting the flight-level change of f over p), $N_{\text{sec}}[i, s]$, $N_{\text{nsb}}[i, s]$, and $N_{\text{cd}}[i, s]$, subject to the permitted forms of complexity resolution, such that the sum of the interval complexities of the chosen sectors is minimized (a Pareto optimization with unit weights).

The search heuristic tries to 4D-position each flight f so that it is never near a boundary of any sector s, so as not to increase any $N_{\text{nsb}}[i, s]$, and never on a climbing or descending segment, so as not to increase any $N_{\text{cd}}[i, s]$. If it cannot avoid positioning a flight near a sector boundary, or on a climbing or descending segment, then it selects a sector with a low a_{nsb} or a_{cd} value, unless the *ff* parameter allows it to re-schedule the flight so that it is not in the target airspace at the sampled moment.[6]

6. Experiments

For the experiments, the air-traffic control center in Maastricht, the Netherlands, is used. The chosen multi-sector airspace consists of five sectors, covering the upper airspace of the three BeNeLux countries and some airspace over northern Germany, depicted in Fig. 2 and characterized in Table 1. They are all high-density, en-route, upper airspace sectors (above FL 245). The sector identified by *sectorId* stretches vertically between flight levels *bottomFL*

[6]The model is implemented using the Optimization Programming Language (OPL) (Van Hentenryck, 2002). As the resulting model has non-linear constraints, the OPL compiler translates the model into code for ILOG Solver, rather than for CPLEX, and constraint solving (Apt, 2003) takes place at runtime; see Flener et al. (2007) for technical details.

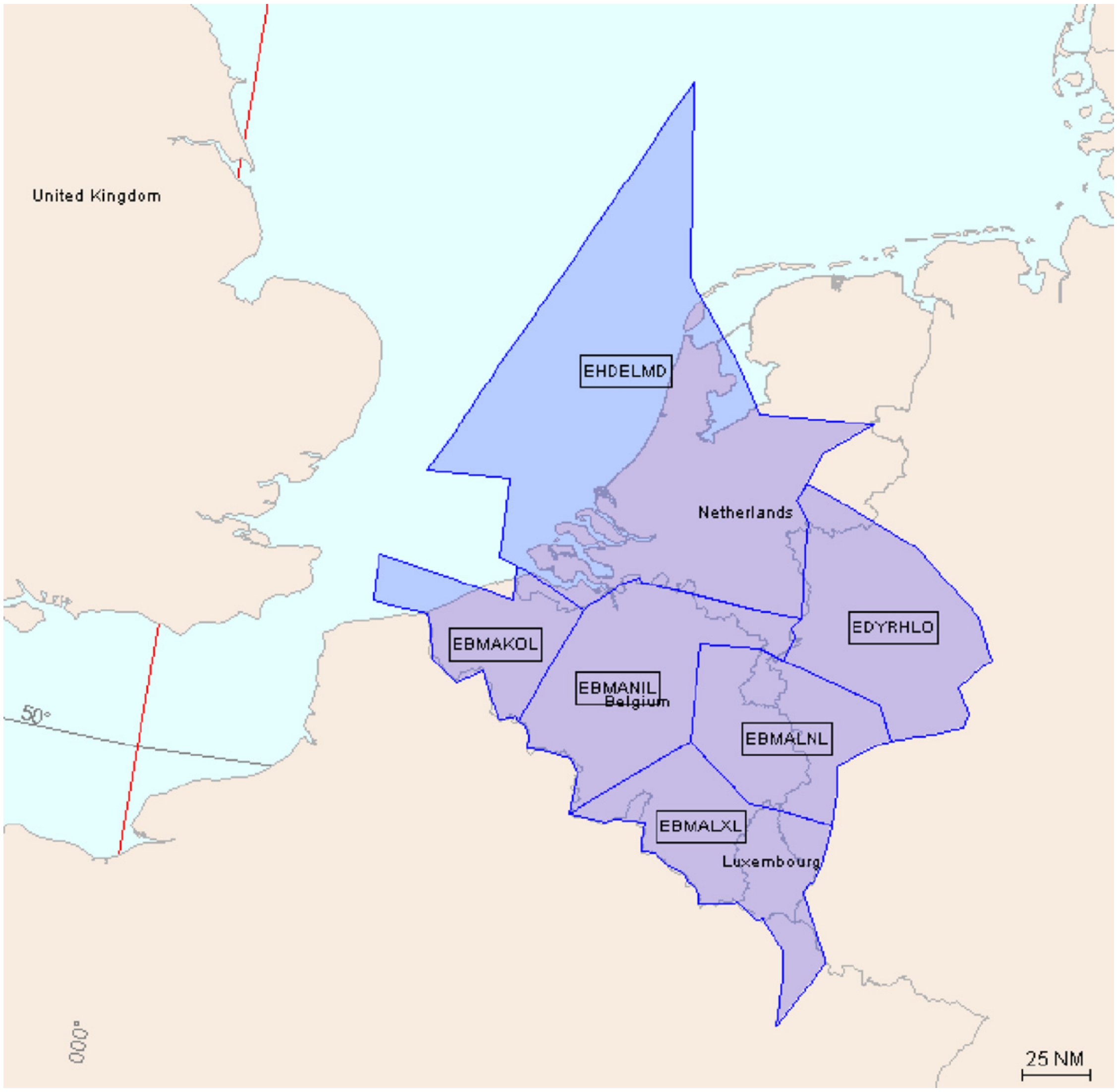

Fig. 2. The chosen multi-sector airspace over Western Europe where sectors EBMAKOL and EBMANIL were collapsed into the sector EBMAWSL.

Table 1
Characteristics of the chosen multi-sector airspace

sectorId	*bottomFL*	*topFL*	a_{sec}	a_{cd}	a_{nsb}	S_{norm}
EBMALNL	245	340	7.74	15.20	5.69	1.35
EBMALXL	245	340	5.78	5.71	15.84	1.50
EBMAWSL	245	340	6.00	7.91	10.88	1.33
EDYRHLO	245	340	12.07	6.43	9.69	1.00
EHDELMD	245	340	4.42	10.59	14.72	1.11

and *topFL*. Unfortunately, none of these sectors is below any other, so that the traffic complexity resolutions cannot consider re-routing a flight through a lower or higher sector. Weights a_{sec}, a_{cd}, a_{nsb} and sector normalization constants S_{norm} of the complexity metric are taken from EuroControl, Directorate of ATM Strategies, Air Traffic Services Division (2004). There are an additional 34 feeder sectors (not listed), for which we only need to know the *bottomFL* and *topFL* values.

The chosen day was 23 June 2004, one of the busiest days of the year, and the chosen hours those with the peak traffic; i.e. from 07:00 to 22:00 local time. The selected flights follow standard routes and use turbo-prop or jet aircraft. The Central Flow Management Unit (CFMU) provided 1798 flight profiles; we statically repaired 761 that included impossibly steep climbs or descents (these would otherwise have to be repaired dynamically during complexity resolution) and discarded 26 flights whose profiles were not repairable. The maximum approach time is 78 min for this sample and can then be changed by integer amounts of minutes within the range $[-4, \ldots, +8]$. This is a sufficiently large range to allow the second form of complexity resolution to lead to significant changes in its own right.

In Table 2, each line summarizes the results of the 180 cases obtained by taking *now* at 5 min intervals between

Table 2
Average planned and resolved complexities in the chosen airspace

lookahead	*k*	*L*	Average planned complexity	Average resolved complexity
20	2	210	87.92	47.69
20	3	180	86.55	50.17
45	2	210	87.20	45.27
45	3	180	85.67	47.81
90	2	210	87.29	44.67
90	3	180	85.64	47.13

07:00 and 22:00 on the chosen day. The average reduction in the average complexity over the five sectors is shown over these instances for various values of the smoothing degree k, the length L (in seconds) of the time steps, and *lookahead*. We kept $ff = 90\%$ of the planned flights in the chosen airspace and used a *timeout* of 120 s. With larger values of *lookahead*, it is possible to get better complexity reductions, because more flights are not airborne and thus offer more opportunities for resolution. With larger values of k (and lower values of L), it is possible to get a slightly better complexity reduction, but with a k of 2, it is possible to get nearly a 50% complexity reduction. The experiments were performed with OPL 3.7 on an Intel Pentium 4 CPU with 2.53 GHz. Most of the computations finished before timing out, or were retrospectively seen (upon a larger value for *timeOut*) to have found near optimal solutions at the moment of timing out, implying that the proofs of optimality were more time-consuming than finding the optima.

Complexity resolution in an MSP context will not be a constraint optimization problem, as here, but rather a constraint satisfaction problem. There will be many additional constraints that have to be satisfied, such as requiring the resolved complexities to be within prescribed bounds. Since CFMU flight profiles derived from the flight plans introduced by airlines are not very accurate (witness the amount of repairs we had to perform) and currently incorporate only an attempt at balancing the numbers of flights (the N_{sec} term of the traffic complexity metric) between sectors, such bounds cannot be imposed on the resolved complexities, as feasible solutions might then not exist. Indeed, there are enormous discrepancies among the planned complexities, and even optimal complexity resolution can often not sufficiently reduce them.

7. Conclusions

Constraint programming offers an effective medium for modeling and efficiently solving the problem of minimizing and balancing the traffic complexities of an airspace of adjacent sectors. The complexity of a sector is defined in terms of the numbers of flights within it, near its border, and on non-level segments within it. The permitted forms of complexity resolution are changing take-off times of not yet airborne flights, changing the approach times into the chosen airspace of already airborne flights by slowing down and speeding up within the two layers of feeder sectors around that airspace, as well as changing the levels of passage over points in that airspace. Experiments with actual European flight profiles obtained from the CFMU show that these forms of complexity resolution can lead to significant complexity reductions and rebalancing.

Acknowledgments

Many thanks to Bernard Delmée, Jacques Lemaître, and Patrick Tasker at EuroControl DAP/DIA, for pre-processing the CFMU raw data.

References

Apt, K.R., 2003. Principles of Constraint Programming. Cambridge University Press, Cambridge.

Bertsimas, D., Stock Patterson, S., 2000. The traffic flow management rerouting problem in air traffic control: a dynamic network flow approach. Transportation Science 34, 239–255.

EuroControl, Directorate of ATM Strategies, Air Traffic Services Division, 2004. Complexity Algorithm Development: Literature Survey and Parameter Identification, Edition 1.0; Complexity Algorithm Development: The Algorithm, Edition 1.0; Complexity Algorithm Development: Validation Exercise, Edition 0.3, Brussels.

EuroControl Experimental Centre, 2003. Pessimistic sector capacity estimation. EEC Note 21/03, Paris, France.

Flener, P., Pearson, J., Ågren, M., Garcia Avello, C., Çeliktin, M., 2007. Air-Traffic Complexity Resolution in Multi-Sector Planning. Technical Report 2007-003 at Department of Information Technology, Uppsala University. Available at ⟨http://www.it.uu.se/research/publications/reports/2007-003/⟩.

Sherali, H.D., Staats, R.W., Trani, A.A., 2003. An airspace planning and collaborative decision-making model: part I—probabilistic conflicts, workload, and equity considerations. Transportation Science 37, 434–456.

Van Hentenryck, P., 2002. Constraint and integer programming in OPL. INFORMS Journal on Computing 14, 345–372.