Planning efficient 4D trajectories in Air Traffic Flow Management[1].

Veronica Dal Sasso, Franklin Djeumou Fomeni, Guglielmo Lulli, Konstantinos G. Zografos

Department of Management Science
Center for Transportation and Logistics (CENTRAL)
Lancaster University Management School
Bailrigg, Lancaster, LA14YX, UK [2]

Abstract

In this paper, we focus on designing efficient 4D trajectories for the planning phase of Air Traffic Flow Management (ATFM). A key feature of the proposed approach is the inclusion of stakeholders' preferences and priorities. In particular, we have implemented two priority mechanisms recently developed by Eurocontrol, namely the *Fleet Delay Reordering* and the *Margins*.

For this purpose, we have customized a multi-objective binary program for the ATFM problem taking into account the specific assumptions required for the ATFM planning phase. To compute the Pareto frontier in a reasonable computational time, we have developed a simulated annealing algorithm. The algorithm has been tested on an instance resembling real world conditions using data extracted from the Eurocontrol data repository. This instance involves four major European airports and their air traffic in one of the busiest days of year 2016, and precisely, October 3rd. The simulated annealing algorithm has shown good computational performances and has provided a good approximation of the Pareto optimal frontier. The results have been validated using Eurocontrol tools and have demonstrated the viability of the proposed approach. Practitioners and stakeholders' representatives have provided positive feedback on the proposed modeling approach and on the inclusion of ATM stakeholders' preferences and priorities.

Keywords: Transportation, Air Traffic Flow Management, 4D trajectories optimization, multi-objective heuristic, multi-criteria decision making.

2010 MSC: 00-01, 99-00

[1]DOI: https://doi.org/10.1016/j.ejor.2019.01.039

[2]Email: veronica.dalsasso@gmail.com, franklin@aims.ac.za, g.lulli@lancaster.ac.uk, k.zografos@lancaster.ac.uk

1. Introduction

Since the advent of commercial air transport, the air transportation system is facing a continuous growth of air traffic demand. Even conflicts and events with a worldwide impact - e.g., the Gulf war, September 11, the 2008 financial crisis - had only a temporary and limited effect on this trend. According to forecasts of Eurocontrol, 14.4 millions flights per year are expected to be operated within the ECAC[3] area by 2035, with a 50% increase with respect to the 2012 number of flights. Since the growth of demand has not been supported by a corresponding development of the capacities of airports and related systems, both the European and the United States air transportation systems are suffering from increased congestion, thus dwindling down the margins of profitability. Indeed, the same forecasts also expect that the current air traffic system will not be able to accommodate the projected air traffic demand of about 1.9 million of flights. Against this backdrop, the need of modernizing the air transportation system is evident. Prominent initiatives have been launched both in Europe (i.e., SESAR) and in the US (i.e., NextGen) to develop the future air transport system that will be more flexible, resilient and scalable than today's one. One of the cornerstones of these initiatives is the implementation of the ICAO Trajectory Based Operations (TBO) concept. The scope of Air Traffic Management (ATM) in a TBO environment is to manage flights' trajectories and their interactions to achieve the optimum system outcome with minimal deviation from the user-requested flight trajectory.A full implementation of the concept requires a high level of automation for all the stakeholders' decision-making processes, which stems from the development of new mathematical models, algorithms and decision support systems. Both SESAR and NextGen have fostered a number of "exploratory" research activities to study several aspects of Air Traffic Management. The OptiFrame research project was part of this effort, aiming to assess the viability of the TBO concept in the planning phase and to determine whether and to what extent the objectives of flexibility and predictability of the Air Traffic Flow Management (ATFM) system can be achieved.

In the planning phase, the scope of Air Traffic Flow Management (ATFM) is to balance the air traffic demand with the air system capacity by adopting control options which consist of assigning to flights either ground delays, alternative routes (reroutings) or both. In the last four decades, ATFM has continuously attracted the interest of the research community. The survey by Vossen et al. [25] provides a thorough review of the literature on ATFM until 2008.

[3]The European Civil Aviation Conference

More recent ATFM research contributions have been published that are relevant for the problem addressed in this paper. That is, "flight-centric" models, which specify the trajectory of each single flight. Bertsimas et al. [5] presented a model that combines accuracy, in terms of ATFM control options, with good mathematical structure to efficiently solve large-sized instances of the problem. The main innovative feature of the model is the formulation of rerouting decisions in a very compact way. The same concept of rerouting has been also used in the model by Augustin et al. [2]. The major difference between the two models lies in the definition of decision variables. Indeed, the former model is node based, meaning that it associates a decision variable to each node of the graph representing the feasible *o-d* routes of a flight, while the latter is edge based. Both these formulations are "compact" formulations. A different modeling approach is the "path" formulation first proposed by Ricard et al. [19] in the ATFM context. Given the large number of variables of these formulations, they are usually solved by means of a column generation approach. Balakrishnan and Chandran [4] designed an algorithm for this class of formulations that is computationally very efficient with good scalability properties. The algorithm solved nation-wide numerical examples from the United States (very large scale instances) in short computational times.

All these models describe the flights' trajectories in a two dimensional geographical space, meaning that no flight level information is attached to the trajectories. In a TBO environment it is fundamental that the information on the flight level is explicitly captured. To the best of our knowledge, the only model focusing on 4D trajectories is the one presented by Sherali et al. [20]. The model prescribes a flight plan for each flight. The flight plan of a flight is selected among a pool of possible candidate plans. The model formulation minimizes an objective function that combines both delays and fuel costs while taking into account air traffic controllers' workload, safety, and equity criteria as constraints.

In this paper, we present a 4D mathematical model for the ATFM planning phase that includes preferences and priorities. In an ideal situation, a preferred trajectory is the trajectory built by the Flight Management System according to the business objectives of the flight operator [21]. However, in the presence of air traffic congestion, preferences can be conceived as a mechanism to absorb delays at tactical level [17]. Priorities refer to the level of importance of a given flight in comparison to all other flights operated by the same airspace user.

The approach herein proposed takes an holistic view of the air traffic system. Indeed, in situations of capacity reductions and/or surges of the demand, with consequent severe demand-capacity

imbalances, it is known that the system-wide optimal ATM strategies might be complex and counterintuitive [16]. Complex because to optimize the performances of the ATM system, it might be necessary to use a mix of ATM control strategies; counterintuitive because efficient ATM solutions may end up in imposing control actions, e.g., delay, rerouting and miles-in-tails, to flights that are not directly affected by ATM constraints. Because relevant information on preferences is not available, we adopt the multi-objective modeling approach presented in [8]. In the planning phase, the number of control options are limited to ground delays, flight level changes and rerouting. Although these assumptions lead to a more compact formulation with respect to those described in the literature, the dimensions of large realistic instances are prohibitive for exact methods as shown in [7]. Therefore, it is important to develop fast heuristic algorithms to approximate the Pareto frontier in reasonable computational time. In this paper, we present a simulated annealing approach to solve the multi-objective ATFM problem. As part of the algorithm design and implementation, we compare and contrast two different neighbourhood structures for the local search routine of the simulated annealing heuristic. The first one mimics stakeholders' preferences and considers limited reroutes; while the second one searches in a larger neighbourhood using a shortest path algorithm as a subroutine. The proposed approach has been tested on instances resembling real world conditions and has been validated using Eurocontrol tools for calculating key performance indicators identified during the First OptiFrame Stakeholders' Workshop [17]. Finally, we perform an analysis of solutions at different levels of aggregation with an initial assessment of two prioritization schemes, i.e., Flight Delay Reordering and Margins, and their impact on the efficient solutions.

In a nutshell, the main contributions of the paper are: *i)* the development of a customized multi-objective mathematical model for ATFM planning that explicitly includes two priority mechanisms, namely Flight Delay Reordering and Margins, recently proposed by Eurocontrol [11]; *ii)* the design and implementation of a simulated annealing algorithm to solve the multi-objective ATFM problem in reasonable computational times; *iii)* the experimentation of the proposed approach on real instances of the problem, verifying the presence of a trade-off between the objective functions and the performance of the algorithm in terms of computational times and approximation of the Pareto frontier; *iv)* the proposal of a method to analyse solutions at different levels of aggregation to facilitate the decision process, as well as an initial assessment of the impact of proposed priority schemes.

The remainder of the paper is organized as follows: Section 2 presents in details the ATFM

planning model with underlying assumptions. Section 3 presents the heuristic algorithm, based on a local search approach. Section 4 presents the case study used to test the algorithm and the computational results, while the analysis of solutions is presented in Section 5. Finally, Section 6 concludes the paper.

2. The ATFM planning model

In this section we present the mathematical model for the ATFM planning phase, whose key feature is the incorporation of Stakeholders' preferences and priorities. Before presenting the model, we first describe preferences and priorities as applied in the ATFM context.

Preferences. A preference is a partial order of feasible options for managing delays at tactical level. These options consist in deviations from the preferred trajectory in terms of either time, flight level, lateral deviation or a combination of them. Because each option has different implications on the total costs of a flight, preferences are specific of the airspace user (AU). However, AUs are in general reluctant to disclose information about preferences, and more specifically about the trade-offs involved among the identified primitives, i.e., delays (time), flight level and lateral deviations. Therefore, in this paper we adopt a multi-objective approach as presented in [7, 8] with the scope of providing to AUs and all stakeholders involved in the decision process all the information about the trade-offs involved with a set of feasible solutions, thus improving the level of awareness of all the parties involved in the decision process.

Priorities. Each flight has its own revenue/cost structure and is more or less sensitive to punctuality. An AU may want to protect specific flights from delays, assigning an higher priority to ensure they depart on time, at the expense of a more consistent displacement of other flights. Within SESAR research activities, the concept of User-Driven Prioritization Process (UDPP) has been developed. The objective of UDPP is to provide airlines with the flexibility to re-arrange their flights [18]. Although initial developments focused on departure slot swapping, the ambition for the SESAR UDPP is to extend the concept of priorities to the planning phase for the full trajectory of flights and not to limit it to slot swapping [11]. On this subject, several mechanisms have been proposed to implement Stakeholders' priorities. In this paper, we focus on two of the most recent schemes proposed by Eurocontrol, namely Fleet Delay Reordering and Margins. The Fleet Delay Reordering mechanism offers AUs the opportunity to assign delays to flights based on their priorities, while the Margins mechanism assigns priorities by applying a *time not before* or/and a *time not after* rule. The interested reader may refer to [11, 18] for further details.

2.1. Formulation

The objective of the mathematical model is to design for each flight a 4D trajectory, taking into account the capacity of the air traffic system while minimizing deviations from the preferred trajectories. In particular, the mathematical model herein formulated is customized for the ATFM planning phase. The planning phase spans a time interval from one or two days to three hours before the departure time of each flight. In this phase, no airborne holding and miles-in-tails (speed variation) are considered as control options. Therefore, the only control options included in the mathematical model are:

1. ground delays,
2. reroutings,
3. flight level changes.

In view of these assumptions, the travel time to fly any en-route sector is fixed for each aircraft. Moreover, the amount of rerouting (route extension) is bounded. The upper bound is specific to the airspace user. Some airspace users may be quite strict, thus accepting only small values of route extension, while others could be more flexible. It is important to highlight, that the assumptions underpinning the mathematical model and its main features have been defined with input provided by relevant stakeholders and practitioners [17].

As quite common in the ATFM literature, in the proposed mathematical model the airspace structure is represented by a graph $G = (V, E)$, where V is a set of nodes and E is the set of arcs. More specifically, the set of nodes represents all the relevant waypoints of the airways system and airports. For clarity of exposition, in the sequel we use the term 2D route for an o-d path on the graph $G = (V, E)$, i.e., the airports of origin and destination as well as the sequence of waypoints along the path. When the information about the flight level (altitude) is included, we use the term 3D route; and finally, we use the term trajectory for a flight route when the time dimension is included. For each flight $f \in \mathcal{F}$, we define the subgraph $G^f = (V^f, E^f)$ that represents all the feasible waypoints (and arcs) that flight f may cross. Without loss of generality, similarly to [5], we assume that the graph is acyclic. The time horizon is discretized in time intervals $t \in \mathcal{T}$. The notation used to formulate the model is here listed.

- $\mathcal{T}$ set of time periods,
- $\mathcal{S}$ set of sectors,
- $\mathcal{F}$ set of flights,

- $G^f = (V^f, E^f)$ feasible network for flight f,
- $\mathcal{L}_e^f$ set of outgoing arcs from the head node of arc $e \in E^f$,
- $\mathcal{P}_e^f$ set of arcs entering the tail node of arc $e \in E^f$,
- $\mathcal{K}_e^f$ set of feasible flight levels for flight f traversing arc e,
- δ_f^+ (δ_f^-) maximum flight level increment (reduction) for flight f,
- α_e^f flight time to travel arc e by flight f,
- $\underline{t}_f$ scheduled departure time of flight f,
- $T_e^f = [\underline{T}_e^f, \bar{T}_e^f]$ set of feasible time periods to fly arc e for flight f,
- $C_{e,l}^f$ cost of using arc e at level l for flight f,
- R_s route charge for crossing en-route sector s.

To formulate the problem, the following decision variables are defined:

$$x_{e,l}^f(t) = \begin{cases} 1 & \text{if flight } f \text{ traverses arc } e \text{ at flight level } l \text{ by time } t \\ 0 & \text{otherwise} \end{cases}$$

The flight level of arcs leaving (*resp.* entering) an airport of origin (*resp.* destination) refers to the top of climb (*resp.* top of descent). In what follows, we denote these arcs for subgraph G^f as $orig_f \in E^f$ (*resp.* $dest_f \in E^f$).

As mentioned above, we adopt a multi-objective approach as described in [7]. In particular we consider the same objective functions as they were identied by consultation with relevant ATM stakeholders. For the sake of completeness, we here describe the three objective functions.

1. **The total ground delays** $D(\mathbf{x})$.

$$D(\mathbf{x}) = \sum_{f \in \mathcal{F}, t \in \mathcal{T}, l \in \mathcal{K}_{orig_f}^f} (t - \underline{t}_f)(x_{orig,l}^f(t) - x_{orig_f,l}^f(t-1))$$

Recall that both airborne holding delay aircraft speed adjustments are not a control option of the model. Therefore, for the purpose of this paper we focus on ground delay only.

2. **The deviation from the preferred routes** $C(\mathbf{x})$. An airspace user suggests the preferred routes according to criteria of efficiency determined by the user itself. Flight efficiency is described by EUROCONTROL in terms of subjective cost criteria such as flight time costs, fuel costs and costs of delays due to air traffic flow and capacity management actions. However, these internal costs are not available - AUs are reluctant to share this information -, therefore we compute the efficiency of a flight in terms of distance from the preferred route, according to costs $C^f_{e,l}$, which are generated to reflect some efficiency factors such as fuel consumption.

$$C(\mathbf{x}) = \sum_{f\in\mathcal{F}, e\in E^f, l\in\mathcal{K}^f_e} C^f_{e,l} x^f_{e,l}(\bar{T}^f_e).$$

Observe that according to the variables' definition, $x^f_{e,l}(\bar{T}^f_e) = 1$ means that flight f has traversed arc e at level l in some time periods.

3. **The route charges** $R(\mathbf{x})$. Each flight has to pay charges when flying the airspace of any European State. Route charges of a flight are proportional to the great circle distance between the entry and the exit point of the charging zone. In our implementation, we have approximated the route charges, by summing up charges for flying an en-route sector. That is, each time a flight $f \in \mathcal{F}$ enters a sector, the corresponding route charge is added to the objective function. The expression used to compute route charge is the following:

$$R(\mathbf{x}) = \sum_{f\in\mathcal{F}} \sum_{s\in\mathcal{S}, e\in E^f\cap\mathcal{I}_s, l\in\mathcal{K}^f_e} R_s x^f_{e,l}(\bar{T}^f_e)$$

where $\mathcal{I}_s$ is the set of arcs entering sector s. The proposed approximation may double count route charges if a flight enters the sector more than once. This event may occur if en-route sectors are non-convex. However, we did not observe such abnormal behavior in our computational experience.

The resulting multi-objective binary programming model can be thus summarized:

$$\min \begin{cases} \text{total ground delays, } D(\mathbf{x}), \\ \text{deviation from preferred routes, } C(\mathbf{x}), \\ \text{total route charges, } R(\mathbf{x}). \end{cases}$$

The model's constraints set is as follows:

$$x^f_{e,l}(t) \leq \sum_{e' \in \mathcal{L}^f_e, l-\delta^- \leq l' \leq l+\delta^+} x^f_{e',l'}(t + \alpha^f_e) \qquad \forall f \in \mathcal{F},\ e \in E^f,\ l \in \mathcal{K}^f,\ t \in T^f_e : e \neq dest_f. \quad (1)$$

$$\sum_{l \in \mathcal{K}^f_e} x^f_{e,l}(\bar{T}^f_e) \leq \sum_{e' \in \mathcal{P}^f_e, l \in \mathcal{K}^f_{e'}} x^f_{e',l}(\bar{T}^f_{e'}) \qquad \forall f \in \mathcal{F},\ e \in E^f\ : e \neq orig_f. \quad (2)$$

$$x^f_{e,l}(t-1) - x^f_{e,l}(t) \leq 0 \qquad \forall f \in \mathcal{F},\ e \in E^f,\ l \in \mathcal{K}^f_e,\ t \in T^f_e. \quad (3)$$

Constraints (1) - (3) impose that each flight travels one single feasible trajectory. In addition to constraints (1) - (3), the vector of decision variables $\mathbf{x}$ belongs to the polyhedron $\mathbf{X}$ defined by all the capacity constraints, both at airports and in en-route sectors, as in the model presented in [7]. The polyhedron $\mathbf{X}$ also includes constraints on the maximum feasible route extension. These constraints may also be used in a pre-processing phase to reduce the set of feasible arcs E^f for each flight f, thus reducing the number of variables of the formulation.

Both Fleet Delay Reordering (FDR) and Margins priority schemes are incorporated in the formulation by fixing the values of certain input parameters of the problem and/or decision variables. For instance, in the FDR mechanism, the decision variables of flights with a specified trajectory are fixed. Moreover, delays constraints are included for flights that are ranked by the AU according to their relative importance. The scope of these constraints is to align the assigned delays with the ranking of the flight. The Margins mechanism is directly incorporated in the input data by shrinking the allowed departing time windows for the prioritized flights, according to the *time not before* and *time not after* rules.

3. A simulated annealing algorithm

The mathematical model presented in Section 2.1 is a large scale model and its dimensions rise dramatically with the increase of the instance size, thus making the solution of the multi-objective model impracticable by means of exact methods. To overcome this issue, we here present a multi-objective simulated annealing algorithm to solve the ATFM problem and thus computing a good approximation of the Pareto frontier in a reasonable amount of time. The motivations for implementing a simulated annealing algorithm are: *i*) its effectiveness in finding good solutions in a short amount of time; *ii*) the robustness, meaning that its performance is not negatively influenced by peculiarities of problem instances, as observed for example in [14]; *iii*) its applicability to a number of optimization problems, see, e.g., [1]; *iv*) its successful implementation for problems with similar features, e.g., the multi-criteria path problems [15].

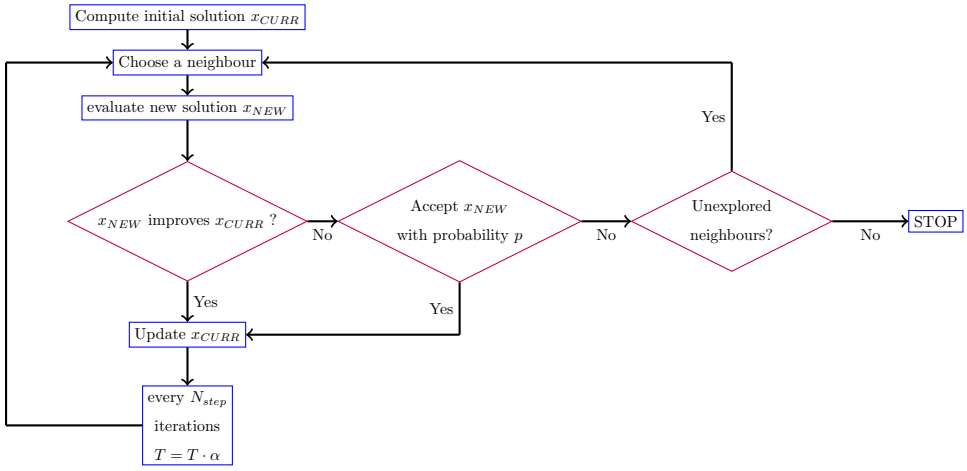


Figure 1: Flow chart of the simulated annealing algorithm

The workflow of the simulated annealing algorithm is shown in Figure 1. The core routine of the algorithm is the problem specific local search; combined with a mechanism that allows "uphill moves" to solutions of higher cost according to the so-called Metropolis criterion, to prevent the algorithm from being trapped in a poor local optimum. At each iteration of the local search that does not provide a better solution, the probability p of accepting the new solution (x_{NEW}) over the current solution (x_{CURR}) is given by $p = \exp(-1/\tau \cdot \Delta(x_{NEW}, x_{CURR}))$, where $\Delta(x_{NEW}, x_{CURR})$ is a weighted sum of the difference in the three objective functions between x_{NEW} and x_{CURR}. Because the objective function variations are comparable for all the three objectives, we used uniform weights to compute $\Delta(x_{NEW}, x_{CURR})$. The parameter τ is initially set so that the initial probability of accepting the move is quite high, usually around 0.9. In practice, the value of τ is set the first time in which we need to choose whether the non improving solution should or should not be accepted, by computing $\tau = -\Delta(x_{NEW}, x_{CURR})/\log_e(0.9)$. The probability then decreases progressively, by being multiplied by a factor $\alpha < 1$ every N_{step} iterations, so that the probability of accepting worsening moves decreases too and the convergence of the method is guaranteed. When the worsening move is not accepted, the algorithm looks for a not yet explored neighbour of the current solution, eventually stopping when no such neighbour is found.

Before describing the local search in detail, we introduce the concept of *dominance* in multi-objective optimization, which is used to compare two solutions, say x_A and x_B. In particular, x_A is said to *dominate* x_B ($x_A \succeq x_B$), if the value of each objective function of x_A is not worse than the corresponding value of x_B; and if x_A is strictly better than x_B in at least one objective

function. x_A *is non-dominated* with respect to x_B ($x_A \nleq x_B$), if x_A outperforms x_B for some of the objectives but it is outperformed in the remaining ones [22].

In the following subsections, we present the main components of the proposed local search, i.e., *i*) the definition of the neighbourhood and its exploration; *ii*) the acceptance criterion; and *iii*) the computation of the initial solution.

3.1. Definition of the neighbourhood and its exploration

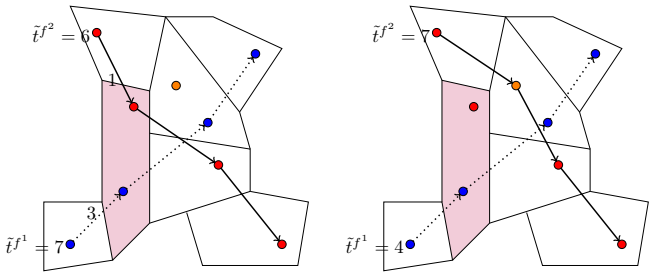
The definition of the neighbourhood of a solution is a crucial element for the development of a local search algorithm. We here present two *neighbourhood structures*, both obtained by re-routing one flight as the "primitive" move. To choose the flight to be rerouted at any iteration, we tested two different criteria, i.e., the "most delayed flight" and the "random". However, after a preliminary analysis, we opted for the random selection criterion as it provides some more "diversification" in the algorithm. For similar reasons, we also randomly choose, whenever needed, an arc to be avoided which belongs to the route of this flight.

In the following example, we show the potential benefits of rerouting flights.

Example 1. *Let us consider two flights f^1 (dashed line) and f^2 (solid line). Suppose that f^1 is planned to take off at $t^{f^1} = 4$ and f^2 at $t^{f^2} = 6$. This would result in both flights entering the shaded sector (see Figure 2), that we suppose can held no more than one flight per time period, at time period $t = 7$. Because of the (shaded) sector capacity constraints, and supposing that the sector is already congested at time periods $t = 8, 9$, flight f^1 needs to be delayed and take off at time period $t = 7$, as shown in Figure 2(a). However, if we reroute f^2 as depicted in Figure 2(b), flight f^1 is able to take off on time. We can see that the overall ground delay is decreased, yielding an improvement on the current solution, even if the reroute forces f^2 to depart with one time period of delay to avoid congestion in the following sector.*

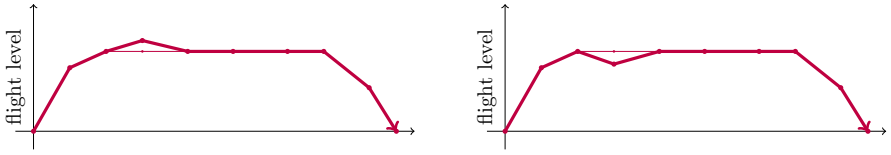
3.1.1. Rerouting according to Stakeholders' preferences

The first neighborhood structure described is the one that mimics the stakeholders' preferences. Although a common agreement on the ranking of alternatives (preference) has not been found yet, a reasonable ranking of options for managing delays is 1) flight level change, 2) horizontal deviation, and 3) ground delays, with the third one the least preferable. Therefore, given a solution $\mathbf{x}$ of the ATFM problem, the neighborhood $\mathcal{N}_p(\mathbf{x})$ is defined as the set of all solutions obtained from $\mathbf{x}$ by rerouting one flight with one of the following moves (which are applied sequentially) :



(a) before rerouting (b) after rerouting

Figure 2: Example of rerouting benefits



(a) increase flight level (b) decrease flight level

Figure 3: Flight level reroute

CHANGE IN THE FLIGHT LEVEL: we choose an arc e entering a sector with no spare capacity left, then we increase or lower the flight level of that arc by one level, ensuring that the new flight level is feasible and that the difference between flight levels in adjacent arcs does not exceed a fixed maximal variation δ_f. Figure 3 illustrates this kind of rerouting.

CHANGE THE 2D ROUTE: Because the amount of rerouting - meaning deviation from the preferred route - accepted by airlines is rather modest and in our analysis is limited to 80 NM, we first restrict the rerouting of flights to waypoints adjacent to the ones currently traversed by the flight. The following two distinct rerouting moves are considered:

- 1-waypoint reroute: we select an arc $e = (i, j)$ between the waypoints i and j that flight f needs to avoid. We then look for another waypoint y such that there exist two arcs (i', y) and (y, j'), where i' and j' belong to the current path and i' does not follow i, j' does not precede j. Examples of rerouted paths are shown in Figure 4.

- 2-waypoints reroute: this second rerouting routine applies a major disruption to the current horizontal trajectory, as shown in Figure 5. Given an arc $e = (i, j)$ that cannot be traversed, we look for an arc $\bar{e} = (y, h)$ such that there exists two arcs (i', y) and (h, j'), where i' and j' are defined as in the previous routine.

(a) (b)

Figure 4: 1-waypoint reroute

(a) (b)

Figure 5: 2-waypoints reroute

Observation 3.1. *If the graph representing the airways system is maximal planar, then the 2-waypoint move can be obtained by applying the 1-waypoint move twice. Indeed, under the assumption of maximal planarity of the graph, any route between an origin-destination pair of airports can be obtained by applying a sequence of the flight level change and 1-waypoint moves.*

The value of the objective functions of the new solution are computed in the following way: the deviation from the preferred route and route charges are updated by subtracting the contribution of the modified flight and by adding the contribution of the new trajectory. Because the modification of a trajectory may violate some capacity constraints, delays are assigned by solving a single objective binary program, precisely a scheduling problem (SP) described in Section 3.3, and thus restoring feasibility. The full routine is summarized in Algorithm 1. It takes as input the current solution x_{CURR} and ends as soon as a new neighbour is selected, or when no suitable neighbour is found.

3.1.2. Neighbourhood $\mathcal{N}_s$ for sparse networks.

The main issue with the previous exploration of the neighbourhood is that it is necessary to consider a fairly well-connected network. This is not always possible in real applications, due to the following reasons: first, the storage of a maximal planar graph for real-size instances can lead to massive memory requirements; second, networks based on historical data are generally sparse. This means that, for all practical purposes, it is unlikely that a full representation of the network between waypoints is used. In this event, the neighbourhood definition given in the previous

Algorithm 1 Local search with neighbourhood structure $\mathcal{N}_p$

procedure FollowPreferences
    given the current solution x_{CURR}
    randomly choose flight f, arc $e \in E^f$ to be avoided
    apply flight level reroute and compute the new solution x_{NEW}
    if $x_{NEW} \succeq x_{CURR}$ **or** $x_{NEW} \not\preceq x_{CURR}$ **then**
        $x_{CURR} = x_{NEW}$
        Stop.
    else
        apply 1-waypoint reroute and compute x_{NEW}
        if $x_{NEW} \succeq x_{CURR}$ **or** $x_{NEW} \not\preceq x_{CURR}$ **then**
            $x_{CURR} = x_{NEW}$
            Stop.
        else
            apply 2-waypoint reroute and compute x_{NEW}
            if $x_{NEW} \succeq x_{CURR}$ **or** $x_{NEW} \not\preceq x_{CURR}$ **then**
                $x_{CURR} = x_{NEW}$
                Stop.
            end if
        end if
    end if
end procedure

subsection may be highly ineffective, as it can even happen that no suitable neighbour is found. Hence, we present an alternative neighbourhood structure ($\mathcal{N}_s$) that is not negatively affected by the sparsity of the network representing the airspace system.

Definition 3.1. *Given the current solution, a neighbour is obtained by forcing one flight to avoid a particular arc and computing the resulting shortest path between its origin and destination. All the flights are then rescheduled accordingly.*

Similarly to the previous reroutes, we randomly select a flight f and an arc e that should be avoided. We set the cost $C^f_{e,l}$ of traveling that arc at the current flight level to $+\infty$ and we compute the shortest path from the airport of departure to the airport of destination. Notice that the preferences of stakeholders are still taken into account, as a change in the flight level will be taken into account as long as a more efficient horizontal reroute is not found. The objectives that express the efficiency of flights and the route charges are updated by modifying the contribution given by flight f, while the amount of delays is determined solving optimally the related SP problem. This routine is summarized in Algorithm 2.

Algorithm 2 Local search with neighbourhood $\mathcal{N}_s$

procedure JUMPSHORTESTPATH
  given the current solution x_{CURR}
  randomly choose flight f, arc $e \in E^f$ to be avoided
  let l be the flight level occupied by fight f along arc e
  set $C^f_{e,l} = +\infty$
  compute the shortest path between d_f and a_f with weights $C^f_{e,l}$ on the arcs
  compute x_{NEW}
  if $x_{NEW} \succeq x_{CURR}$ **or** $x_{NEW} \not\preceq x_{CURR}$ **then**
    $x_{CURR} = x_{NEW}$
    Stop.
  end if
end procedure

3.2. Acceptance criterion

When a neighbour solution is evaluated, the algorithm has to decide if it improves the current solution and hence if it should be accepted as the new current solution. Because of the multi-objective context, when comparing a new solution x_A with the current solution x_B, we can have three possible outcomes:

1. x_A *dominates* x_B ($x_A \succeq x_B$);
2. x_A *is dominated* by x_B ($x_A \preceq x_B$);
3. x_A *is non-dominated* with respect to x_B ($x_A \not\preceq x_B$).

As pointed out in [22], case 3 is specific of multi-objective formulations and can be treated in different ways: we can either always consider non-dominated solutions as improving ones, or we can always discard them, or set a probability according to which we accept them as improving solutions. In this paper we always consider non-dominated solutions as improvements and hence we accept them.

3.3. Computation of an initial solution

To compute an initial feasible solution, we use the following mathematical programming approach. We assume that each aircraft flies its preferred 3D route. This implies that one of the objectives, i.e., the deviation from the preferred routes $C(\mathbf{x})$, is minimum. In this specific application, fixing the 3D routes also unambiguously determine the value of the route charges (third objective of the model presented in Section 2.1). Hence, the following observation holds:

Observation 3.2. *If the* 3D *routes are fixed, the multi-objective ATFM problem reduces to find a feasible schedule for all the flights that can be computed by solving a single-objective binary program of the type:*

min	total delays
s. t.	capacities are not exceeded

This model is much more compact than the "full" model presented in §2.1, as it has a single objective and we only need to decide the estimated departure time for each flight. We here introduce some additional notation to be used in the formulation of this scheduling problem (SP).

- $\mathcal{S}$ set of sectors,
- $\mathcal{A}$ set of airports,
- d_f departure airport of flight f,
- a_f arrival airport of flight f,
- A_k^t arrival capacity of airport $k \in \mathcal{A}$ at time t,
- D_k^t departure capacity of airport $k \in \mathcal{A}$ at time t,

– Γ_s^t capacity of sector $s \in \mathcal{S}$ at time t,

– $\Gamma_{s,l}^t$ capacity of sector s at flight level l and time t,

– $\mathcal{I}_s$ set of arcs entering sector s,

– $T^f = [\underline{t}^f, \bar{T}^f]$ the set of feasible time periods for flight f to take off. Recall that $\underline{t}_f$ is the scheduled departure time of flight f.

Moreover, we introduce the quantity

$$\theta_e^f = \sum_{\epsilon \lessdot e} \alpha_\epsilon^f,$$

which is equal to the time needed to reach arc e along the path used by flight f. Notice that the symbol $\lessdot$ is used to denote the arcs that preceed arc e along the path. For ease of notation, $\bar{\theta}^f$ denotes the amount of time needed to fly the entire route of flight f.

We formulate the problem using the following binary variables

$$x^f(t) = \begin{cases} 1 & \text{if flight } f \text{ takes off at time } t \in T^f, \\ 0 & \text{otherwise.} \end{cases}$$

The scheduling problem (SP) is formulated as follows:

$$(SP) \quad \min \sum_{f \in \mathcal{F}, t \in T^f} (t - \underline{t}_f) x^f(t)$$

$$\text{s. t.} \sum_{t \in T^f} x^f(t) = 1 \qquad \forall f \in \mathcal{F} \qquad (4)$$

$$\sum_{f \in \mathcal{F}: d_f = k} x^f(t) \leq D_k^t \qquad \forall\, k \in \mathcal{A},\, \forall\, t \in \mathcal{T} \qquad (5)$$

$$\sum_{f \in \mathcal{F}: a_f = k} x^f(t - \bar{\theta}^f) \leq A_k^t \qquad \forall\, k \in \mathcal{A},\, \forall\, t \in \mathcal{T} \qquad (6)$$

$$\sum_{f \in \mathcal{F}, e \in E^f \cap \mathcal{I}_s} \delta_e^f \cdot x^f(t - \theta_e^f) \leq \Gamma_s^t \qquad \forall\, s \in \mathcal{S},\, \forall\, t \in \mathcal{T} \qquad (7)$$

$$\sum_{f \in \mathcal{F}, e \in E^f \cap \mathcal{I}_s} \delta_{e,l}^f \cdot x^f(t - \theta_e^f) \leq \Gamma_{s,l}^t \qquad \forall\, s \in \mathcal{S},\, \forall\, t \in \mathcal{T},\, \forall\, l \in \mathcal{L} \qquad (8)$$

$$x^f(t) \in \{0, 1\} \qquad \forall\, f \in \mathcal{F},\, t \in T^f$$

where parameter $\delta_{e,l}^f$ is equal to one if flight f crosses arc e at flight level l, and zero otherwise; and parameter δ_e^f is the aggregation over the feasible set of flight levels ($\delta_e^f = \sum_{l \in \mathcal{K}^f} \delta_{e,l}^f$).

Constraints (4) ensure that each flight is scheduled, i.e. it needs to take off by the end of its feasible time window. Constraints (5) - (6) are the departures and arrivals capacity constraints respectively, while constraints (7) and (8) impose en-route sector capacity constraints. These constraints take into account the total workload of air traffic controllers. The scope of constraints (7) is to have a more accurate measure of the capacity. Finally, observe that this procedure to compute an initial solution of the simulated annealing is valid as long as the SP problem is feasible, which is guaranteed if the set of feasible time periods (T^f) is sufficiently large for each flight $f \in \mathcal{F}$.

4. Computational results

The proposed heuristic was implemented using C++ language. To solve the SP problem, we used SCIP libraries combined with CPLEX. We tested the performance of the algorithm on a case study involving four major European airports, namely, London Heathrow, Paris Charles De Gaulle, Amsterdam Schiphol and Frankfurt. Data were gathered from the Eurocontrol Demand Data Repository (DDR2), as described in [3]. We selected October 3^{rd}, 2016 because it was a busy day of the year. 186 flights, operated by six airlines, flew between origin-destination pairs of the four selected airports during the whole day, and the routes traversed in total 694 relevant waypoints. We simplified the underlying network and we built an acyclic, connected network between each pair of the four selected airports. For each flight, we estimated the cost ($C^f_{e,l}$) for traversing an arc e of the acyclic graph taking into account both the time needed to fly the arc and the flight level l. We also assumed that the preferred route corresponds to the shortest path between the origin and destination because relevant information was not available. We limited the amount of rerouting (route extension) to 80 NM for all airlines, as suggested during the OptiFrame workshop [17]. Information on capacity was also retrieved from the DDR2 database. For each sector, the capacity profile over time was the residual capacity computed by subtracting the capacity used by all the other flights with different origin and/or destination. The values of the flight level capacities was computed by dividing the total sector capacity by the number of flight levels and by rounding up the results. The time horizon was discretized into 10-minute periods, yielding a total of 144 time periods. Finally, for each flight f and arc e, the set of feasible flight levels $\mathcal{K}^f_e$ included: *i*) the preferred flight level; *ii*) one level above the preferred flight level; and *iii*) two flight levels below. From now on, we refer to this instance as instance I1.To account for the effect of an air traffic growth, as forecasted by EUROCONTROL in 2035, we also generated a

second instance with 300 flights, labeled I2. The remaining characteristics of Instance I2 are the same as in the case of Instance I1. 114 additional flights were randomly generated and assigned to origin-destination pairs and airlines. This instance would also provide the opportunity to test the scalability of the proposed algorithm.

To examine the effects of different air traffic system disruptions, for both instances we also analyzed the following scenarios in the interest of practitioners [17]:

1. Airport Closure (AC): one of the four airports is closed for one hour, i.e. for six time periods, both for departures and arrivals,
2. Airport Restriction (AR): one airport for one hour has reduced capacity for both take offs and landings,
3. Sector Restriction (SR): a sector has reduced capacity throughout the day.

The approximated Pareto frontier was generated by combining the non-dominated solutions obtained from 10 runs of the algorithm (obtained with different seeds of the random number generator). The time limit for each run was set to 3 hours.

To compare the computational performances of the two neighbourhood structures and to assess the quality of the solutions with reference to the Pareto frontier, we use metrics indicators specifically designed for multi-objective optimization [23, 9, 13, 12]. The scope of these indicators is to assess two major requirements of a heuristic algorithm: convergence and uniform diversity. The convergence property ensures that the approximated solutions are not too far from the actual Pareto frontier, while the uniform diversity ensures that a wide portion of the efficient frontier is explored. In this section, we use the *Generational Distance* and the *Generalized Spread* indicators to measure the convergence and the diversity respectively. A brief description of these indicators is here given:

1. Generational Distance (GD) [9]: it measures how far the elements of the approximation set are from the Pareto frontier. Let N be the number of non-dominated solutions in the approximation set and $d_i, i \in \{1, \ldots, N\}$ be the distance between each approximated solution and the nearest point in the Pareto frontier. The GD indicator is computed as:

$$GD = \frac{\sum_{i=1}^{N} d_i^2}{N}$$

The smaller this indicator is, the better the approximation of the Pareto front we get. In particular, if this indicator is equal to zero then every point of the approximation set is also Pareto optimal.

Instance	$\mathcal{N}_p$				$\mathcal{N}_s$			
	# non-dom		time (s)		# non-dom		time (s)	
	av.	σ	av.	σ	av.	σ	av.	σ
I1	10.60	1.85	78.85	32.28	121.90	23.47	585.02	14.47
I1 AC	8.70	2.57	70.45	23.04	167.50	27.95	696.35	79.42
I1 AR	9.40	2.65	77.92	31.11	124.50	21.76	637.69	110.86
I1 SR	10.00	2.57	91.44	32.79	144.10	18.66	702.03	59.09

Table 1: Average performance of the local search routine

2. Generalized Spread (Δ) [9]: it generalizes the spread indicator. This indicator is based on the computation of the distance between two consecutive solutions. When dealing with more than two objectives, however, the concept of consecutive solutions is not easily defined, thus the need of a more general indicator. Let S be the approximation set and $S^\star$ the Pareto frontier, $d(X, S)$ the distance between the solution X and the nearest point belonging to S, $e_1, \ldots, e_m$ the extreme solutions, given m the number of objective functions, and let

$$\bar{d} = \frac{1}{|S|} \sum_{X \in S} d(X, S), \qquad \bar{d^\star} = \frac{1}{|S^\star|} \sum_{X \in S^\star} d(X, S).$$

The Δ indicator is given by:

$$\Delta = \frac{\sum_{i=1}^{m} d(e_i, S) + \sum_{X \in S} |d(X, S) - \bar{d}|}{\sum_{i=1}^{m} d(e_i, S) + |S| \cdot \bar{d^\star}}$$

Higher values of this indicator show that the heuristic spans a wider area of the feasible space.

4.1. Comparison of the two neighbourhood structures

In this section we compare the two neighbourhood structures presented in Section 3, namely $\mathcal{N}_s$ and $\mathcal{N}_p$. For the four instances with the current air traffic demand (group I1) and for each neighbourhood structure, Table 1 displays the average value (*av.*) and standard deviation (σ) obtained over 10 runs of the algorithm for both the number of non-dominated solutions and the computational time in seconds.

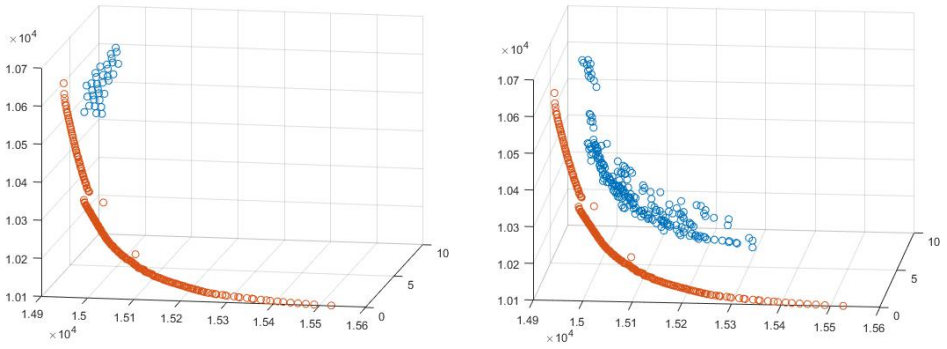
The statistics reported in the Table 1, clearly demonstrates that the neighbourhood structure $\mathcal{N}_p$, the one mimicking the stakeholders' preferences, only explores a limited portion of the feasible space and is able to find only a small set of non-dominated solutions. On the other

Instance	(Pref)		(ShortP)	
	GD	Δ	GD	Δ
I1	3.68	0.03	3.30	0.12
I1 AC	5.22	0.05	4.29	0.08
I1 AR	3.97	0.04	3.16	0.11
I1 SR	4.88	0.04	2.34	0.12

Table 2: Performance indicators on different neighbourhhoods

hand, the neighbourhood structure $\mathcal{N}_s$ is able to better explore the feasible region and computes a significantly larger set of non-dominated solutions, which obviously comes at the cost of larger computational times. However, such computational times are still tractable for the considered application domain. It is also important to observe that given the large number of non dominated solutions computed at each run of the algorithm, a smaller number of runs might be sufficient to have a good representation of the Pareto frontier.

The different behavior of the two neighbourhood explorations is also highlighted in Figure 6. In order to produce these charts, we collect the overall non-dominated solutions achieved during the ten runs of each instance to form the approximated Pareto frontier, i.e., among all the non-dominated solutions obtained from the ten runs, we only keep those that are not dominated by any other solution. In Figure 6 we show the approximated frontiers for the $\mathcal{N}_p$ neighbourhood, on the left, and the $\mathcal{N}_s$ neighbourhood, on the right. Both the approximated Pareto frontiers are compared with the exact one. The blue dots in each graph represent the non-dominated approximated solutions in a 3D space in which each axis represents one of the objective functions, while the red dots represent the Pareto frontier computed with the Quadrant Shrinking Method [6] (further described in Section 4.2). It is clear that, due to the network structure of the instances considered (i.e. graphs are too sparse), the local search that mimics the stakeholders' preferences is trapped in the neighbourhood of the initial solution, thus leaving unexplored a large portion of the feasible space. This is quantitatively expressed by the *Generational Distance* and the *Generalized Spread* indicators introduced above, whose values are reported in Table 2. Indeed, the $\mathcal{N}_s$ neighbourhood yields a considerably larger value for the spread 0.12 *vs* 0.03 for the $\mathcal{N}_p$ neighbourhood. $\mathcal{N}_s$ neighbourhood performs better also in terms of the *Generational Distance* indicator, meaning that the associated approximation set is closer to the actual Pareto frontier (3.68 *vs* 3.60).



(a) $\mathcal{N}_p$ (b) $\mathcal{N}_s$

Figure 6: Non-dominated solutions for neighbourhoods $\mathcal{N}_p$ and $\mathcal{N}_s$.

In view of these results, in the remaining of this section we use the local search with the $\mathcal{N}_s$ neighbourhood within the simulated annealing algorithm.

4.2. Performance of the algorithm.

In this section, we compare the set of efficient solutions computed by the simulated annealing algorithm with the Pareto frontier obtained by the Quadrant Shrinking Method (QSM) [6]. This exact solution method, specifically developed to solve tri-objective integer programs, iteratively computes all the Pareto optimal solutions. At each iteration, the method solves two integer problems: the first problem computes a weekly efficient solution by minimizing only one objective (e.g., the third one) while imposing an upper bound on the other two objectives; the second problem minimizes the sum of the three objectives while bounding the each objective to the objective value of the solution computed in the first integer program. At each iteration the boundaries of the feasible region are shrinked. A detailed description of how this method has been adapted to the ATFM problem herein studied can be found in [7].

In addition to showing the number of non-dominated solutions (*# non-dom*) and the computational time in seconds (*time*) for both the exact method and the simulated annealing algorithm, Table 3 provides the hypervolume (HV) ratio between the value of the heuristic set of efficient solutions (HV-H) and the value of the exact Pareto frontier (HV-E). The hypervolume is a metric to assess the quality of multi-objective optimization algorithm. For a problem with n objectives, the hypervolume gives a measure of the n-dimensional space enclosed between all the solutions of the Pareto frontier and a reference point identified by the worst possible value for each objective

Instance	QSM		Simulated annealing		$\frac{HV-H}{HV-E}$
	# non-dom	time (s)	# non-dom	time (s)	(%)
I1	232	20722	257	5875	70
I1 AC	233	18333	317	6221	51
I1 AR	232	18910	220	7053	68
I1 SR	232	19039	228	6818	64
I2	185	43236	216	10792	67
I2 AC	183	51303	293	11575	53
I2 AR	184	54240	311	7964	52
I2 SR	185	48788	279	12699	66

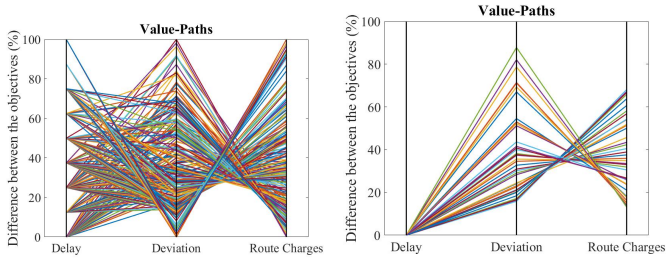
Table 3: Comparison between QSM and simulated annealing algorithms.

function. The ratio between the hypervolume for the heuristic and the exact methods gives a measure of the proximity between the two frontiers. The higher is the ratio, the better is the approximation. This metric has the property of capturing both the generational distance and the spread in the objective space of the approximation set of the Pareto front. Moreover, it is also the only known indicator that reflects Pareto dominance, i.e., if an approximation set weakly dominates another this will be reflected in the values of HV. To compute the hypervolume, we have here used an algorithm based on Lebesgue Measure Algorithm, described in [10].

The simulated annealing heuristic shows significantly smaller computational times and better scalability properties, without sacrificing the quality of the solutions computed. Indeed, the simulated annealing heuristic is able to produce non-dominated solutions that are close to the Pareto frontier and span a large portion of it, as demonstrated by the values of the *GD* and Δ indicators shown in Table 2; and the HV ratio displayed in Table 3. Although the HV ratio reduces for some of the disruption scenarios, the average value is 61%, which is satisfactory for the application under consideration.

5. Solutions' analysis to support decisions.

To represent the trade-off between the three objectives of the ATFM problem, Figures 7(a) and 8(a) show the *value paths* of the non-dominated solutions for instances I1 and I2 respectively. Each solution is represented by a colored piecewise linear curve (value path) which connects the three values along the vertical lines, corresponding to the three objective considered. The values reported on the vertical axes represent how much each solution differs from the optimal solution of the corresponding objective. The deviation values are computed as the percentage difference



(a) Nominal Scenario (b) Minimum Delay

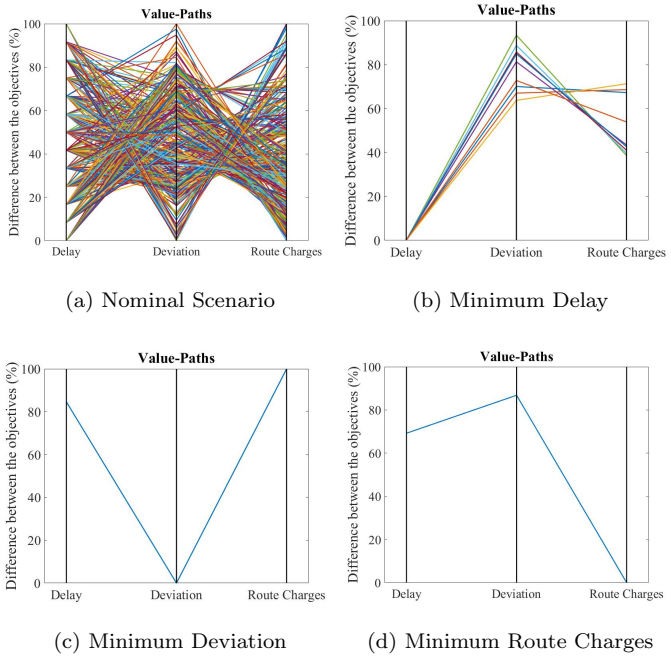
(c) Minimum Deviation (d) Minimum Route Charges

Figure 7: Value paths of non-dominated solutions, instance I1

between each solution and the optimal solution of the corresponding objective (Gap). In formula, the Gap for each of the three objectives $f_1, \ldots, f_3$ and for each solution i is computed as follows:

$$\text{Gap}_n(i) = \frac{f_n(i) - \min_i\{f_n(i)\}}{\max_i\{f_n(i)\} - \min_i\{f_n(i)\}} \cdot 100$$

Figures 7(b), (c), (d) and 8(b), (c), (d) highlight the solutions that reached minimum values for at least one of the objective functions. Notice that there are multiple non-dominated solutions that yield the minimum amount of delay, i.e., there are different ways of assigning flights trajectories that lead to the same amount of total delay (Figure 7(b), 8(b)). This does not happen when minimizing the deviation from the preferred routes or the route charges. In these cases, there is a unique solution as shown see figures (c) and (d) of Figure 7 and Figure 8. However, for the route charges objective this is not in general the case. From both instances, it is evident that the minimization of one of the objectives leads to a significant deterioration of the other two objectives. Indeed, the values of the gap for the other objectives are considerably higher, see for instance solutions with minimum deviation or route charges. This shows that there is actually a trade-off between the objectives of the model, thus justifying the choice of a three-objective optimization approach.



(a) Nominal Scenario (b) Minimum Delay

(c) Minimum Deviation (d) Minimum Route Charges

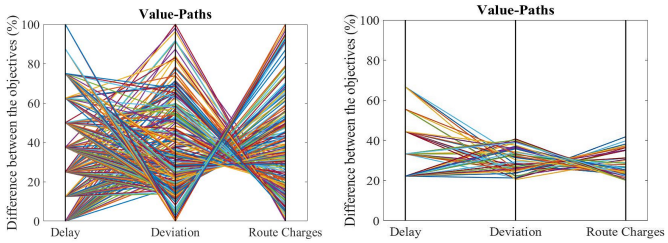
Figure 8: Value paths of non-dominated solutions, instance I2

5.1. Solutions' filtering

The scope of the multi-objective approach is to enable the stakeholders to make more informed decisions by providing information about the trade-offs involved among the three objective function values. However, as shown in the illustrated examples, the multi-objective approach may produce a large number of non-dominated solutions, e.g., 250 non-dominated solutions for instance I1. It is important to provide the stakeholders with tools that enable them to discern the "best" solution out of all the non-dominated solution. Therefore, it is fundamental to provide criteria to filter out solutions of the Pareto frontiers. The criteria can be stakeholder specific. For instance, consider the following two criteria:

- limiting the (feasible) values of a specific objective within a certain range;
- limiting all the objectives in specified intervals of the percentage gap.

Any of these criteria or any their combination leads to a reduction of non-dominated solutions to be considered and can facilitate the decision making process. As an example, we may disregard all the solutions that for any of the three objective fall either in the worst 20% or in the best 20%. The rationale for discarding the best solutions for a given objective lies in the trade-off embedded in



(a) I1 - All (b) I1 - Filtered

Figure 9: Solutions with percantage gap between the 20% and 80% for each objective function.

any solution. Stakeholders accept suboptimality (small deterioration) of one objective in exchange of better performances of the other objectives. Applying these criteria to the approximated Pareto frontier of instance I1, we reduce the number of solutions from 257 to 48, as shown in Figure 9.

The decision regarding the selection of a solution of the Pareto frontier to be implemented is a very central and crucial issue regarding the implementation of the TBO concept. The criticality and importance of this decision stems from the fact that each solution of the efficient frontier may impact the stakeholders differently. Furthermore, different stakeholders may assign different importance to the different objectives expressing their preferences. Therefore, it is important to provide the capability to drill down from aggregate values of the objective functions to single airline values. For instance, Figure 10 displays the values of the three objectives for each airline together with a chart that shows the airline's share of each objective, for three randomly selected non-dominated solutions. With this additional level of information, each airline will be able to evaluate all the efficient solutions and the corresponding trade-off.

5.2. Incorporation of priority schemes

In this section we consider the implementability and impact of incorporating two priority schemes, namely the Flight Delay Reordering (FDR) and Margins schemes. Both schemes are incorporated at a pre-processing level as discussed in Section 2. In the following subsections we present preliminary results on the effects on the overall system performance derived from the incorporation of priorities. All tests have been performed on instance I1.

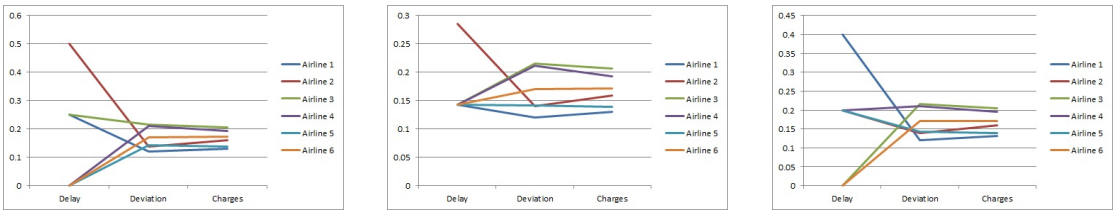
5.2.1. Application of FDR priority scheme

We applied the FDR priority scheme to an airline that incurred a delay at one of the airports in some of the efficient solutions. We suppose that, given the probability of the flight to be delayed,

	Delay	Deviation	Charges
Airline 1	1	1811	1343
Airline 2	2	2090	1650
Airline 3	1	3253	2136
Airline 4	0	3175	1992
Airline 5	0	2148	1436
Airline 6	0	2550	1783
Total	4	15027	10340

	Delay	Deviation	Charges
Airline 1	1	1813	1339
Airline 2	2	2107	1636
Airline 3	1	3251	2126
Airline 4	1	3179	1977
Airline 5	1	2141	1431
Airline 6	1	2574	1757
Total	7	15065	10266

	Delay	Deviation	Charges
Airline 1	2	1808	1344
Airline 2	1	2101	1635
Airline 3	0	3263	2107
Airline 4	1	3169	2000
Airline 5	1	2148	1437
Airline 6	0	2575	1755
Total	5	15064	10278



(a) (b) (c)

Figure 10: Analysis of results at airlines' level

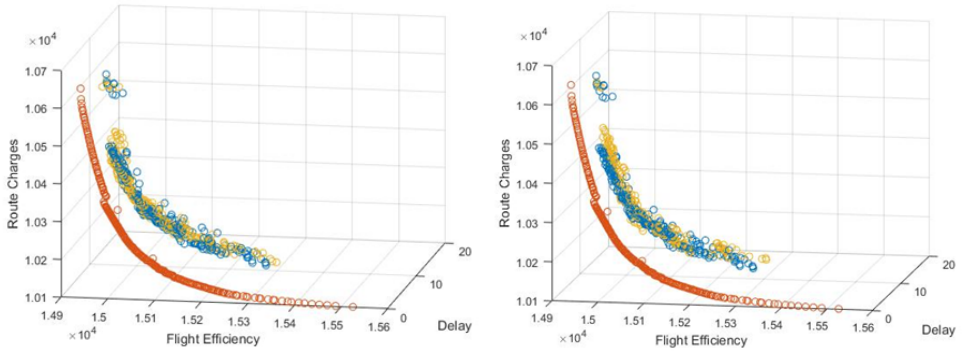
the airline decides to give it a higher priority and hence its departure time is swapped with the departure time of a subsequent flight of the same airline. Total delays are computed taking into account the original departure times.

Figure 11a shows in a 3D representation the alternative Pareto frontiers each identified with a different color. More precisely, in red is depicted the optimal Pareto Frontier of the instance without priorities; in blue the efficient solutions found by the heuristic algorithm; and in yellow the efficient solutions found by the heuristic algorithm under the priority scheme. In this case study, the efficient solutions with and without the priority scheme, though different, have comparable quality. Therefore, in this specific example, the introduction of the priority scheme does not weaken the quality of the initial efficient solutions found.

Notice that this priority scheme is unlikely to produce infeasible solutions if the original schedule allows a feasible one, because flights may be displaced in time as much as it is necessary to comply with the capacity constraints. As a drawback, if too many priorities are taken into account, it may result in an overall increase of the total delay that can hit back the flights with high priority.

5.2.2. Application of Margins scheme

We tested the Margins priority scheme by selecting a pair of flights to be prioritized for each airline and reducing the feasible time window for their departure. In this way, we force the flight



(a) I1 - FDR

(b) I1 - Margins

Figure 11: Impact of priority schemes

to follow the time-not-before and time-not-after rules set by the scheme.

In Figure 11b we show the performance of the heuristic algorithm under this priority scheme. Similarly to the previous figure, we represent in yellow colour the solutions with the margin priorities and in the without margin priorities. Also in this case, we do not observe a relevant deterioration of the quality of efficient solutions by applying the margins scheme. Indeed, the increment of the objective functions values is rather modest and can be noticed only for the portion of the Pareto frontier with higher values of deviation and route charges. This is a reasonable behaviour, as the introduction of margins forces the solution to decrease the delay for the selected flights, which are more likely to be rerouted in order to be assigned to a feasible trajectory. However, it is also important to highlight that an extensive use of margins may lead to infeasible instances; event that is more likely if the underlying network is not highly connected. In fact, to accommodate a large number of flights departing on time or with small delays, a great availability of possible reroutes is needed to ensure all capacity constraints are satisfied.

6. Conclusions

In this paper we presented a multi-objective approach for trajectory based operations in ATFM. We customized the model presented in [7] to exploit specific assumptions of the ATFM planning phase. To compute the Pareto frontier we proposed a simulated annealing heuristic that proved to be a viable approach to solve realistic instances of the problem in reasonable computational times.

The algorithm has been tested on a case study involving four major airports within Europe and has been validated in terms of scalability and response to disruptive scenarios such as airport closure, airport restriction and sector restriction, using a number of tools provided by Eurocontrol [24], i.e., NEVAC, NEST and BADA. The results herein described were also presented to key stakeholders during the OPtiFrame Final Workshop that took place at Eurocontrol HQ (Brussels) on February 14, 2018. Stakeholders provided useful feedback regarding the potential of the proposed approach to further support the development of the TBO concept; and the following directions for further research were recommended: *i)* better calibration of the model to limit the number of altitude changes; *ii)* identification of the appropriate methodology to compute flight level capacities; *iii)* refinement of the process leading to the representation of the underlying ATM network used in the mathematical model; and *iv)* development of a post-processing engine that will automate the analysis of the alternative efficient solutions generated by the algorithm and will provide Decision Support for reaching a consensus on the solution that should be implemented.

Acknowledgements

The author are very grateful to the referees for their constructive comments that improved the paper. This research is part of the OptiFrame project funded by the SESAR Joint Undertaking under grant agreement No 699275 within the EU Horizon 2020 research and innovation program. Opinions herein expressed reflect the authors' views only. SESAR and/or EUROCONTROL shall not be considered liable for them or for any use that may be made of the information herein contained.

References

[1] E. Aarts, J. Korst, and W. Michiels. Simulated annealing. In E. Burke and G. Kendall, editors, *Search Methodologies*, pages 187–210. Springer, 2005.

[2] A. Agustin, A. Alonso-Ayuso, L.F. Escudero, and C. Pizarro. On air traffic flow management with rerouting. part I: Deterministic case. *European Journal of Operational Research*, 219:156–166, 2012.

[3] G. Andreatta, L. Capanna, L. De Giovanni, and L. Righi. Routines for data extraction and for the generation of ATFM algorithm input data. Technical report, OptiFrame Consotium, 02 2017. Deliverable of the OptiFrame project.

[4] H. Balakrishnan and B.G. Chandran. Optimal large-scale air traffic flow management. 2014. Available at http://web.mit.edu/hamsa/www/publications.html.

[5] D. Bertsimas, G. Lulli, and A. Odoni. An integer optimization approach to large scale air traffic flow management. *Operations Research*, 59(1):211–227, 2011.

[6] N. Boland, H. Charkhgard, and M. Savelsbergh. The quadrant shrinking method: A simple and efficient algorithm for solving tri-objective integer programs. *European Journal of Operational Research*, 260(3):873–885, 2017.

[7] V. Dal Sasso, F. Djeumou Fomeni, G. Lulli, and K. G. Zografos. Incorporating stakeholders' priorities and preferences in 4D trajectory optimization. *Transportation Research Part B*, 117:594–609, 2018.

[8] F. Djeumou Fomeni, G. Lulli, and K. G. Zografos. An optimization model for assigning 4D-trajectories to flights under the TBO concept. In *Twelfth USA/Europe Air Traffic Management Research and Development Seminar (ATM2017)*, June 2017. Seattle, Washington, USA.

[9] J.J. Durillou and A.J. Nebro. jMetal: a Java framework for multi-objective optimization. *Advances in Engineering Software*, 42:760–771, 2011.

[10] M. Fleisher. The measure of pareto optima: Applications to multiobjective metaheuristics. Technical report, Institute for Systems Research, University of Maryland, 2002. ISR TR 2002-32.

[11] L. Guichard. UDPP concept. SESAR Project 07.2. Technical report, Eurocontrol, 2017. Available on request.

[12] M. Hamdy, A. Nguyen, and J.L.M. Hensen. A performance comparison of multi-objective optimization algorithms for solving nearly-zero-energy-building design problems. *Energy and Buildings*, 121:57–71, 2016.

[13] H. Ishibuchi, H. Masuda, Y. Tanigaki, and Y. Nojima. Modified distance calculation in generational distance and inverted generational distance. In A. Gaspar-Cunha, C. Henggeler Antunes, and C. Coello, editors, *Evolutionary Multi-Criterion Optimization. EMO 2015*, volume 9019 of *Lecture Notes in Computer Science*, pages 110–125. Springer, Cham, 2015.

[14] F. Knust and L. Xie. Simulated annealing approach to nurse rostering benchmark and real-world instances. *Annals of Operations Research*, pages 1–30, 2017.

[15] L. Liu, H. Mu, H. Luo, and X. Li. A simulated annealing for multi-criteria network path problems. *Computers and Operations Research*, 39:3119–3135, 12 2012.

[16] G. Lulli and A. Odoni. The European air traffic flow management problem. *Transportation Science*, 41(4):431–443, 2007.

[17] OptiFrame Consortium. Report of the OptiFrame workshop: the stakeholders views, Brussels, 5/10/2016. Technical report, OptiFrame Consotium, 2016. Deliverable of the OptiFrame project.

[18] N. Pilon, S. Ruiz, A. Bujor, A. Cook, and L. Castelli. Improved flexibility and equity for airspace users during demand-capacity imbalance. In D. Schafer, editor, *Proceedings of the SESAR Innovation Days*, 2016.

[19] O. Richard, S. Constans, and R. Fondacci. Computing 4D near-optimal trajectories for dynamic air traffic flow management with column generation and branch-and-price. *Transportation Planning and Technology*, 34(5):389–411, 2011.

[20] H.D. Sherali, J.C. Smith, and A.A. Trani. An airspace planning model for selecting flight-plans under workload, safety, and equity considerations. *Transportation Science*, 36(4):387–397, 2002.

[21] S. Torres and K. L. Delpome. An integrated approach to air traffic management to achieve trajectory based operations. In *2012 IEEE/AIAA 31st Digital Avionics Systems Conference (DASC)*, pages 3E6–1–3E6–16, Oct 2012.

[22] E. L. Ulungu, J. Teghem, P. H. Fortemps, and D. Tuyttens. MOSA Method: A tool for solving multiobjective combinatorial optimization problems. *Journal of Multi-Criteria Decision Analysis*, 8:221–236, 1999.

[23] D.A. Veldhuizen and G.B. Lamont. On measuring multiobjective evolutionary algorithm performance. In *Proceedings of the 2000 Congress on Evolutionary Computation*, volume 1, pages 204–211. IEEE, 2000.

[24] R.J.D. Verbeek. D6.3: Detailed assessment of the OptiFrame computational framework for normal and disturbance cases. Technical report, OptiFrame Consortium, 02 2018. Delivearable of the OptiFrame project.

[25] T.W.M. Vossen, R. Hoffman, and A. Mukherjee. Air traffic flow management. In C. Barnhart and B. Smith, editors, *Quantitative Problem Solving Methods in The Airline Industry: A Modeling Methodology Handbook*, pages 387–455. Springer Science + Business Media, 2012.