Maximizing the number of conflict-free aircraft using mixed-integer nonlinear programming

Sonia Cafieri, David Rey Rey

HAL Id: hal-01432814

https://enac.hal.science/hal-01432814

Submitted on 12 Jan 2017

Maximizing the Number of Conflict-free Aircraft using Mixed-Integer Nonlinear Programming

Sonia Cafieri

ENAC, University of Toulouse, F-31055 Toulouse, France

David Rey

School of Civil and Environmental Engineering,
UNSW (University of New South Wales) Australia, Sydney,
2052, NSW, Australia

Abstract

We address the conflict detection and resolution problem in air traffic control, where an aircraft conflict is a loss of separation between aircraft trajectories. Conflict avoidance is crucial to ensure flight safety and remains a challenging traffic control problem. We focus on speed control to separate aircraft and consider two approaches: (i) maximize the number of conflicts resolved and (ii) identify the largest set of conflict-free aircraft. Both problems are modeled using mixed-integer nonlinear programming and a tailored greedy algorithm is proposed for the latter. Computational efficiency is improved through a pre-processing algorithm which attempts to reduce the size of the conflict resolution models by detecting the existence of pairwise potential conflicts. Numerical results are provided after implementing the proposed models and algorithms on benchmark conflict resolution instances. The results highlight the benefits of using the proposed pre-processing step as well as the versatility and the efficiency of the proposed models.

1

1   Introduction

One of the missions of air traffic control services is to ensure the safety of aircraft throughout their flight. This task is achieved by continuously monitoring aircraft trajectories, anticipating potential loss of separation, known as *conflict*, and issuing appropriate conflict resolution maneuvers. Due to the forecasted increase in flight travel volume in the next decades (see, e.g., [1]), the automation of conflict detection and resolution procedures has received a growing attention over the past few years. Indeed, the traffic growth is likely to impact air traffic controllers workload, with consequent safety issues which could lead to considerable delays.

Aircraft conflict avoidance can be achieved through a variety of maneuvers to separate aircraft trajectories. Flight heading angle deviation is the most widely used separation maneuver. Alternatively, flight level changes can be performed, even if they are usually not preferred due to passenger comfort and fuel consumption considerations. Another possibility is to adjust flights' speed subject to aerodynamical and passenger comfort constraints. In contrast to other separation maneuvers, speed control often requires some level of automation to be operated due to the (small) magnitude of the possible speed adjustments and is therefore seldom used by air traffic controllers ([2]). See [3] for a review up to the year 2000 of mathematical models for aircraft conflict avoidance.

The approach proposed in the present work aims at performing conflict avoidance by aircraft speed control and is based on mixed-integer optimization, which is attracting a growing attention in Air Traffic Management (ATM), in particular for air conflict detection and resolution problems. First approaches of this kind date back to 2002 ([4, 5]), where mixed-integer linear programs were proposed, based on aircraft separation by heading or speed changes, derived from geometrical considerations on aircraft trajectories. More recently, the works of [6, 7, 8] extend the models of [4] through mixed-integer nonlinear programs based on various aircraft separation techniques. [9] proposes a mixed-integer linear programming model for air conflict resolution which combines speed and heading maneuvers and relies on

2

a space-discretization of aircraft trajectories. Mixed-integer nonlinear programming for aircraft speed change maneuvers executed in time windows is proposed in [10]. Recent works that focus exclusively on speed regulation maneuvers also include [11, 12, 13], where a space-discretization approach is used to represent aircraft trajectories and separation. An overview of MINLP modeling for aircraft conflict avoidance is presented in [14].

In most of the above works, it is implicitly assumed that the range of the selected separation maneuvers is broad enough to resolve all conflicts. In this context, the aim is generally detecting and solving all conflicts that may occur in the monitored airspace during a time horizon, and the focus is to optimize the amount of resources—typically minimizing the deviation to aircraft nominal trajectories—used for deconfliction. However, if one restricts himself to a subset of possible separation maneuvers, it may happen that not all conflicts can be resolved. For instance, if the available separation maneuvers are limited (e.g. finite number of alternative trajectories) or if their range is tightly constrained (as in the case of speed deviations), and the number or the severity of the potential conflicts is significant (e.g. high aircraft density), then the selected separation maneuvers may not suffice to ensure the feasibility of the corresponding optimization problem where one aims at separating all aircraft. Hence, it can be of interest to optimize the number of conflicts that can be solved when a separation maneuver is applied, in order to easily distinguish between conflicts that can be solved and those that require the application of another separation strategy. This lead in [11, 12, 13, 15] to propose objective functions and models seeking to maximize the number of conflicts resolved or to minimize the total conflict duration.

The interest for this kind of approaches specially applies to deconfliction based on speed control. Speed control-based conflict resolution models emerged at the beginning of the XXI century with the advent of advanced flight management systems that provided more accurate trajectory predictions [16]. Building on this technological development, speed control methods offered a new way to improve traffic conditions by adjusting aircraft future longitudinal positions. Soon after, the concept of *subliminal speed control* was introduced on the premise that minor speed adjustments around aircraft nominal speeds were almost imperceptible to air traffic controllers. This paradigm has been validated in field experiments [17] in the context of the ERASMUS [18] project. Due to its limited impact on air traffic controllers' workload [19], conflict avoidance by subliminal speed control is considered a

promising approach to introduce some automation in ATM in the near future to improve traffic conditions using a very moderate amount of control resources. It constitutes a competitive candidate to introduce a "filter" on air traffic, prior to the use of more trajectory-intensive conflict avoidance methods (e.g. heading and altitude maneuvers) that would require either controller intervention or a fully automated traffic control paradigm. From a practical point of view, the approach proposed in this paper can be intended as a first step, acting as a pre-processing, to efficiently perform conflict avoidance.

In this paper, building on [15], we first present a model from mixed-integer nonlinear programming (MINLP) to maximize the number of solved conflicts (pairwise separated aircraft) when this maneuver is applied. This model, following [10], does not require any form of discretization, unlike the most of previous works in the area, thus resulting in a compact mixed-integer nonlinear program. The computational performance of such a MINLP is improved by a novel technique to identify potential aircraft conflicts, based on the solution of a simple concave bound-constrained continuous maximization problem. The detection of potential conflicts is carried out in a pre-processing phase where pairwise conflicts are identified before a complete model, *i.e.* with all the relevant pairs of aircraft, is solved. Then, based on the observation that for a large number of aircraft in the observed air sector it may be difficult to solve many conflicts simultaneously, and that, even not imposing solving all conflicts, the problem can be computational demanding, we propose a novel approach to identify the largest set of conflict-free aircraft. To this aim, we first propose a MINLP model and then, to improve its computational sustainability, a greedy algorithm that iteratively removes conflicting aircraft from the original set containing all the observed aircraft. Numerical experiments show that the proposed approach allows us to find competitive solutions.

The paper is organized as follows. In Section 2 we first discuss about the mathematical representation of aircraft trajectories separation, then we propose a mathematical programming formulation, from MINLP, for aircraft conflict avoidance achieved through speed regulation. Section 3 focuses on conflict detection, and presents an algorithm to identify potential aircraft conflicts when speed control is applied. In Section 4 we propose an approach to identify the largest set of aircraft that are conflict-free when speed regulation is applied. This is done through a MINLP problem, as well as a greedy aircraft-removal algorithm tailored on the problem. In Section 5 we present

4

and discuss the results of numerical experiments carried out to validate the proposed approaches. Section 6 draws some conclusions and presents some perspectives.

2 Problem Modeling

In this section, we first discuss the mathematical representation of the crucial condition of separation between aircraft trajectories. Then, we propose a mathematical programming formulation for aircraft conflict avoidance, where the above separation condition clearly represents the main constraint on the decision variables.

2.1 Representation of Aircraft Separation

Let us consider a set A of n aircraft flying during their cruise flight in a given air sector, all at the same flight level. Let $\mathbf{x}_i(t)$ be the vector representing the position of flight i at time t. The relative position of aircraft i and j at time t can be represented as

$$\mathbf{x}_{ij}^r(t) = \mathbf{x}_i(t) - \mathbf{x}_j(t).$$

Let d be the horizontal separation standard, aircraft i and j are separated if

$$\|\mathbf{x}_{ij}^r(t)\| \geq d, \forall t \tag{1}$$

where $\| \cdot \|$ is the Euclidean norm in the two-dimensional space formed by aircraft trajectories. Assuming that uniform motion laws apply, $\mathbf{x}_{ij}^r(t)$ can also be expressed as the sum of the initial relative position of flights i and j, $\mathbf{x}_{ij}^{r0}$, and their relative speed, $\mathbf{v}_{ij}^r$: $\mathbf{x}_{ij}^r(t) = \mathbf{x}_{ij}^{r0} + \mathbf{v}_{ij}^r t$. Therefore squaring Equation (1) we obtain the separation condition

$$f_{ij}(t) \equiv \|\mathbf{v}_{ij}^r\|^2 t^2 + 2\mathbf{x}_{ij}^{r0} \cdot \mathbf{v}_{ij}^r t + \|\mathbf{x}_{ij}^{r0}\|^2 - d^2 \geq 0 \tag{2}$$

where $\cdot$ is the inner product in the Euclidean space. From Equation (2), $f_{ij}(t)$ is a 2nd order convex polynomial in t which is minimal when its derivative vanishes:

$$f_{ij}'(t) = 0 \quad \Leftrightarrow \quad t_{ij}^m = \frac{-\mathbf{x}_{ij}^{r0} \cdot \mathbf{v}_{ij}^r}{\|\mathbf{v}_{ij}^r\|^2} \tag{3}$$

t_{ij}^m is the time at which $f_{ij}(t)$ is minimal and therefore represents the instant at which the aircraft relative position is minimal. Following [10], we observe that, substituting t_{ij}^m in $f_{ij}(t)$, the separation condition (2) can be represented as

$$\|\mathbf{v}_{ij}^r\|^2(\|\mathbf{x}_{ij}^{r0}\|^2 - d^2) - (\mathbf{x}_{ij}^{r0} \cdot \mathbf{v}_{ij}^r)^2 \geq 0. \tag{4}$$

that does not depend anymore on t.

Observe that the discriminant of the 2nd order equation $f_{ij}(t) = 0$, Δ_{ij}, is equal to

$$\Delta_{ij} = 4(\mathbf{x}_{ij}^{r0} \cdot \mathbf{v}_{ij}^r)^2 - 4\|\mathbf{v}_{ij}^r\|^2(\|\mathbf{x}_{ij}^{r0}\|^2 - d^2), \tag{5}$$

so the following cases can occur:

1. if $\Delta_{ij} \leq 0$, then $f_{ij}(t)$ has no roots and therefore $\|\mathbf{x}_{ij}^r(t)\| \geq d, \forall t$, hence aircraft are separated

2. otherwise, aircraft are in conflict. Let t_{ij}' and t_{ij}'' be the roots of $f_{ij}(t) = 0$:

$$t_{ij}' = \frac{-2(\mathbf{x}_{ij}^{r0} \cdot \mathbf{v}_{ij}^r) - \sqrt{\Delta_{ij}}}{2\|\mathbf{v}_{ij}^r\|^2} \quad \text{and} \quad t_{ij}'' = \frac{-2(\mathbf{x}_{ij}^{r0} \cdot \mathbf{v}_{ij}^r) + \sqrt{\Delta_{ij}}}{2\|\mathbf{v}_{ij}^r\|^2} \tag{6}$$

where t_{ij}' and t_{ij}'' are the instants of entry and exit from the conflict. In this case, aircraft are separated only when both roots are negative (the conflict was in the past). We will refer to the above conditions as *condition 1* and *condition 2* on Δ_{ij}. Hence, note that imposing the separation condition (4) is equivalent as imposing $\Delta_{ij} \leq 0$. This condition is employed in the mathematical model presented below.

2.2 Maximizing the Number of Solved Conflicts

In [15] a nonlinear mixed-integer program was introduced to maximize the number of aircraft conflicts that can be solved by only performing speed control. Here we recall the main elements of such a model and propose some extensions. The main motivation for this kind of modeling is that, as discussed in Section 1, when a subset of possible separation maneuvers is considered, we are not guaranteed to be able to solve all conflicts, i.e. the resulting mathematical program may be infeasible. This may specially occur for approaches focused on speed control, and is reinforced by the request of a subliminal control which imposes minor speed variations, thus restricting the

feasible region of the problem. In contrast, alternative separation maneuvers, such as heading angle changes and vertical maneuvers, are generally sufficient to resolve all conflicts.

As we want to maximize the number of solved conflicts, that is equivalent to maximize the number of aircraft pairs that are separated, we introduce the following binary decision variable, for each pair (i, j):

$$z_{ij} = \begin{cases} 1 \text{ if aircraft } i \text{ and } j \text{ are separated} \\ 0 \text{ otherwise} \end{cases} \tag{7}$$

Since we focus on speed control to separate aircraft trajectories, the other main decision variables are continuous variables $q_i, \forall i \in A$, representing the speed variation rate of each aircraft, expressed as a percentage of its norm. A value $q_i = 1$ corresponds to an unchanged speed. Each q_i is restricted to a fixed range $[\underline{q}_i, \overline{q}_i]$, which ensure a subliminal control when $\underline{q}_i$ and $\overline{q}_i$ are chosen in such a way that the aircraft speed is between -6% and $+3\%$ of the original speed (see [17]). The actual velocity of aircraft i is then given by $\mathbf{v}_i q_i$, with $\mathbf{v}_i$ the airspeed vector, and the relative speed of aircraft i and j is expressed by

$$\mathbf{v}_{ij}^r = \mathbf{v}_i q_i - \mathbf{v}_j q_j. \tag{8}$$

Using the above condition $\Delta_{ij} \leq 0$, and the definition of z_{ij}, aircraft separation can then be represented using the following constraint:

$$(2z_{ij} - 1)\|\mathbf{v}_{ij}^r\|^2(\|\mathbf{x}_{ij}^{r0}\|^2 - d^2) \geq (2z_{ij} - 1)(\mathbf{x}_{ij}^{r0} \cdot \mathbf{v}_{ij}^r)^2 \qquad \forall i, j \in A : i < j \tag{9}$$

Note that Eq. (9) does not impose a separation condition to all pairs of aircraft. It is instead used to account for pairs of aircraft (i, j) for which the separation condition is verified ($z_{ij} = 1$), either thanks to an employed speed control (at least one between q_i and q_j is not equal to 1), or because i and j have separated trajectories regardless of their speed variations. Remark that (9) does not guarantee to count all pairs of aircraft that are separated (for instance, those for which condition 2 on Δ_{ij} holds).

The mathematical programming model seeking to maximize the number of conflicts that are solved amongst all pairs of aircraft using speed control can be summarized as follows, in Model 1.

Model 1 (Max.VC).

$$\max \sum_{i,j \in A: i < j} z_{ij} \tag{10}$$

subject to

$$(2z_{ij} - 1)\|\mathbf{v}_{ij}^r\|^2(\|\mathbf{x}_{ij}^{r0}\|^2 - d^2) \geq (2z_{ij} - 1)(\mathbf{x}_{ij}^{r0} \cdot \mathbf{v}_{ij}^r)^2 \qquad \forall i,j \in A: i < j \tag{11}$$

$$\mathbf{v}_{ij}^r = \mathbf{v}_i q_i - \mathbf{v}_j q_j \qquad \forall i,j \in A: i < j \tag{12}$$

$$\underline{q}_i \leq q_i \leq \bar{q}_i \qquad \forall i \in A \tag{13}$$

$$z_{ij} \in \{0,1\} \qquad \forall i,j \in A: i < j \tag{14}$$

The proposed model can be enriched adjoining further conditions to check for aircraft separation. In particular, this can be done taking into account the geometry of the considered trajectories. Aircraft trajectories are usually represented by straight lines, that, for a pair of aircraft in an air sector observed during a time window, can be supposed intersecting in no more than one point. Thus, the inner product $\mathbf{x}_{ij}^{r0} \cdot \mathbf{v}_{ij}^r$ can be used to identify if aircraft are converging or diverging. If aircraft are initially separated and their trajectories are diverging, then they will remain separated. Namely:

- if $t_{ij}^m \geq 0$, *i.e.* the instant of minimal separation is in the future, then aircraft are converging and they can be involved in a conflict;

- if $t_{ij}^m < 0$, *i.e.* the instant of minimal separation is in the past, then aircraft are diverging and collision has therefore already been avoided, although aircraft could still be in conflict (violate the separation standards).

Recall that the time of minimal separation is given by $t_{ij}^m = -\mathbf{x}_{ij}^{r0} \cdot \mathbf{v}_{ij}^r/\|\mathbf{v}_{ij}^r\|^2$. If $\|v_{ij}^r\| = 0$, then aircraft trajectories are parallel and aircraft fly at the same speed. If the smallest distance between aircraft trajectories is less than the separation norm, there is a risk of conflict. Two cases may occur: if aircraft fly in the same direction, then their separation can be determined by comparing their initial distance with the separation norm; otherwise, aircraft fly in opposite directions and will be in conflict upon crossing one another. In both cases, this situation can be easily checked in a pre-processing step, and the pair of aircraft can be omitted from the set of considered pairs. Assume that $\|\mathbf{v}_{ij}^r\| > 0$, the sign of t_{ij}^m depends on the sign of the inner product $\mathbf{x}_{ij}^{r0} \cdot \mathbf{v}_{ij}^r$ which can be determined using a linear combination of vector components. Let y_{ij} be the binary variable defined as

$$y_{ij} = \begin{cases} 1 \text{ if aircraft } i \text{ and } j \text{ are diverging and } \|\mathbf{x}_{ij}^{r0}\| \geq d \\ 0 \text{ otherwise} \end{cases}$$

The following constraint can then be adjoined to the formulation:

$$(2y_{ij} - 1)(\mathbf{x}_{ij}^{r0} \cdot \mathbf{v}_{ij}^{r}) \geq 0 \tag{15}$$

To ensure that a pair of diverging is separated we need to make sure that both aircraft are initially separated. We define the sets $P_S \equiv \{i, j \in A : i < j, \|\mathbf{x}_{ij}^{r0}\| \geq d\}$ and $P_{NS} \equiv \{i, j \in A : i < j, \|\mathbf{x}_{ij}^{r0}\| < d\}$. Constraint (15) is indexed on set P_S and, in a pre-processing phase, we set $y_{ij} = 0 \; \forall (i, j) \in P_{NS}$. Note that, although Constraint (15) is nonlinear, it can be linearized using a Fortet's reformulation [20] for products of binary and continuous variables.

Finally, we account for a separated pair (i, j) either when $\Delta_{ij} \leq 0$ or the two trajectories are diverging. The *or* condition is modeled introducing a new variable $w_{ij} \geq 0$ and corresponding constraints

$$w_{ij} \geq z_{ij}, \quad w_{ij} \geq y_{ij}, \quad w_{ij} \leq z_{ij} + y_{ij} \qquad \forall i, j \in A : i < j$$

The resulting model is summarized in Model 2.

Model 2 (Max.VC+)**.**

$$\max \sum_{i,j \in A : i < j} w_{ij} \tag{16}$$

subject to

$$(2z_{ij} - 1)\|\mathbf{v}_{ij}^{r}\|^2(\|\mathbf{x}_{ij}^{r0}\|^2 - d^2) \geq (2z_{ij} - 1)(\mathbf{x}_{ij}^{r0} \cdot \mathbf{v}_{ij}^{r})^2 \qquad \forall i, j \in A : i < j \tag{17}$$

$$\mathbf{v}_{ij}^{r} = \mathbf{v}_i q_i - \mathbf{v}_j q_j \qquad \forall i, j \in A : i < j \tag{18}$$

$$\underline{q}_i \leq q_i \leq \overline{q}_i \qquad \forall i \in A \tag{19}$$

$$(2y_{ij} - 1)(\mathbf{x}_{ij}^{r0} \cdot \mathbf{v}_{ij}^{r}) \geq 0 \qquad \forall (i, j) \in P_S \tag{20}$$

$$w_{ij} \geq z_{ij} \qquad \forall i, j \in A : i < j \tag{21}$$

$$w_{ij} \geq y_{ij} \qquad \forall i, j \in A : i < j \tag{22}$$

$$w_{ij} \leq z_{ij} + y_{ij} \qquad \forall i, j \in A : i < j \tag{23}$$

$$z_{ij}, y_{ij} \in \{0, 1\} \qquad \forall i, j \in A : i < j \tag{24}$$

$$0 \leq w_{ij} \leq 1 \qquad \forall i, j \in A : i < j \tag{25}$$

Model 2 is a mixed-integer nonlinear program that contains more variables and (linear) constraints than Model 1, while including an additional way to identify separated aircraft trajectories. The impact of this additional feature is discussed in Section 5.

3 Conflict Detection

Aircraft separation constraints appearing in Models 1 and 2 are indexed on all pairs of aircraft, which implies a large number of constraints, $n(n-1)/2$, for increasing n. Motivated by the observation that some pairs of trajectories may always be separated, regardless of the possible speed variations, the natural approach to reduce the size of the optimization problem of interest is to try to identify some pairs of aircraft for which the induced separation constraints can be omitted. In the remainder of this section, we define aircraft *nominal* and *potential* conflicts, and propose a novel algorithm to identify potential conflicts, based on the solution of a bound-constrained convex optimization problem.

Recall that, for each pair of aircraft, their trajectories are separated when the condition $\Delta_{ij} \leq 0$ holds, where Δ_{ij} is the discriminant of $f_{ij}(t)$. Let $\Delta_{ij}(\mathbf{q})$ be the function

$$\mathbf{q} = (q_i, q_j) \in \mathbb{R}^2 \to \Delta_{ij}(\mathbf{q}) \in \mathbb{R}$$

representing the value of Δ_{ij} for a combination of speed variations $\mathbf{q} = (q_i, q_j)$ of aircraft i and j. We define the set of *nominal conflicts* N as the set of pairs $(i, j) \in A, i < j$ which violate the separation distance when aircraft nominal speeds are maintained, *i.e.* no speed control is applied. It follows from the definition that N can be determined by evaluating for each pair (i, j) the function $\Delta_{ij}(\mathbf{q})$ at the point $(q_i, q_j) = (1, 1)$, that is for $\mathbf{q}$ corresponding to no variations of aircraft speeds. This procedure is summarized in Algorithm 1.

Algorithm 1: Nominal Conflict Detection

> **Input**: A set of aircraft A with initial position $\mathbf{x}_{ij}^{r0}$ and speed $\mathbf{v}_{ij}^{r}$,
>   separation norm d, a speed control vector $\mathbf{q}$
> **Output**: A set of nominal conflicts N
> $N \leftarrow \emptyset$
> **for** $i \in A, j \in A : i < j$ **do**
> > **if** $\Delta_{ij}(\mathbf{q} = \mathbf{1}) > 0$ **then**
> > > $N \leftarrow N \cup \{(i, j)\}$
>
> **return** N

The set N, however, cannot be used to simply reduce the number of aircraft pairs on which separation constraints are indexed. Solving Model 1

by indexing its constraints using the set N may, indeed, solve some or all of these conflicts, but the newly obtained speed control vector $\mathbf{q}$ may also result in generating new conflicts among pairs of aircraft initially separated. To account for this issue, we seek to determine the set of *potential conflicts* P which contains all the pairs of aircraft that could possibly result in a loss of separation when speed control is applied. This set is then defined considering all possible values of $\mathbf{q} = (q_i, q_j)$ for each pair of aircraft. To decide the existence of a potential conflict for a pair (i, j), let us consider the following two-variables problem:

$$c_{ij} = \max \left\{ \Delta_{ij}(\mathbf{q}) : \underline{\mathbf{q}} \leq \mathbf{q} \leq \overline{\mathbf{q}} \right\}, \quad \forall i, j \in A : i < j \qquad (26)$$

If $c_{ij} > 0$, then there exists at least a combination of speed variations such that $\Delta_{ij}(\mathbf{q}) > 0$, which means that aircraft may be in conflict—the pair (i, j) then belongs to P. Otherwise, aircraft i and j cannot be in conflict (regardless of the speed variations, they will always remain separated), and the associated separation constraint can be omitted in the optimization problem. To identify the pairs belonging to P, we need to solve problem (26) for all (i, j). We next show that the conditions under which $\Delta_{ij}(\mathbf{q})$ is a concave function only depend on the initial relative position of the aircraft and their initial velocity.

Proposition 1. *Let $i, j \in A : i < j$ be two aircraft and let q_i and q_j be their respective speed variation rate. We index the vector components by 1 and 2 respectively. The function $\Delta_{ij}(\mathbf{q})$ for all $\mathbf{q}$ such that $\underline{\mathbf{q}} \leq \mathbf{q} \leq \overline{\mathbf{q}}$ is concave if and only if Conditions C_1, C_2 and C_3 hold:*

$$C_1: \qquad v_{i,1}^2(d^2 - (x_{ij,2}^{r0})^2) + v_{i,2}^2(d^2 - (x_{ij,1}^{r0})^2) \leq 0 \qquad (27)$$

$$C_2: \qquad v_{j,1}^2(d^2 - (x_{ij,2}^{r0})^2) + v_{j,2}^2(d^2 - (x_{ij,1}^{r0})^2) \leq 0 \qquad (28)$$

$$C_3: \qquad (v_{i,1}v_{j,2} - v_{i,2}v_{j,1})^2((x_{ij,1}^{r0})^2 - d^2)((x_{ij,2}^{r0})^2 - d^2) \geq 0 \qquad (29)$$

Proof. Recall that:

$$\Delta_{ij}(\mathbf{q}) = 4(\mathbf{x}_{ij}^{r0} \cdot \mathbf{v}_{ij}^r)^2 - 4\|\mathbf{v}_{ij}^r\|^2(\|\mathbf{x}_{ij}^{r0}\|^2 - d^2)$$
$$= 4(\mathbf{x}_{ij}^{r0} \cdot (\mathbf{v}_i q_i - \mathbf{v}_j q_j))^2 - 4\|\mathbf{v}_i q_i - \mathbf{v}_j q_j\|^2(\|\mathbf{x}_{ij}^{r0}\|^2 - d^2)$$

The factor 4 in the expression of $\Delta_{ij}(\mathbf{q})$ can be ignored since it does not

change the sign of c_{ij}. Therefore $\Delta_{ij}(\mathbf{q})$ can be expressed as

$$\Delta_{ij}(\mathbf{q}) = \left(x^{r0}_{ij,1}(v_{i,1}q_i - v_{j,1}q_j) + x^{r0}_{ij,2}(v_{i,2}q_i - v_{j,2}q_j)\right)^2 \\ - \left((v_{i,1}q_i - v_{j,1}q_j)^2 + (v_{i,2}q_i - v_{j,2}q_j)^2\right)\left((x^{r0}_{ij,1})^2 + (x^{r0}_{ij,2})^2 - d^2\right)$$

The second partial derivatives of $\Delta_{ij}(\mathbf{q})$ are thus:

$$\frac{\partial^2 \Delta_{ij}(\mathbf{q})}{\partial q_i^2} = 2v_{i,1}^2(d^2 - (x^{r0}_{ij,2})^2) + 2v_{i,2}^2(d^2 - (x^{r0}_{ij,1})^2)$$

$$\frac{\partial^2 \Delta_{ij}(\mathbf{q})}{\partial q_j^2} = 2v_{j,1}^2(d^2 - (x^{r0}_{ij,2})^2) + 2v_{j,2}^2(d^2 - (x^{r0}_{ij,1})^2)$$

$$\frac{\partial^2 \Delta_{ij}(\mathbf{q})}{\partial q_i \partial q_j} = 2v_{i,1}v_{j,1}((x^{r0}_{ij,2})^2 - d^2) + 2v_{i,2}v_{j,2}((x^{r0}_{ij,1})^2 - d^2)$$

Let $D_{k,n}$ be the nth principal minor of order k of the Hessian matrix of function $\Delta_{ij}(\mathbf{q})$. To show that $\Delta_{ij}(\mathbf{q})$ is concave, it suffices to show that its Hessian matrix is negative semidefinite, $i.e.$ that $(-1)^k D_{k,n} > 0$, for each principal minor. The first order principal minors are $D_{1,1} = \partial^2 \Delta_{ij}(\mathbf{q})/\partial q_i^2$ and $D_{1,2} = \partial^2 \Delta_{ij}(\mathbf{q})/\partial q_j^2$. By construction, if C_1 and C_2 hold, then $-D_{1,1} \geq 0$ and $-D_{1,2} \geq 0$. The second order principal minor is

$$D_{2,1} = \frac{\partial^2 \Delta_{ij}(\mathbf{q})}{\partial q_i^2}\frac{\partial^2 \Delta_{ij}(\mathbf{q})}{\partial q_j^2} - \left(\frac{\partial^2 \Delta_{ij}(\mathbf{q})}{\partial q_i \partial q_j}\right)^2 \\ = (v_{i,1}v_{j,2} - v_{i,2}v_{j,1})^2((x^{r0}_{ij,1})^2 - d^2)((x^{r0}_{ij,2})^2 - d^2)$$

Hence if C_3 holds, then $D_{2,1} \geq 0$ and $\Delta_{ij}(\mathbf{q})$ is concave.

$\square$

Although function $\Delta_{ij}(\mathbf{q})$ is not always concave, it is reasonable to assume that some pairs of aircraft will fall into this category. An immediate consequence of Proposition 1 is that, assuming that the initial state of flights meets the conditions C_1, C_2 and C_3, the value of c_{ij} can be determined a $priori$ using a generic NLP solver, since any local maximum of a concave function is also a global optimum. Figure 1 depicts an example of potential conflict amongst two flights with a look-ahead of 15 min and trajectories intersecting with a right angle; the maximum of the function is attained when both flights fly at the same speed. Further, its concavity can be decided analytically by evaluating Conditions C_1, C_2 and C_3.

12
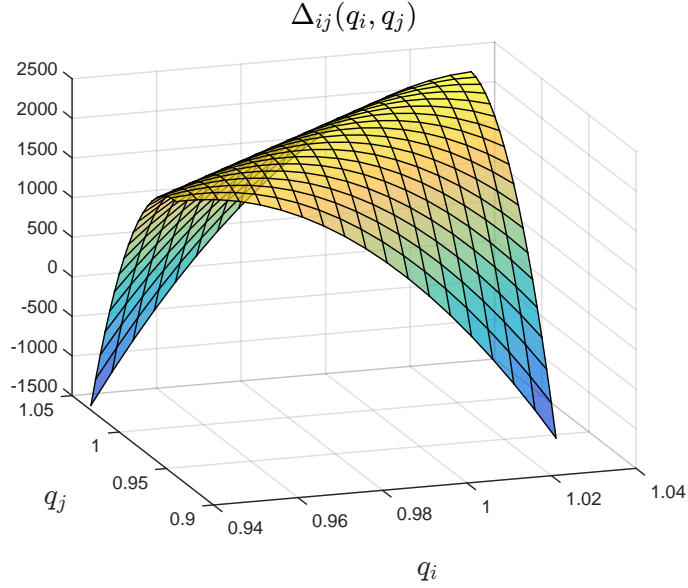
Figure 1: Plot of function $\Delta_{ij}(\mathbf{q})$ for two flights whose trajectories intersect with a right angle and distance of 100 NM from the intersection point. The separation distance is 5 NM and the speed variation is $[0.94, 1.03]$.

We hereby denote $C^{ij}_{1,2,3}$ the binary parameter defined as:

$$C^{ij}_{1,2,3} = \begin{cases} 1 \text{ if } C_1, C_2 \text{ and } C_3 \text{ hold for } i, j \in A : i < j \\ 0 \text{ otherwise} \end{cases}$$

Algorithm 2 returns a set P of pairs of aircraft that can be used to index the separation constraints in Model 1. Namely, in such constraints we can substitute the statement $\forall i < j$ by $(i, j) \in P$.

13

Algorithm 2: Potential Conflict Detection

Input: A set of aircraft A with initial position $\mathbf{x}_{ij}^{r0}$ and speed $\mathbf{v}_{ij}^r$,
  speed control bounds $[\underline{\mathbf{q}}, \overline{\mathbf{q}}]$

Output: A set of potential conflicts P

$P \leftarrow \{i, j \in A : i < j\}$

for $(i, j) \in P$ **do**

    if $C_{1,2,3}^{ij} = 1$ **then**

        $c_{ij} \leftarrow \max \left\{ \Delta_{ij}(\mathbf{q}) : \underline{\mathbf{q}} \leq \mathbf{q} \leq \overline{\mathbf{q}} \right\}$

        if $c_{ij} \leq 0$ **then**

            $P \leftarrow P \setminus \{(i, j)\}$

return P

The impact of the detection of potential conflicts on the numerical solution of the MINLP Model 1 is discussed in Section 5.

4   Largest Conflict-free Aircraft Set

In this section, we present an approach to identify the largest set of aircraft among which all conflicts are solved when speed control is applied. In other words, an aircraft belongs to such a set when its trajectory is separated from that of all other aircraft in the observed air sector (and separation is achieved by speed control). Conflict resolution through speed control as represented by Models 1 and 2 seeks to solve a maximum number of conflicts given a set of aircraft. These models can provide trajectory adjustments that minimize the number of unresolved conflicts and thus work in the aircraft pairs space. Therefore, if not all conflicts can be resolved by speed control only, then their outcome indicates which pairs of aircraft remain in potential conflict. We also observe that, depending on the traffic configuration, it may be critical to work in the aircraft space, *i.e.*, identify which aircraft should be selected for additional separation maneuvers (*e.g.* heading angle deviation, altitude change). To address this operational challenge, we introduce a new conflict resolution model which focuses on individual aircraft behavior instead of considering pairs of aircraft. Specifically, we propose to identify the largest set of conflict-free aircraft using speed control. This is equivalent to remove the minimal number of conflicting aircraft from the initial aircraft set. From an air traffic control perspective, this approach has a simple interpretation: the trajectories of the retained aircraft are separated while the

14

removed aircraft can be handled by air traffic control services for trajectory management through heading and/or altitude change.

We first propose a mathematical programming formulation for the identification of the largest conflict-free aircraft set. Then, for the sake of computational efficiency, we propose an aircraft-removal algorithm aiming at solving the same problem.

4.1 Largest Conflict-free Set: MINLP

The mathematical programming formulation is based on Model 1. A new binary variable is introduced for each aircraft to decide if it belongs to the conflict-free set. It is defined, for each $i \in A$, as follows:

$$s_i = \begin{cases} 1 \text{ if aircraft } i \text{ is in the conflict-free set} \\ 0 \text{ otherwise} \end{cases} \tag{30}$$

Recall that, for each pair of aircraft i and j, the decision variable z_{ij} is used to represent their separation. If i and j are separated ($z_{ij} = 1$), then both i and j could be included in the conflict-free set (if they are also separated from all other aircraft in the conflict-free set). In this case, variables s_i and s_j are free. On the other hand, if i and j are not separated ($z_{ij} = 0$), then both aircraft cannot belong to the conflict-free set. This means that either s_i or s_j must take the value 0. Reciprocally, if both aircraft belong to the conflict-free set ($s_i = s_j = 1$), then they are separated from each other and so $z_{ij} = 1$. Therefore, the linking constraint between variables z_{ij}, s_i and s_j reads

$$s_i + s_j \leq 1 + z_{ij} \qquad \forall i, j \in A : i < j \tag{31}$$

Equation (31) is then added to the constraints of Model 1. The objective function is a linear function expressed by the sum of s_i, $\forall i$, and has to be maximized. The mathematical programming model seeking to maximize the size of the set of conflict-free aircraft using speed control can be summarized as follows, in Model 3.

Model 3 (Max.CS).

$$\max \sum_{i \in A} s_i \tag{32}$$

subject to

$$s_i + s_j \leq 1 + z_{ij} \qquad\qquad \forall i, j \in A : i < j \quad (33)$$

$$(2z_{ij} - 1)\|\mathbf{v}_{ij}^r\|^2(\|\mathbf{x}_{ij}^{r0}\|^2 - d^2) \geq (2z_{ij} - 1)(\mathbf{x}_{ij}^{r0} \cdot \mathbf{v}_{ij}^r)^2 \qquad \forall i, j \in A : i < j \quad (34)$$

$$\mathbf{v}_{ij}^r = \mathbf{v}_i q_i - \mathbf{v}_j q_j \qquad\qquad \forall i, j \in A : i < j \quad (35)$$

$$\underline{q}_i \leq q_i \leq \bar{q}_i \qquad\qquad \forall i \in A \quad (36)$$

$$s_i \in \{0, 1\} \qquad\qquad \forall i \in A \quad (37)$$

$$z_{ij} \in \{0, 1\} \qquad\qquad \forall i, j \in A : i < j \quad (38)$$

Model 3 is again a MINLP, that contains $n(n-1)/2$ more linear constraints (in (33)) and n more integrality constraints (on variables s_i) than Model 1. Similarly as for the case of the Model 1, Model 3 can be enhanced by introducing variables y_{ij} and w_{ij} and appending Constraints (20) to (23) to separate pairs based on aircraft divergence. We refer to the resulting model as Model 4 (Max.CS+).

We next propose an algorithm to identify the largest conflict-free set.

4.2 Largest Conflict-free Set: Algorithm

Alternatively to solve the above MINLP Model 3, we propose an algorithm that aims at removing the minimal number of conflicting aircraft from the initial aircraft set. It provides a "greedy" method for aircraft-removal, iteratively removing the aircraft which penalizes the most other aircrafts' trajectories.

For this section it is convenient to write the conflict resolution model using a set $C \subseteq P$ to represent the set of current conflicts and a set $A_C \subseteq A$ to represent the set of current aircraft. Specifically, we use C instead of P and A_C instead of A in Model 1. These sets are updated during the execution of the algorithm.

The algorithm starts by solving the MINLP problem 1, with constraints defined over C and A_C, using the set of all aircraft and the set of nominal conflicts obtained by executing Algorithm 1. Then, the algorithm iteratively checks for unsolved conflicts and terminates if there are none. Otherwise, if all the current conflicts C have been solved, the set of not solved conflicts nS is

added to C; if one or more current conflict has not been solved, the algorithm selects an aircraft to be removed. This selection is based on two sequential criteria: (i) the number of times each aircraft is involved in an unresolved conflict and, if there is more than one aircraft maximizing this criterion, (ii) the number of times each of these aircraft are involved in a potential conflict. For each solve, $z^\star$ denotes the value of the objective function and $\mathbf{q}^\star$ the current optimal solution. The iterative process terminates when there are no more elements in the set nS. The algorithm is described in Algorithm 3.

Algorithm 3: Algorithm for finding the Largest Conflict-free Set

Input: A set of aircraft A with initial position $\mathbf{x}_{ij}^{r0}$ and speed $\mathbf{v}_{ij}^{r}$, separation norm d, speed control vector $\mathbf{q}$ and bounds $[\underline{\mathbf{q}}, \overline{\mathbf{q}}]$

Output: A conflict-free aircraft set $A_C \subseteq A$ and an optimized speed control vector $\mathbf{q}^\star$ for these aircraft

$N \leftarrow$ Algorithm $1(A, \mathbf{x}_{ij}^{r0}, \mathbf{v}_{ij}^{r}, d, \mathbf{q} = \mathbf{1})$
$P \leftarrow$ Algorithm $2(A, \mathbf{x}_{ij}^{r0}, \mathbf{v}_{ij}^{r}, d, [\underline{\mathbf{q}}, \overline{\mathbf{q}}])$
$nS \leftarrow N$
$C \leftarrow N$
$A_C \leftarrow A$
while $|nS| \neq 0$ **do**

    // solve Model 1 or Model 2 with current aircraft and conflicts
    $z^\star \leftarrow$ solve Model 1 or Model 2
    // find not solved conflicts
    $nS \leftarrow$ Algorithm $1(A_C, \mathbf{x}_{ij}^{r0}, \mathbf{v}_{ij}^{r}, d, \mathbf{q}^\star)$
    // if all current conflicts solved, add not solved to C
    if $z^\star = |C|$ **then**
        $C \leftarrow C \cup nS$
    // else, remove aircraft most present in not solved conflicts
    else
        $\mu \leftarrow 0$, $\nu \leftarrow 0$
        for $(i, j) \in C$ **do**
            if $z_{ij}^\star = 0$ **then**
                $\mu_i \leftarrow \mu_i + 1$
                $\mu_j \leftarrow \mu_j + 1$
        $\bar{\mu} \leftarrow \arg\max\{\mu_i : i \in A_C\}$
        if $|\bar{\mu}| = 1$ **then** $r \leftarrow i \in \bar{\mu}$;
        // if many, remove aircraft most present in potential conflicts
        else
            for $k \in \bar{\mu}$ **do**
                for $(i, j) \in P$ **do**
                    if $i = k$ **then** $\nu_i \leftarrow \nu_i + 1$;
                    if $j = k$ **then** $\nu_j \leftarrow \nu_j + 1$;
            // if many, choose at random within the obtained subset
            $r \leftarrow rand(\arg\max\{\nu_i : i \in \bar{\mu}\})$
        $A_C \leftarrow A_C \setminus \{r\}$
return $A_C, \mathbf{q}^\star$

5 Numerical Results

5.1 Testing environment

All considered mathematical programming models are implemented using the AMPL modeling language ([21]).

MINLP problems (Models 1, 2 and 3) are solved using the global optimization COIN-OR solver COUENNE ([22]), that implements a spatial Branch-and-Bound algorithm based on convex relaxations, with a 300s time limit for independent solves and a 60s time limit for solves within Algorithm 3. For the detection of potential conflicts, IPOPT ([23]), implementing an Interior-Point method, is used to solve the NLPs for each pair of trajectories that form a concave program. All tests are performed on a Windows 64-bit machine with an i7 processor, a CPU at 2.9 GHz and 8 Gb of RAM.

Numerical experiments are carried out on randomly generated instances. Two types of initial aircraft configurations are considered:

RCP instances : They are generated by placing n aircraft with the same speed (400 NM/h) in a 2-dimensional airspace represented by a circle of radius 100 NM or 200 NM, which headings are randomly and uniformly deviated with an angle comprised between $-30°$ and $+30°$ from the diameter of the circle. These test problems are henceforth referred to as Random Circle Problems (RCP) instances (similar instances have been used to test conflict resolution models by [24, 12, 13]).

RTP instances : They are generated by placing n aircraft with the same speed (400 NM/h) in a 2-dimensional airspace represented by a square of edge 100 NM with random headings. These test problems are designed to represent random traffic configurations where some aircraft may be diverging and/or initially violating the 5 NM separation standard and are henceforth referred to as Random Trajectory Problems (RTP) instances (similar instances have been used to test conflict resolution models by [25]).

Figure 2 depicts two benchmark problems: an RCP instance with 20 aircraft (2a) and an RTP instance with 20 aircraft (2b). RCP instances seek to represent high-density traffic configurations as no pair of aircraft is initially diverging, *i.e.* for each pair $(i, j) \in A$, the instant of minimal separation is always in the future $t_{ij}^m > 0$. We use Model 1 to resolve a maximum number of

19

(a) An RCP instance with 20 aircraft   (b) An RTP instance with 20 aircraft
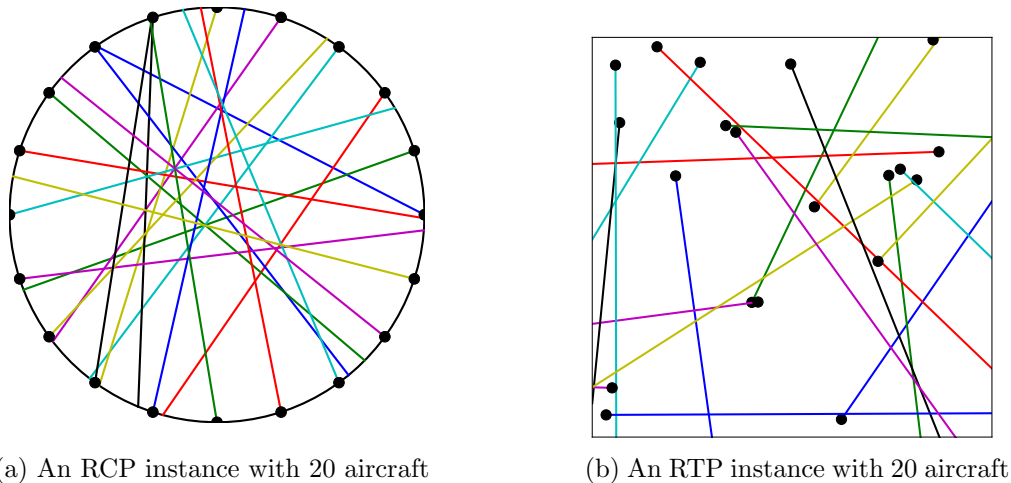
Figure 2: Illustration of Conflict Resolution benchmark problems

conflicts in RCP instances. In turn, RTP instances are designed to represent random aircraft configurations where aircraft may be converging or diverging. In this case, Model 2 is selected for conflict resolution since some pairs of aircraft may be separated using the y_{ij} variables. In all tests we consider a subliminal speed control range, *i.e.*, we set $\underline{q}_i = 0.94$ and $\bar{q}_i = 1.03$. This corresponds to speed variations comprised between -6% to $+3\%$ of aircraft nominal speed (400 NM/h).

We next present results obtained using Models 1 and 2 with and without pre-processing the set of potential pairwise conflicts using Algorithm 2.

5.2 Results: Maximizing the number of resolved conflicts, with and without potential conflict detection

Tables 1 and 2 summarize the results obtained solving Model 1 with and without pre-processing data by detecting potential conflicts (Algorithm 2) on 20 RCP instances with 10 and 20 aircraft, respectively. Specifically, for each instance, Model 1 is first solved by indexing the constraints on all pairs of aircraft (columns *without CD*), and then by indexing the constraints on the set P of potential conflicts, detected by running Algorithm 2 (columns *with CD*). The headings are as follows: Pairs is the total number of pairs of aircraft, $|N|$ is the cardinality of set N (nominal conflicts), $|P|$ is the cardinality of set P (potential conflicts), $\sum C_{1,2,3}^{ij}$ gives the number of pairs of

aircraft which satisfy Conditions C_1, C_2 and C_3 (corresponding to a concave program), R_P is the percentage of conflicting aircraft with respect to the total number of pairs, Time CD is the total pre-processing time (*i.e.* running Algorithm 2) expressed in seconds. The next four columns give the objective value and the runtime of Model 1 without and with pre-processing, respectively. The three right-most columns provide information on the solution *with CD*: Nb. Acc. is the number of aircraft accelerated, Nb. Dec. is the number of aircraft decelerated and Avg. Dev. is the average deviation in absolute value from aircraft nominal speed expressed in % *i.e.* $\frac{1}{|A|} \sum_{i \in A} |1 - q_i|$.

The average number of nominal conflicts for 10-aircraft RCP instances is 7.35 and the average number of potential conflicts is 20.75. These values are increased to 16.75 and 71.05 for 20-aircraft RCP instances. The results in Tables 1 and 2 show the viability of Model 1 and Algorithm 2. For the most of the results, where a solution is computed, a number of aircraft pairs close to the total number of pairs, is separated in a reasonable time. Moreover, results show that using Algorithm 2 improves the computational performance in solving Model 1 by both (i) improving runtime and (ii) finding better solutions, *i.e.* with more separated pairs of aircraft. Specifically, when the set of potential conflicts P is used to index the constraints of Model 1, runtime is significantly reduced for some instances (best results are in bold face) and it is comparable in the other cases. In the case of comparable runtimes, often more conflicts are solved when Algorithm 2 is used, even though global optimality is not proven (COUENNE with a time limit is employed to solve the MINLP). Note that it is not guaranteed that the reduced MINLP provide a better solution than the initial MINLP—with all pairs of aircraft—when the time limit is reached. Indeed, in this scenario, the spatial branch-and-bound algorithm implemented within COUENNE may terminate with a less effective solution when Algorithm 2 is used that without conflict detection. For 10-aircraft RCP instances, where the total number of aircraft pairs is 45, a number of separated pairs between 40 and 45 is obtained. For some 20-aircraft RCP instances, solving Model 1 without any pre-processing is unsuccessful in finding a feasible solution (the objective function value is 0) within the allocated computing time, whereas a feasible solution is always found when potential conflicts are detected. Notice that the pre-processing time is quite stable: approximately 2.8 seconds for 10-aircraft instances and 5 to 6.5 seconds for 20-aircraft instances. Hence, in most cases the runtime of Algorithm 2 is small compared to the solve time for the MINLP models.

| | | | | | | Time | Without CD | | With CD | | | | |
| | | | | | | | Obj. | Time | Obj. | Time | Nb. | Nb. | Avg. |
Instance	Pairs	$|N|$	$\sum C_{1,2,3}^{ij}$	$|P|$	R_P	CD	Max.VC	Max.VC	Max.VC	Max.VC	Acc.	Dec.	Dev.
RCP_10_1	45	10	36	20	44.4	2.9	41	300.8	43	300.7	3	7	3.7
RCP_10_2	45	11	36	23	51.1	2.8	43	300.6	44	300.6	3	7	3.6
RCP_10_3	45	8	36	20	44.4	2.9	41	300.8	43	300.6	6	4	2.4
RCP_10_4	45	7	36	23	51.1	2.9	43	300.8	43	300.1	6	4	3.0
RCP_10_5	45	7	36	19	42.2	2.7	43	300.8	44	7.5	3	7	2.8
RCP_10_6	45	3	36	17	37.8	2.9	44	300.3	44	4.6	1	9	1.8
RCP_10_7	45	8	36	22	48.9	2.7	41	300.6	42	300.6	3	7	2.0
RCP_10_8	45	3	36	19	42.2	2.7	44	300.8	44	117.1	1	9	5.0
RCP_10_9	45	8	36	22	48.9	2.7	42	300.6	43	300.5	3	7	2.7
RCP_10_10	45	8	36	24	53.3	2.7	43	300.5	43	300.8	5	5	3.6
RCP_10_11	45	7	36	22	48.9	2.8	42	300.6	42	300.6	5	5	2.0
RCP_10_12	45	10	36	21	46.7	2.9	40	301.6	42	300.6	4	6	2.9
RCP_10_13	45	5	36	20	44.4	2.7	44	300.9	44	55.5	2	8	3.5
RCP_10_14	45	6	36	17	37.8	2.7	42	300.7	44	16.5	2	8	4.3
RCP_10_15	45	7	36	21	46.7	2.8	42	300.6	44	301.0	2	8	4.4
RCP_10_16	45	7	36	23	51.1	2.7	44	300.6	44	300.8	4	6	2.6
RCP_10_17	45	5	36	21	46.7	2.9	42	300.8	42	300.5	3	7	2.5
RCP_10_18	45	7	36	19	42.2	2.9	45	1.9	45	0.3	3	7	2.3
RCP_10_19	45	10	36	21	46.7	2.9	44	300.5	44	24.9	4	6	2.2
RCP_10_20	45	10	36	21	46.7	2.8	40	300.7	42	300.6	2	8	2.6

Table 1: Results of the solution of Model 1 with and without Conflict Detection (CD) on RCP instances with 10 aircraft and a radius of 100NM. CD is performed using Algorithm 2. The time limit of Couenne is 300 seconds, all time measures are in seconds.

						Time	Without CD		With CD				
							Obj.	Time	Obj.	Time	Nb.	Nb.	Avg.
Instance	Pairs	$\lvert N\rvert$	$\sum C_{1,2,3}^{ij}$	$\lvert P\rvert$	R_P	CD	Max.VC	Max.VC	Max.VC	Max.VC	Acc.	Dec.	Dev.
RCP_20_1	190	11	172	74	38.9	5.4	0	300.7	187	300.8	13	7	2.5
RCP_20_2	190	15	172	63	33.2	5.3	188	300.3	189	300.5	5	15	2.1
RCP_20_3	190	15	172	68	35.8	4.7	187	300.2	183	300.7	11	9	2.7
RCP_20_4	190	19	172	72	37.9	5.8	189	300.7	186	300.7	12	8	2.3
RCP_20_5	190	19	172	81	42.6	6.2	188	300.8	176	301.0	11	9	1.7
RCP_20_6	190	13	172	67	35.3	5.6	188	301.5	185	300.8	12	8	3.0
RCP_20_7	190	12	172	69	36.3	5.2	187	301.3	184	300.5	11	9	2.3
RCP_20_8	190	15	172	60	31.6	5.7	177	300.5	186	300.6	13	7	3.0
RCP_20_9	190	18	172	72	37.9	5.9	188	300.0	183	301.0	9	11	2.6
RCP_20_10	190	10	172	62	32.6	5.3	190	28.5	190	1.4	6	14	2.2
RCP_20_11	190	21	172	73	38.4	6.4	0	300.2	184	300.6	5	15	2.6
RCP_20_12	190	17	172	78	41.1	5.1	0	300.4	183	300.6	9	11	2.1
RCP_20_13	190	16	172	71	37.4	4.9	187	300.2	189	300.4	3	17	2.5
RCP_20_14	190	17	172	80	42.1	5.4	183	300.2	176	300.6	9	11	1.7
RCP_20_15	190	14	172	68	35.8	5.4	179	301.0	190	4.2	9	11	2.1
RCP_20_16	190	23	172	80	42.1	5.5	0	300.1	176	300.5	17	3	2.0
RCP_20_17	190	22	172	72	37.9	5.4	187	300.2	183	300.9	12	8	2.7
RCP_20_18	190	23	172	70	36.8	5.7	0	300.2	183	300.9	10	10	2.6
RCP_20_19	190	20	172	77	40.5	5.6	0	301.3	181	300.5	11	9	2.5
RCP_20_20	190	15	172	64	33.7	5.6	188	301.2	187	301.0	8	12	2.4

Table 2: Results of the solution of Model 1 with and without Conflict Detection (CD) on RCP instances with 20 aircraft and radius of 200NM. CD is performed using Algorithm 2. The time limit of Couenne is 300 seconds, all time measures are in seconds.

The average number of nominal conflicts for 10-aircraft RTP instances is 4.35 and the average number of potential conflicts is 13.1. These values are increased to 17.7 and 57.15 for 20-aircraft RTP instances. Tables 3 and 4 summarize the results obtained solving Model 2 with and without pre-processing data using Algorithm 2 on 20 RTP instances with 10 and 20 aircraft, respectively. As in the previous case studies, for each instance, Model 2 is solved by indexing the constraints on all pairs of aircraft (columns *without CD*) and by indexing the constraints on the set P of potential conflicts, detected by running Algorithm 2 (columns *with CD*). The headings are identical to the ones of Tables 1 and 2 with the addition of the fourth left-most column $|P_{NS}|$ which gives the cardinality of the set of initially non-separated pairs of aircraft; these pairs will remain in conflict regardless of the decisions taken. Therefore the maximum number of separable pairs (by any separation maneuver) is $45 - |P_{NS}|$ for 10-aircraft instances and $190 - |P_{NS}|$ for 20-aircraft instances.

The tests conducted on the RTP instances highlight the versatility of Model 2 which is able to solve several instances to optimality within the allocated computing time. In 10-aircraft instances, the optimal solution is systematically found within the allocated computing time, with or without conflict detection. While pre-processing the set of potential conflicts helps in reducing the solve time, the improvement is generally marginal. In contrast, for 20-aircraft instances, the initial MINLP struggles to find feasible solutions (instances with an objective value of 0) whereas the reduced MINLP systematically finds feasible—and sometimes optimal—solutions within the allocated solve time.

In all the tests conducted, all aircraft are systematically accelerated or decelerated. This is due to the models used which do not optimize the amount of resources required to separate aircraft. The average deviation is always comprised between 1% and 6% with most cases between 2% and 4%. While in most instances the number of aircraft accelerated and decelerated is relatively balanced, it can happen that a majority of aircraft is accelerated or decelerated, in which case more speed control resources are used.

Instance	Pairs	$|N|$	$|P_{NS}|$	$\sum C_{1,2,3}^{ij}$	$|P|$	R_P	Time CD	Without CD		With CD				
								Obj. Max.VC+	Time Max.VC+	Obj. Max.VC+	Time Max.VC+	Nb. Acc.	Nb. Dec.	Avg. Dev.
RTP_10_1	45	4	1	38	10	22.2	2.7	43	1.4	43	0.3	3	7	2.8
RTP_10_2	45	5	0	39	13	28.9	0.6	43	2.7	43	4.9	4	6	4.2
RTP_10_3	45	4	0	29	18	40.0	2.4	45	1.6	45	0.1	2	8	2.6
RTP_10_4	45	4	0	37	11	24.4	0.6	45	0.2	45	0.2	2	8	2.2
RTP_10_5	45	3	1	36	15	33.3	0.6	44	1.5	44	0.2	1	9	2.0
RTP_10_6	45	2	0	39	11	24.4	0.7	45	0.2	45	0.1	3	7	2.0
RTP_10_7	45	6	1	39	11	24.4	0.6	43	2.8	43	0.7	3	7	2.6
RTP_10_8	45	5	1	36	14	31.1	2.6	44	3.1	44	0.2	4	6	2.0
RTP_10_9	45	4	0	39	12	26.7	0.6	44	1.7	44	0.2	2	8	2.5
RTP_10_10	45	6	0	32	18	40.0	0.5	41	3.3	41	0.1	2	8	1.7
RTP_10_11	45	4	1	38	12	26.7	2.8	44	68.1	44	3.0	2	8	4.2
RTP_10_12	45	4	0	40	9	20.0	0.9	44	1.5	44	0.6	3	7	3.3
RTP_10_13	45	5	1	37	13	28.9	2.7	42	5.4	42	3.7	1	9	1.5
RTP_10_14	45	5	0	35	16	35.6	2.8	41	1.5	41	0.5	1	9	1.8
RTP_10_15	45	4	1	37	13	28.9	2.8	43	42.5	43	0.6	3	7	2.7
RTP_10_16	45	5	0	34	15	33.3	0.7	43	1.1	43	0.5	3	7	2.6
RTP_10_17	45	6	0	38	14	31.1	2.8	43	8.3	43	0.7	4	6	3.0
RTP_10_18	45	4	0	34	15	33.3	0.7	44	2.9	44	0.8	3	7	2.3
RTP_10_19	45	6	0	38	13	28.9	0.8	41	1.5	41	0.5	1	9	1.4
RTP_10_20	45	1	0	37	9	20.0	2.8	45	0.3	45	0.1	2	8	3.1

Table 3: Results of the solution of Model 2 with and without Conflict Detection (CD) on RTP instances with 10 aircraft. CD is performed using Algorithm 2. The time limit of Couenne is 300 seconds, all time measures are in seconds.

								Without CD		With CD				
Instance	Pairs	$\|N\|$	$\|P_{NS}\|$	$\sum C^{ij}_{1,2,3}$	$\|P\|$	R_P	Time CD	Obj. Max.VC+	Time Max.VC+	Obj. Max.VC+	Time Max.VC+	Nb. Acc.	Nb. Dec.	Avg. Dev.
RTP_20_1	190	17	0	158	54	28.4	4.9	183	301.6	185	272.3	9	11	3.9
RTP_20_2	190	10	1	159	49	25.8	5.3	185	301.6	186	9.8	5	15	2.1
RTP_20_3	190	18	2	155	70	36.8	4.6	0	301.1	120	300.6	8	12	4.5
RTP_20_4	190	20	1	154	64	33.7	7.0	0	300.9	179	300.2	7	13	3.5
RTP_20_5	190	19	2	157	55	28.9	5.3	0	301.1	135	300.1	2	18	5.7
RTP_20_6	190	13	2	155	61	32.1	5.8	0	300.1	186	39.9	6	14	2.4
RTP_20_7	190	20	3	156	54	28.4	5.8	182	302.4	181	300.3	8	12	3.1
RTP_20_8	190	16	0	160	53	27.9	5.6	0	302.0	185	300.1	11	9	3.5
RTP_20_9	190	16	0	154	54	28.4	5.6	0	301.2	183	300.1	9	11	3.2
RTP_20_10	190	21	0	162	55	28.9	6.0	0	301.4	189	12.2	6	14	2.0
RTP_20_11	190	18	1	154	57	30.0	6.1	0	301.6	184	25.3	5	15	2.5
RTP_20_12	190	23	0	154	66	34.7	5.2	0	300.8	176	300.3	10	10	3.1
RTP_20_13	190	22	2	153	65	34.2	5.6	0	300.9	125	300.1	4	16	5.4
RTP_20_14	190	14	2	153	52	27.4	5.4	0	300.3	185	157.9	6	14	2.1
RTP_20_15	190	19	2	153	59	31.1	5.2	0	301.6	187	58.4	6	14	2.9
RTP_20_16	190	21	1	155	55	28.9	5.5	0	301.7	183	27.1	5	15	2.0
RTP_20_17	190	18	2	151	64	33.7	5.2	0	302.0	126	300.1	5	15	5.2
RTP_20_18	190	22	2	146	68	35.8	5.0	0	302.0	180	300.1	8	12	3.3
RTP_20_19	190	10	3	159	43	22.6	5.3	186	303.6	186	6.4	8	12	3.8
RTP_20_20	190	17	1	168	45	23.7	5.3	0	300.9	181	300.3	4	16	3.5

Table 4: Results of the solution of Model 2 with and without Conflict Detection (CD) on RTP instances with 20 aircraft. CD is performed using Algorithm 2. The time limit of Couenne is 300 seconds, all time measures are in seconds.

5.3 Results: Finding the largest conflict-free aircraft set

In this section, we discuss the results of numerical experiments on the problem of finding the largest conflict-free aircraft set. These results are obtained running COUENNE on Models 3 and 4, and on the other side solving the same problems, Max.CS and Max.CS+, using Algorithm 3, within which COUENNE is run on Models 1 and 2. The results are presented in Tables 5, 6, 7 and 8. The headings are as follows: the five (resp. six) left-most columns are identical to the ones in Tables 1 and 2 (resp. Tables 3 and 4), the next two columns give the objective value and the runtime after solving Model 3 (resp. 4) and the five next columns provide some detail on the implementation of Algorithm 3: $|A_C|$ is the cardinality of the obtained largest conflict-free aircraft set, *Nb. It.* is the total number of iterations in the **while** loop, *Nb. Add.* is the number of times all current conflicts are solved during the iterative process (**if** $z^\star = |C|$ is true) and *Nb. Rem.* is the number of times at least one conflict is not solved (**if** $z^\star = |C|$ is false); finally the next column gives the total time spent solving Model 1 (resp. 2) within the iterative process. The three right-most columns give the number of accelerated and decelerated aircraft and the average deviation from aircraft nominal speeds. A time limit of 300 seconds is imposed for each direct solve of Models 3 and 4 and a time limit of 60 seconds is imposed for each solve of Models 1 and 2 within Algorithm 3. For both solution methods, Algorithm 2, whose efficiency has been proven in the above numerical experiments, is used to pre-process the set of potential conflicts.

From Tables 5 and 6, we observe a drastic reduction of the computing time on most of the instances when using Algorithm 3 with respect to that of the direct solution of Max.CS by COUENNE, as expected for a greedy algorithm. As for the quality of the solution, we report that the size of the final conflict-free set is always very close to its maximum possible value n, the largest number of removed aircraft being 3 for the tested instances. This is very encouraging to obtain an air sector where only a very small number of remaining conflicting aircraft need to be handled by air traffic services. Further, comparing $|A_C|$ to the objective function value of Max.CS, it appears that Algorithm 3 is able to find comparable or even better solutions (recall that the solution process of Max.CS through COUENNE is stopped when a time limit is reached). On 20-aircraft instances, the direct solution of Max.CS is only able to prove optimality on three instances including two

Instance	$|N|$	$|P|$	R_P	Time CD	Obj. Max.CS	Time Max.CS	$|A_C|$	Nb. It.	Nb. Add.	Nb. Rem.	Total Time Max.VC	Nb. Acc.	Nb. Dec.	Avg. Dev.
RCP_10_1	10	20	44.4	2.7	8	300.1	8	3	1	2	39.1	3	7	3.4
RCP_10_2	11	23	51.1	2.8	9	300.8	9	2	1	1	69.0	3	7	3.0
RCP_10_3	8	20	44.4	2.7	8	300.4	8	3	1	2	52.0	4	6	2.7
RCP_10_4	7	23	51.1	2.6	8	300.5	8	3	1	2	4.2	3	7	2.6
RCP_10_5	7	19	42.2	2.6	9	35.8	9	2	1	1	2.5	4	6	3.0
RCP_10_6	3	17	37.8	2.6	9	19.5	9	2	1	1	0.4	4	6	2.7
RCP_10_7	8	22	48.9	2.6	8	300.1	8	3	1	2	13.8	6	4	3.2
RCP_10_8	3	19	42.2	2.6	9	300.1	9	2	1	1	1.2	4	6	2.8
RCP_10_9	8	22	48.9	2.6	8	300.3	8	3	1	2	18.0	3	7	2.5
RCP_10_10	8	24	53.3	2.6	8	300.6	8	3	1	2	22.9	3	7	3.8
RCP_10_11	7	22	48.9	2.6	8	300.2	8	3	1	2	9.6	3	7	2.0
RCP_10_12	10	21	46.7	2.5	7	300.4	7	4	1	3	87.5	3	7	2.8
RCP_10_13	5	20	44.4	2.5	9	300.1	9	2	1	1	1.7	5	5	1.7
RCP_10_14	6	17	37.8	2.5	9	6.0	9	2	1	1	3.0	4	6	4.1
RCP_10_15	7	21	46.7	0.5	9	300.0	9	2	1	1	64.5	3	7	3.8
RCP_10_16	7	23	51.1	2.5	9	300.2	9	2	1	1	81.2	7	3	2.6
RCP_10_17	5	21	46.7	2.5	7	300.4	7	4	1	3	4.6	1	9	2.6
RCP_10_18	7	19	42.2	2.6	10	0.4	10	1	1	0	0.2	4	6	2.9
RCP_10_19	10	21	46.7	0.6	9	95.3	9	2	1	1	66.8	4	6	2.3
RCP_10_20	10	21	46.7	2.6	8	300.2	8	3	1	2	64.5	4	6	2.9

Table 5: Results of Algorithm 3 on RCP problems with 10 aircraft and a radius of 100 NM. The time limit of CouENNE is 300 seconds for the direct solution of Model 3 and 60 seconds per solve within Algorithm 3.

for which the entire set of flights is separable. In contrast, Algorithm 3 systematically finds feasible and competitive solutions with a maximum of 4 aircraft removed from the initial set.

The tests conducted to find the largest conflict-free set of aircraft on RTP instances exhibits similar trends. The 10-aircraft instances are efficiently solved to optimality using Model 4 and Algorithm 3 performs quite well except for instance RTP_10_6 in which the largest set found by the heuristic consists of 7 aircraft instead 10. The 20-aircraft instances are more challenging for Algorithm 3 with 5 to 12 aircraft being removed in each test, thus showing that the greedy approach can occasionally be myopic in terms of selecting which aircraft to remove from the conflict-free set.

6   Conclusions

In this paper, we have presented aircraft conflict resolution models which rely exclusively on speed control to separate aircraft. The proposed models are formulated using MINLP and are solved with a global optimization engine

Instance	$|N|$	$|P|$	R_P	Time CD	Obj. Max.CS	Time Max.CS	$|A_C|$	Nb. It.	Nb. Add.	Nb. Rem.	Total Time Max.VC	Nb. Acc.	Nb. Dec.	Avg. Dev.
RCP_20_1	11	74	38.9	5.5	16	300.3	19	2	1	1	67.2	7	13	2.8
RCP_20_2	15	63	33.2	5.1	19	300.8	19	2	1	1	62.5	5	15	2.2
RCP_20_3	15	68	35.8	5.1	15	300.4	19	2	1	1	84.1	8	12	2.3
RCP_20_4	19	72	37.9	4.9	20	29.0	20	1	1	0	1.1	7	13	2.5
RCP_20_5	19	81	42.6	4.9	13	300.6	18	3	1	2	181.0	9	11	2.3
RCP_20_6	13	67	35.3	4.8	16	300.4	20	1	1	0	2.2	5	15	2.8
RCP_20_7	12	69	36.3	4.9	0	300.4	19	2	1	1	120.5	9	11	2.9
RCP_20_8	15	60	31.6	4.9	18	300.5	19	2	1	1	63.6	7	13	2.7
RCP_20_9	18	72	37.9	4.6	16	300.3	18	3	1	2	181.1	4	16	3.3
RCP_20_10	10	62	32.6	4.7	20	6.1	20	1	1	0	2.1	4	16	2.0
RCP_20_11	21	73	38.4	4.7	15	300.5	17	4	1	3	244.7	6	14	2.6
RCP_20_12	17	78	41.1	4.7	13	300.7	18	3	1	2	181.7	9	11	2.4
RCP_20_13	16	71	37.4	4.7	14	300.4	19	2	1	1	88.1	11	9	2.8
RCP_20_14	17	80	42.1	4.7	12	300.4	18	3	1	2	129.5	10	10	2.1
RCP_20_15	14	68	35.8	5.0	20	32.8	19	2	1	1	131.9	7	13	2.1
RCP_20_16	23	80	42.1	5.0	14	300.8	16	5	1	4	243.4	4	16	3.2
RCP_20_17	22	72	37.9	5.0	16	300.4	17	4	1	3	181.6	8	12	2.3
RCP_20_18	23	70	36.8	4.9	15	300.6	18	3	1	2	240.6	7	13	2.7
RCP_20_19	20	77	40.5	5.0	0	300.3	19	2	1	1	121.0	9	11	2.6
RCP_20_20	15	64	33.7	4.7	16	300.4	19	2	1	1	88.7	5	15	2.7

Table 6: Results of Algorithm 3 on RCP with 20 aircraft and a radius of 200 NM. The time limit of Couenne is 300 seconds for the direct solution of Model 3 and 60 seconds per solve within Algorithm 3.

Instance	$|N|$	$|P_{NS}|$	$|P|$	R_P	Time CD	Obj. Max.CS+	Time Max.CS+	$|A_C|$	Nb. It.	Nb. Add.	Nb. Rem.	Total Time Max.VC+	Nb. Acc.	Nb. Dec.	Avg. Dev.
RTP_10_1	4	1	10	22.2	2.7	7	1.2	7	3	0	3	0.8	2	8	5.4
RTP_10_2	5	0	13	28.9	0.6	7	0.5	6	5	1	4	0.9	3	7	2.5
RTP_10_3	4	0	18	40.0	0.4	8	0.4	7	4	1	3	0.5	2	8	5.4
RTP_10_4	4	0	11	24.4	0.6	10	0.3	10	0	0	0	0.9	6	4	4.0
RTP_10_5	3	1	15	33.3	0.5	9	0.3	8	3	1	2	1.1	3	7	3.2
RTP_10_6	2	0	11	24.4	0.6	10	0.2	7	4	1	3	0.6	3	7	4.1
RTP_10_7	6	1	11	24.4	0.6	8	0.5	8	3	1	2	1.6	3	7	4.6
RTP_10_8	5	1	14	31.1	0.6	7	0.3	7	4	1	3	1.1	3	7	5.1
RTP_10_9	4	0	12	26.7	0.6	8	0.6	7	4	1	3	0.5	0	10	6.0
RTP_10_10	6	0	18	40.0	2.5	6	3.3	6	4	0	4	0.4	0	10	6.0
RTP_10_11	4	1	12	26.7	0.6	9	0.5	9	2	1	1	0.9	4	6	3.6
RTP_10_12	4	0	9	20.0	0.6	8	1.6	8	3	1	2	1.0	4	6	4.6
RTP_10_13	5	1	13	28.9	0.5	7	2.2	6	5	1	4	1.8	2	8	5.3
RTP_10_14	5	0	16	35.6	0.5	7	2.0	6	5	1	4	1.2	5	5	4.0
RTP_10_15	4	1	13	28.9	2.5	8	0.5	7	4	1	3	0.6	5	5	4.5
RTP_10_16	5	0	15	33.3	0.5	8	0.9	8	3	1	2	1.1	4	6	4.8
RTP_10_17	6	0	14	31.1	0.5	8	1.0	8	3	1	2	1.3	2	8	5.0
RTP_10_18	4	0	15	33.3	0.5	9	33.1	8	2	0	2	1.0	3	7	4.5
RTP_10_19	6	0	13	28.9	2.6	6	1.1	6	5	1	4	1.4	3	7	4.0
RTP_10_20	1	0	9	20.0	0.5	10	0.2	10	1	1	0	0.9	5	5	4.3

Table 7: Results of Algorithm 3 on RTP problems with 10 aircraft. The time limit of Couenne is 300 seconds for the direct solution of Model 4 and 60 seconds per solve within Algorithm 3.

Instance	$|N|$	$|P_{NS}|$	$|P|$	R_P	Time CD	Obj. Max.CS+	Time Max.CS+	$|A_C|$	Nb. It.	Nb. Add.	Nb. Rem.	Total Time Max.VC+	Nb. Acc.	Nb. Dec.	Avg. Dev.
RTP_20_1	17	0	54	28.4	4.6	13	300.2	11	10	1	9	37.4	9	11	4.0
RTP_20_2	10	1	49	25.8	4.6	17	10.6	15	6	1	5	15.9	7	13	4.0
RTP_20_3	18	2	70	36.8	4.4	0	300.2	11	10	1	9	12.3	2	18	2.9
RTP_20_4	20	1	64	33.7	4.6	12	300.3	10	11	1	10	14.7	4	16	3.0
RTP_20_5	19	2	55	28.9	4.4	13	300.1	11	10	1	9	59.1	5	15	4.1
RTP_20_6	13	2	61	32.1	4.4	15	300.2	13	8	1	7	13.8	5	15	3.9
RTP_20_7	20	3	54	28.4	4.4	13	300.3	11	10	1	9	67.7	6	14	3.1
RTP_20_8	16	0	53	27.9	4.6	13	300.5	12	9	1	8	25.9	2	18	4.9
RTP_20_9	16	0	54	28.4	4.7	14	300.2	13	8	1	7	31.5	6	14	3.2
RTP_20_10	21	0	55	28.9	4.5	15	300.5	13	8	1	7	10.0	3	17	3.6
RTP_20_11	18	1	57	30.0	4.5	14	300.4	11	10	1	9	16.6	5	15	2.6
RTP_20_12	23	0	66	34.7	4.3	0	300.4	12	9	1	8	137.6	9	11	3.6
RTP_20_13	22	2	65	34.2	4.4	0	300.6	10	11	1	10	12.6	1	19	4.0
RTP_20_14	14	2	52	27.4	4.4	15	300.5	13	8	1	7	25.1	4	16	4.7
RTP_20_15	19	2	59	31.1	4.3	15	300.3	12	8	0	8	21.4	6	14	3.4
RTP_20_16	21	1	55	28.9	4.4	13	300.5	12	9	1	8	16.8	3	17	4.0
RTP_20_17	18	2	64	33.7	4.5	13	300.1	13	8	1	7	15.3	4	16	4.0
RTP_20_18	22	2	68	35.8	4.4	12	300.3	8	13	1	12	15.2	3	17	3.1
RTP_20_19	10	3	43	22.6	4.4	15	300.6	14	7	1	6	15.2	5	15	4.4
RTP_20_20	17	1	45	23.7	4.5	14	7.4	12	9	1	8	25.1	6	14	3.0

Table 8: Results of Algorithm 3 on RTP problems with 20 aircraft. The time limit of Couenne is 300 seconds for the direct solution of Model 4 and 60 seconds per solve within Algorithm 3.

(Couenne). In contrast to previous research on conflict resolution, the presented approach does not assume any form of discretization (space or time) to solve the aircraft collision avoidance problem, hence providing exact flight trajectory adjustments. Due to the nature of the problem at hand, two formulations are proposed: (i) maximize the number of conflicts resolved and (ii) find the largest conflict-free aircraft set. Both formulations take into account the fact that not all aircraft may be separated by speed control only and thus highlight the necessity to count separated pairs of aircraft or conflict-free aircraft. The initial formulations are indexed on all pairs of aircraft, then a pre-processing step is proposed to reduce the number of necessary separation constraints by identifying potential air conflicts based on the available aircraft speed regulation range. This procedure works by considering each pair of aircraft independently and can be executed efficiently by solving a concave program, for each pair of aircraft satisfying geometric conditions only depending on their initial position and speed. Finally, a heuristic algorithm is proposed to find the largest conflict-free aircraft set based on an aircraft-removal strategy which selects aircraft penalizing the most other aircraft trajectories.

The proposed models and algorithms are tested on air traffic control benchmark instances and the results obtained highlight the benefits of using a pre-processing step to reduce the size of the MINLP formulations as well as the versatility and the efficiency of the proposed models. The heuristic algorithm is shown to significantly reduce computing time while finding competitive—if not optimal—solutions. Based on the reported runtimes, it is clear that MINLP models occasionally struggle to prove optimality although they may be efficient in finding feasible solutions with some pre-processing that reduces the constraint set size. This suggests that fast heuristic algorithms could be designed on the premise that a relatively small computing time limit is sufficient to find a feasible solution.

Future works will be focused on improving runtime and solution quality by developing further efficient solution approaches. Hybrid conflict resolution maneuvers, combining the proposed speed control-based models with heading angle deviation and/or altitude change, will be investigated to provide a more comprehensive conflict resolution framework. Finally, the robustness of the proposed solution approaches with regards to the uncertainty on aircraft trajectory prediction models will be quantified and relevant models developed.

Acknowledgment

References

[1] EUROCONTROL, Eurocontrol long-term forecast: IFR Flight Movements 2010–2030, Tech. rep., Eurocontrol–Air Traffic Statistics and Forecast (2010).

[2] G. Granger, N. Durand, J.-M. Alliot, Optimal resolution of en-route conflicts, in: 1st USA/Europe Air Traffic Management Research and Development Seminar, Saclay, France, 1997.

[3] J. Kuchar, L. Yang, A review of conflict detection and resolution modeling methods, Intelligent Transportation Systems, IEEE Transactions on 1 (4) (2000) 179–189. doi:10.1109/6979.898217.

[4] L. Pallottino, E. M. Feron, A. Bicchi, Conflict resolution problems for air traffic management systems solved with mixed integer programming, Intelligent Transportation Systems, IEEE Transactions on 3 (1) (2002) 3–11.

[5] A. Richards, J. How, Aircraft trajectory planning with collision avoidance using mixed integer linear programming, in: Proceedings of the American Control Conference, Vol. 3, 2002, pp. 1936–1941. doi:10.1109/ACC.2002.1023918.

[6] A. Alonso-Ayuso, L. F. Escudero, F. J. Martín-Campo, A mixed 0-1 nonlinear optimization model and algorithmic approach for the collision avoidance in ATM: velocity changes through a time horizon, Computers & Operations Research 39 (12) (2012) 3136–3146. doi:10.1016/j.cor.2012.03.015.

[7] A. Alonso-Ayuso, L. F. Escudero, P. Olaso, C. Pizarro, Conflict avoidance: 0-1 linear models for conflict detection & resolution, TOP 21 (3) (2013) 485–504. doi:10.1007/s11750-011-0224-6.

[8] A. Alonso-Ayuso, L. F. Escudero, F. J. Martín-Campo, On modeling the air traffic control coordination in the collision avoidance problem by mixed integer linear optimization, Ann. Oper. Res. 222 (2014) 89–105. doi:10.1007/s10479-013-1347-y.

[9] J. Omer, A space-discretized mixed-integer linear model for air-conflict resolution with speed and heading maneuvers, Computers & Operations Research 58 (2015) 75–86.

[10] S. Cafieri, N. Durand, Aircraft deconfliction with speed regulation: new models from mixed-integer optimization, Journal of Global Optimization 58 (4) (2014) 613–629. doi:10.1007/s10898-013-0070-1.

[11] D. Rey, C. Rapine, R. Fondacci, N.-E. El Faouzi, Minimization of potential air conflicts through speed regulation, Transportation Research Record: Journal of the Transportation Research Board (2012) 59–67.

[12] D. Rey, C. Rapine, V. V. Dixit, S. T. Waller, Equity-oriented aircraft collision avoidance model, IEEE Transactions on Intelligent Transportation Systems 16 (1) (2015) 172–183.

[13] D. Rey, C. Rapine, R. Fondacci, N.-E. El Faouzi, Subliminal speed control in air traffic management: Optimization and simulation, Transportation Science 50 (1) (2015) 240–262. doi:dx.doi.org/10.1287/trsc.2015.0602.

[14] S. Cafieri, MINLP in air traffic management: Aircraft conflict avoidance, in: T. Terlaky, M. Anjos, S. Ahmed (Eds.), Advances and Trends in Optimization with Engineering Applications, MOS-SIAM Series on Optimization, SIAM, Philadelphia, 2016, in press.

[15] S. Cafieri, Maximizing the number of solved aircraft conflicts through velocity regulation, in: MAGO 2014, 12th Global Optimization Workshop, Málaga, Spain, 2014, pp. pp 1–4.
URL https://hal-enac.archives-ouvertes.fr/hal-01018051

[16] A. Warren, Trajectory prediction concepts for next generation air traffic management, in: 3th USA/Europe Air Traffic Management Research and Development Seminar, Napoli, Italia, 2000.

[17] F. Drogoul, P. Averty, R. Weber, Erasmus strategic deconfliction to benefit sesar, in: 8th USA/Europe Air Traffic Management Research and Development Seminar, Napa, USA, 2009.

[18] M. Brochard, Erasmus - en route air traffic soft management ultimate system, Tech. rep., Eurocontrol Experimental Centre (2006).

[19] S. Athènes, P. Averty, S. Puechmorel, D. Delahaye, C. Collet, ATC complexity and controller workload: Trying to bridge the gap, in: Proceedings of the International Conference on HCI in Aeronautics, 2002, pp. 56–60.

[20] R. Fortet, Applications de l'algèbre de boole en recherche opérationelle, Revue Française de Recherche Opérationelle 4 (1960) 17–26.

[21] R. Fourer, D. M. Gay, B. W. Kernighan, AMPL: A Modeling Language for Mathematical Programming, 2nd Edition, Brooks/Cole, 2002.

[22] P. Belotti, J. Lee, L. Liberti, F. Margot, A. Wachter, Branching and bounds tighteningtechniques for non-convex minlp, Optimization Methods Software 24 (4-5) (2009) 597–634. doi:10.1080/10556780903087124. URL `http://dx.doi.org/10.1080/10556780903087124`

[23] A. Wächter, L. T. Biegler, On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming, Mathematical programming 106 (1) (2006) 25–57.

[24] C. Vanaret, D. Gianazza, N. Durand, J.-B. Gotteland, Benchmarking conflict resolution algorithms, in: 5th International Conference on Research in Air Transportation, ICRAT, Berkeley, USA, 2012, pp. 113–114.

[25] A. Alonso-Ayuso, L. Escudero, F. Martín-Campo, Collision avoidance in air traffic management: A mixed-integer linear optimization approach, Intelligent Transportation Systems, IEEE Transactions on 12 (1) (2011) 47–57. doi:10.1109/TITS.2010.2061971.