Journal of Global Optimization
https://doi.org/10.1007/s10898-024-01393-1

Aircraft conflict resolution with trajectory recovery using mixed-integer programming

Fernando Dias[1] · David Rey[2]

Received: 1 March 2022 / Accepted: 18 February 2024

Abstract
To guarantee the safety of flight operations, decision-support systems for air traffic control must be able to improve the usage of airspace capacity and handle increasing demand. This study addresses the aircraft conflict avoidance and trajectory recovery problem. The problem of finding the least deviation conflict-free aircraft trajectories that guarantee the return to a target waypoint is highly complex due to the nature of the nonlinear trajectories that are sought. We present a two-stage iterative algorithm that first solves initial conflicts by manipulating their speed and heading control and then identifying each aircraft's optimal time to recover its trajectory towards their nominal one. We extend existing mixed-integer programming formulations by modelling speed and heading control as continuous variables while recovery time is treated as a discrete variable. We develop a novel iterative approach which shows that the trajectory recovery costs can be anticipated by inducing avoidance trajectories with higher deviation, therefore obtaining earlier recovery time within a few iterations. Numerical results on benchmark conflict resolution problems show that this approach can solve instances with up to 30 aircraft within 10 min.

Keywords Air traffic control · Conflict resolution · Trajectory recovery · Mixed integer programming

1 Introduction

Conflict detection and resolution (CDR) has a significant impact on the workload of air traffic controllers [1]. With increased airspace usage, controllers are expected to simultaneously supervise an increased number of aircraft, thus reducing their cognitive capacity for potential conflict detection and increasing their stress levels. This calls for the development of CDR algorithms and their integration as decision-support tools for air traffic controllers.

✉ Fernando Dias
fernando.dias@aalto.fi

David Rey
david.rey@skema.edu

[1] Department of Mathematics and System Analysis, Aalto University, 02150 Espoo, Finland

[2] SKEMA Business School, Université Côte d'Azur, Sophia Antipolis, France

Published online: 17 September 2024

Airspace usage has seen an increase in demand throughout the last decades (except during specific periods such as post-09/11 attacks and the COVID-19 pandemic). Aligned with this trend, the limited airspace has put the current air traffic management (ATM) system under intense pressure. This is reflected in the increasing amount of control necessary to guarantee safety, which is a crucial aspect of air traffic control (ATC). With the advance of remote-controlled aerial systems and urban air mobility, denser and more congested traffic configurations are also expected. In such a scenario, aircraft safety might be impaired. Besides safety issues, conflict detection and resolution (CDR) has a significant impact on the workload of air traffic controllers [1]. With increased airspace usage, controllers are expected to simultaneously supervise an increased number of aircraft, thus reducing their cognitive capacity for potential conflict detection and increasing their stress levels. This calls for the development of CDR algorithms and their integration as decision-support tools for air traffic controllers. Nevertheless, state-of-the-art methods for aircraft traffic control are reaching their limits and new approaches, including more automation, have recently received significant attention in the field [2, 3]. In air traffic control, two or more aircraft are said to be in conflict if their trajectories violate horizontal and vertical separation norms. Typically, the horizontal separation norm is 5NM (nautical mile—1NM is 1852 m), and the vertical separation norm is 1000 ft [4].

The aircraft conflict avoidance and resolution problem can be formulated as an optimization problem in which the goal is to find conflict-free trajectories while minimizing a cost function, e.g. the deviation to the original flight path. Then, trajectory recovery is related to additional maneuvers required to return the aircraft to their original flight path. Many strategies have been proposed to address trajectory recovery based on the type of maneuvers issued to aircraft: speed, heading or altitude control. These strategies can be applied separately or in combination. Conflict resolution using global optimization has recently received growing attention due to its ability to provide optimal solutions considering all aircraft present within an airspace region. Following, we review the literature on conflict avoidance in Sect. 1.1 before focusing on efforts to develop approaches for trajectory recovery in Sect. 1.2. We outline our contributions in Sect. 1.3.

1.1 Conflict avoidance

One of the first global optimization approaches for aircraft conflict resolution was introduced by [5], which proposed two formulations: one focusing on speed control and another focusing on heading control, and both minimize overall flight time. In the proposed mixed-integer programming (MIP) formulation for conflict resolution with speed control, the authors derived linear pairwise aircraft separation constraints based on the geometric construction introduced by [6]. These separation conditions are obtained by projecting an intruder's shadow (the projection of an aircraft trajectory into another trajectory) onto a reference aircraft's trajectory.

In [7], the authors were the first to observe that this geometric construction provided a basis to characterize the set of pairwise conflict-free trajectories via linear half-planes in the relative velocity (speed and heading) plane. The authors introduced a nonconvex formulation for the conflict resolution problem with speed and heading control and proposed a convex relaxation based on semi-definite programming and a heuristic algorithm to find feasible solutions to problems with up to 10 aircraft.

Subsequent approaches proposed speed control and altitude level assignment to minimize fuel consumption by imposing separation at locations where aircraft trajectories intersect [8].

In [9], the authors proposed a two-stage stochastic optimization model accounting for wind uncertainty and using speed control. Multi-objective optimization formulations attempting to perform a trade-off among deconfliction maneuvers (velocity, heading or altitude change) with the total number of maneuvers, building on the work of [5] were proposed by [10, 11]. Subliminal speed control methods, which focus on speed control only for conflict resolution, have also proven to be efficient and with low impact in terms of deviation and fuel consumption. However, they may fail to resolve all conflicts [12–14].

More recently, nonlinear global optimization approaches have received increasing attention in the literature. In [15], the authors proposed a hybrid algorithm that uses the optimal solution of a Mixed-Integer Linear Program (MILP) as the starting point for solving a nonlinear formulation of the same problem. In [13], a Mixed-Integer Nonlinear Programming (MINLP) approach for conflict resolution was proposed. Using only speed control highlights that subliminal speed control alone may not be sufficient to resolve all conflicts in dense traffic scenarios. Using a similar framework, Cafieri and Omheni [16] presented a two-step approach where a maximum number of conflicts are first solved using speed control only, and outstanding conflicts are solved by heading control. In [17], the author proposed a speed regulation strategy for aircraft deconfliction using feasibility pump. In [18], a formulation based on bi-level optimization with multiple follower problems is introduced. Each follower problem represents a two-aircraft separation problem. The authors presented two formulations, one using speed control and another using heading control. Recently, Lehouillier et al. [19] proposed a maneuver-discretized model in which predefined sets of maneuvers are available for aircraft, and a clique-based formulation is proposed to find the optimal combination of conflict-free maneuvers. A cut generation algorithm is proposed to solve the corresponding bi-level optimization problems. More recently, a complex number formulation for speed and heading control without any form of discretization was proposed by [20]. Further, it was extended by [21] in which an exact constraint generation algorithm was proposed. A systematic review of mathematical programming methods in ATC can be found in [3].

1.2 Trajectory recovery

Aircraft conflict resolution with trajectory recovery is the problem of finding conflict-free trajectories that ensure that aircraft recover their initial flight path upon completion of the maneuvers. This problem has received very little attention in the literature due to its challenging nature (i.e. nonlinear and nonconvexity). Mathematical programming approaches that have jointly addressed the avoidance and recovery problems often assume that a predetermined set of alternative maneuvers is available to choose from before optimization or attempt to simplify the formulation by discretizing the domain of trajectory control variables.

In the literature of aircraft conflict resolution, approaches based on mathematical programming often opt for a two-stage model. These stages are called avoidance and recovery. Avoidance corresponds to the maneuvers necessary to avoid conflicts, and recovery corresponds to the maneuvers necessary to restore the aircraft to its original trajectory. Some modelling approaches ignore this two-stage trajectory model structure and instead develop nonlinear trajectory models wherein aircraft deviate from and return to the initial trajectory in a more integrated, possibly smooth, fashion. However, because of the difficulty to model this integrated trajectory design problem using mathematical programming, formulation based on mathematical programming often separate these two stages.

Despite their potential effectiveness, most efforts in conflict resolution have focused on ensuring conflict avoidance. Such efforts do not take into account the challenge associated with modelling trajectory recovery or the costs induced by avoidance maneuvers. This may be critical when conflict resolution is performed using heading control, which may significantly deviate the aircraft from its initial trajectory, thus possibly increasing flight operating costs.

Meta-heuristics such as genetic algorithms are an alternative in order to avoid the challenges of expressing analytical trajectories (due to trigonometric functions). Thus, such meta-heuristics results in the aircraft path being expressed in a discretized manner. In [22], the aircraft path is divided into different segments based on possible conflict points and turning points, thus predefining a set of eligible trajectories. An ant colony algorithm was also proposed by [23], where the authors determined the target point for each aircraft trajectory and discretized its trajectory using a specific timestep. In [24], a different formulation is presented in which time and space are represented as discrete variables. In order to generate a smooth aircraft path (as in [25]) and derive conflict-free solutions, genetic algorithms were implemented. In addition, in [25], the authors proposed a model which uses an analogy with light propagation theory to create conflict-free aircraft trajectories with recovery. In [26], the authors proposed a B-splines model which uses way-points of a given trajectory to design conflict-free trajectories, including their recovery maneuvers. The authors did not explicitly separate avoidance from trajectory recovery in [24–26]. In [27], the author proposed a formulation providing a parallel trajectory recovery (which is an alternative trajectory obtained after performing any recovery maneuver that results in a trajectory which is parallel to initial trajectory of the aircraft) while minimizing fuel consumption and delays. In this model, aircraft are assumed to perform a preventive maneuver before intersection with other trajectories. In [11], the author introduced a two-step approach. The first stage was a non-convex mixed integer nonlinear optimization model for avoidance based on geometric constructions. The second stage solved a set of unconstrained quadratic models to impose trajectory recovery.

This literature review highlights that despite recent improvements in the modelling of aircraft conflict resolution problems, designing scalable global optimization methods remains a challenge, especially if trajectory recovery is considered. Few methods for aircraft trajectory recovery have been proposed, and they typically assume a simplified trajectory design (as expressed beforehand). This can be justified by the difficulties in formulating trajectory recovery (trigonometric and nonlinear functions), especially using mathematical programming.

1.3 Our contributions

As highlighted in our literature review, most state-of-the-art models using mathematical programming focus on conflict avoidance only. This study presents a new two-stage algorithm for aircraft conflict resolution with trajectory recovery. Our approach is based on decomposing the problem into two stages: conflict avoidance and trajectory recovery. In this approach, the speed and heading of aircraft are first optimized to avoid conflicts while minimizing the deviation from their initial trajectories. Then, in the second stage, aircraft trajectories are modified to recover a target position on the aircraft's initial trajectories. These two stages are incorporated into an iterative algorithm, where the recovery cost is used as part of the avoidance model to infer how decisions in the avoidance stage have consequences in the recovery stage and vice-versa. Hence, this algorithm aims to obtain a non-trivial solution where the overall cost of aircraft maneuvers is minimized. The main contributions of this study relative to the literature are:

1. The extension of state-of-the-art mixed-integer formulations for conflict avoidance with aircraft selection variables that model the choice of deviating or not an aircraft from its nominal trajectory,
2. The design of a penalty-based algorithm that iterates the conflict-avoidance and trajectory recovery stages to find non-trivial solutions to the problem at hand, and
3. Extensive numerical experiments on benchmarking problems from the literature that demonstrate the benefits of the proposed approach.

The paper is organized as follows: in Sect. 2, the two-stage algorithm is detailed, starting with the conflict avoidance formulation and then the trajectory recovery. In Sect. 3, the experimental framework, the results and discussions are presented. Section 4 provides the conclusion and future research directions.

2 Conflict resolution with trajectory recovery

In this section, we present a novel conflict resolution approach with trajectory recovery based on decomposing the problem into two stages. The proposed approach aims to solve these two sub-problems repeatedly: it first solves the conflict avoidance stage parameterized by an estimate of the recovery costs, and then it solves the trajectory recovery stage based on the outputs of the first stage.

2.1 Preliminaries

We assume that aircraft current and target positions are known and that aircraft are not in conflict at $t = 0$. This sets the context of the optimization problem of interest: given a set of aircraft with known current and target positions, find the least-deviating conflict-free trajectories for all aircraft, such that aircraft may safely reach their target destination. To address this problem, we propose decomposing the trajectory optimization problem into two stages: (1) conflict avoidance and (2) trajectory recovery. The first stage focuses on controlling aircraft heading and speed to avoid all conflicts and the second stage focuses on calculating the optimal time for aircraft to recover safely towards their target position. We focus on the two-dimensional conflict resolution problem and only consider horizontal aircraft maneuvers for brevity.

The avoidance stage aims to determine the optimal variation in speed and angle for each aircraft to avoid conflicts. In the recovery stage, it is necessary to calculate and identify the maneuvers required to recover the aircraft's initial trajectories. In order to impose airspace safety, it is also necessary to guarantee separation conditions in the recovery stage. In mathematical programming, recovery approaches are challenging for several reasons. Even under the simplifying assumption that aircraft trajectories can be modelled as linear uniform motion over time, using the Euclidean distance to measure their relative distance and evaluate the separation condition yields computational hurdles. This leads to equations using quadratic and trigonometric elements, that are difficult to handle. Thus, such formulations may not scale easily and be used in large scenarios. Second, Euclidean distance creates quadratic terms that are nonlinear and nonconvex concerning decision variables, resulting in formulations that cannot be easily solved. As presented by [5, 10, 20], nonlinear constraints can be further simplified into a set of integer-linear equations with regards to aircraft velocity. However, these formulations only focus on separating aircraft to solve existing conflicts and ignore the process of planning aircraft recovery to a target destination. As shown in Fig. 1,

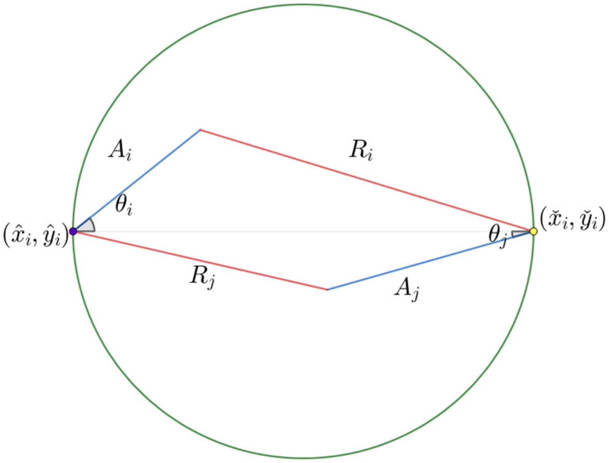


Fig. 1 Example of a two-aircraft conflict resolution problem. If both aircraft followed their nominal trajectory (in grey), aircraft i is in conflict with aircraft j. In order to avoid conflict, conflict avoidance maneuvers have to be taken. For aircraft i, it deviates from its nominal maneuver by following the avoidance trajectory (A_i). After a certain amount of time, the aircraft can safely return towards its target point $(\check{x}_i, \check{y}_i)$ via the trajectory recovery (R_i)

trajectories in blue and in red segments are examples of conflict avoidance and trajectory recovery, respectively. At first, the avoidance (in blue) is performed, followed by another maneuver (in red) to redirect the aircraft towards its target point (in yellow). In this study, we focus on developing a global optimization approach to design piecewise linear conflict-free trajectories (avoidance and recovery) for a set of aircraft to minimize the total deviation with regards to aircraft nominal trajectories.

2.2 Conflict avoidance

Consider a set of aircraft $\mathcal{A}$ sharing the same flight level. For each aircraft $i \in \mathcal{A}$, assuming uniform motion laws apply, its position is: $p_i(t) = [x_i(t) = \widehat{x_i} + q_i \hat{v}_i \cos(\widehat{\theta_i} + \theta_i)t,\ y_i(t) = \widehat{y_i} + q_i \hat{v}_i \sin(\widehat{\theta_i} + \theta_i)t]^\top$ in which v_i is the speed, $\widehat{x_i}$ and $\widehat{y_i}$ are the initial coordinates of i at the time of optimization, $\widehat{\theta_i}$ is its initial heading angle, θ_i is its deviation angle and q_i is the speed deviation. The relative velocity vector of i and j, denoted v_{ij}, can be expressed as:

$$v_{ij} = [v_{ij}^x, v_{ij}^y]^\top = [v_{i,x} - v_{j,x}, v_{i,y} - v_{j,y}]^\top, \tag{1}$$

where:

$$v_{i,x} = q_i \hat{v}_i \sin(\widehat{\theta_i} + \theta_i), \tag{2}$$

$$v_{i,y} = q_i \hat{v}_i \cos(\widehat{\theta_i} + \theta_i). \tag{3}$$

Incorporating these elements into the equation of motion gives:

$$x_i(t) = \widehat{x_i} + v_{i,x} t, \tag{4a}$$

$$y_i(t) = \widehat{y_i} + v_{i,y} t. \tag{4b}$$

The relative position of aircraft i and j at time t can be represented as $p_{ij}(t) = p_i(t) - p_j(t)$. Let d be the horizontal separation norm, typically $d = 5$ NM. Two aircraft $i, j \in \mathcal{A}$ are horizontally separated if and only if: $||p_{ij}(t)|| \geq d$, for all $t \geq 0$.

Let $\mathcal{P}$ be the set by the pairs of aircraft, i.e. $\mathcal{P} = \{i \in \mathcal{A}, j \in \mathcal{A}, i < j\}$. For each pair $(i, j) \in \mathcal{P}$, the relative position vector $p_{ij}(t)$ is:

$$p_{ij}(t) = [x_{ij}(t), y_{ij}(t)]^\top, \tag{5}$$

and the relative velocity vector $v_{ij} = [v^x_{ij}, v^y_{ij}]^\top$, is:

$$v^x_{ij} = q_i \widehat{v}_i \cos(\widehat{\theta}_i + \theta_i) - q_j \widehat{v}_j \cos(\widehat{\theta}_j + \theta_j), \tag{6a}$$

$$v^y_{ij} = q_i \widehat{v}_i \sin(\widehat{\theta}_i + \theta_i) - q_j \widehat{v}_j \sin(\widehat{\theta}_j + \theta_j). \tag{6b}$$

Imposing the separation condition, gives for each pair $(i, j) \in \mathcal{P}$:

$$||p_{ij}(t)|| \geq d \Leftrightarrow \sqrt{(x_i(t) - y_j(t))^2 + (y_i(t) - y_j(t))^2} \geq d, \quad \forall t \geq 0. \tag{7}$$

Let $\widehat{x}_{ij} = \widehat{x}_i - \widehat{x}_j$ and $\widehat{y}_{ij} = \widehat{y}_i - \widehat{y}_j$. Squaring both sides in Eq. (7), we obtain:

$$f_{ij}(t) \equiv ((v^x_{ij})^2 + (v^y_{ij})^2)t^2 + (2v^x_{ij}\widehat{x}_{ij} + 2v^y_{ij}\widehat{y}_{ij})t + \widehat{x}^2_{ij} + \widehat{y}^2_{ij} - d^2 \geq 0. \tag{8}$$

The function $f_{ij}(t)$ is a second-order polynomial in t which is minimal for $f'_{ij}(t) = 0$:

$$f'_{ij}(t) = 0 \Rightarrow t^{\min}_{ij} \equiv -\frac{\widehat{x}_{ij} v^x_{ij} + \widehat{y}_{ij} v^y_{ij}}{(v^x_{ij})^2 + (v^y_{ij})^2}. \tag{9}$$

The time instant $t^{\min}_{ij}$ represents the time of minimal separation of aircraft i and j. As noted in several studies [13, 14, 20], if $t^{\min}_{ij} \leq 0$, then aircraft i and j are diverging and, assuming aircraft are separated at $t = 0$, they are thus separating for any $t \geq 0$. Further, substituting $t^{\min}_{ij}$ in $f_{ij}(t)$ yields:

$$g_{ij}(v^x_{ij}, v^y_{ij}) \equiv f_{ij}(t^{\min}_{ij}) = (v^y_{ij})^2(\widehat{x}^2_{ij} - d^2) + (v^x_{ij})^2(\widehat{y}^2_{ij} - d^2) - (2\widehat{x}_{ij}\widehat{y}_{ij}v^x_{ij}v^y_{ij}). \tag{10}$$

Hence, if $g_{ij}(v^x_{ij}, v^y_{ij}) \geq 0$, then aircraft i and j are separated. Therefore, we obtain the following disjunctive pairwise aircraft separation conditions:

$$||p_{ij}(t)|| \geq d, \forall t \geq 0 \Leftrightarrow g_{ij}(v^x_{ij}, v^y_{ij}) \geq 0 \vee t^{\min}_{ij} \leq 0. \tag{11}$$

The separation condition (11) can be further linearized following the approach described by [20, 21]. By alternatively fixing variables v^x_{ij} and v^y_{ij} and solving the resulting quadratic equations, we can obtain the solution for $g_{ij}(v^x_{ij}, v^y_{ij}) = 0$. By isolating each variable, we obtain the discriminants:

$$\begin{cases} \Delta_{v^x_{ij}} = 4d^2(v^y_{ij})^2(\widehat{x}^2_{ij} + \widehat{y}^2_{ij} - d^2), \\ \Delta_{v^y_{ij}} = 4d^2(v^x_{ij})^2(\widehat{x}^2_{ij} + \widehat{y}^2_{ij} - d^2). \end{cases} \tag{12}$$

Assuming aircraft are separated at $t = 0$, then $\widehat{x}^2_{ij} + \widehat{y}^2_{ij} - d^2 \geq 0$ holds and thus the discriminants are positive, and the roots of equation $g(v^x_{ij}, v^y_{ij}) = 0$ are the lines defined by the system of equations:

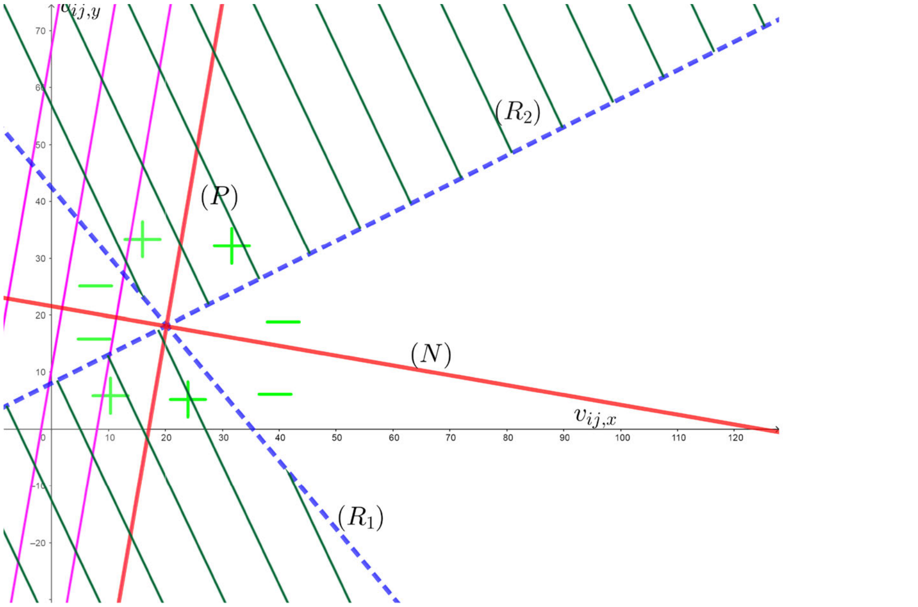


Fig. 2 Illustration of a two-aircraft conflict in the plane $\{(v^x_{ij}, v^y_{ij}) \in \mathbb{R}^2\}$. The red lines represent the lines P and N. The dashed blue lines correspond to the linear equations R_1 and R_2 that are the roots of $g(v^x_{ij}, v^y_{ij}) = 0$. The sign of $g(v^x_{ij}, v^y_{ij})$ is shown by the + and − green symbols. The hashed dark green region represents $g(v^x_{ij}, v^y_{ij}) \geq 0$. The hashed blue half-plane represents diverging trajectories, i.e. $t^{\min}_{ij}(v^x_{ij}, v^y_{ij}) \leq 0$. (Color figure online)

$$(\hat{y}^2_{ij} - d^2)v^x_{ij} - (\hat{x}_{ij}\hat{y}_{ij} + d\sqrt{\hat{x}^2_{ij} + \hat{y}^2_{ij} - d^2})v^y_{ij} = 0, \tag{13a}$$

$$(\hat{y}^2_{ij} - d^2)v^x_{ij} - (\hat{x}_{ij}\hat{y}_{ij} - d\sqrt{\hat{x}^2_{ij} + \hat{y}^2_{ij} - d^2})v^y_{ij} = 0, \tag{13b}$$

$$(\hat{x}^2_{ij} - d^2)v^y_{ij} - (\hat{x}_{ij}\hat{y}_{ij} + d\sqrt{\hat{x}^2_{ij} + \hat{y}^2_{ij} - d^2})v^x_{ij} = 0, \tag{13c}$$

$$(\hat{x}^2_{ij} - d^2)v^y_{ij} - (\hat{x}_{ij}\hat{y}_{ij} - d\sqrt{\hat{x}^2_{ij} + \hat{y}^2_{ij} - d^2})v^x_{ij} = 0. \tag{13d}$$

If all coefficients in Eq. (13) are non-zero, they define two lines, denoted R_1 and R_2, in the plane $\{(v^x_{ij}, v^y_{ij}) \in \mathbb{R}^2\}$ and the sign of $g_{ij}(v^x_{ij}, v^y_{ij})$ can be characterized based on the position of (v^x_{ij}, v^y_{ij}) relative to these lines (see Fig. 2). Recall that according to Eq. (9), the sign of the dot product $\hat{p}_{ij} \cdot v_{ij}$ indicates aircraft convergence or divergence. Let (P) be the equation of the line corresponding to the dot product $\hat{p}_{ij} \cdot v_{ij}$.

$$v^x_{ij}\widehat{x}_{ij} + v^y_{ij}\widehat{y}_{ij} = 0. \tag{P}$$

The line defined by (P) splits the plane $\{(v^x_{ij}, v^y_{ij}) \in \mathbb{R}^2\}$ in two half-planes, each of which representing converging and diverging trajectories, respectively. Consider the line normal to (P), denoted (N):

$$v^y_{ij}\hat{x}_{ij} - v^x_{ij}\hat{y}_{ij} = 0. \tag{N}$$

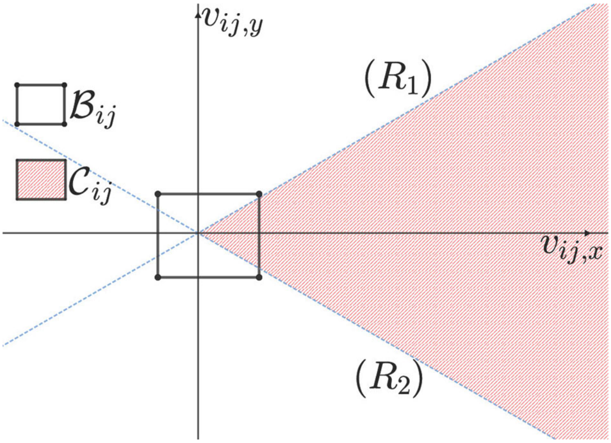


Fig. 3 Illustration of a two-aircraft conflict in the plane $\{(v_{ij}^x, v_{ij}^y) \in \mathbb{R}^2\}$. The inner box with black lines corresponds to the velocity bounds $\mathcal{B}$ in the deterministic scenario. The region hashed in red corresponds to the conflict region $\mathcal{C}$. If relative velocity $\mathcal{B}$ intersects with the conflict region $\mathcal{B}$, then there exists a risk of conflict. (Color figure online)

Recall that any point (v_{ij}^x, v_{ij}^y) such that $t_{ij}^{\min} \leq 0$ or $g_{ij}(v_{ij}^x, v_{ij}^y) \geq 0$ corresponds to a pair of conflict-free trajectories.

The conflict region of a pair of aircraft represents the set of relative velocity vectors (v_{ij}^x, v_{ij}^y), which corresponds to conflicts. As depicted in Fig. 2, the feasible region is non-convex. However, it can be represented via a disjunctive formulation. The following set of constraints can be used to represent this disjunction, which we model using the binary variable $z_{ij} \in \{0, 1\}$, where the values of $\gamma_{ij}^l, \phi_{ij}^l$ and $\gamma_{ij}^u, \phi_{ij}^u$ are coefficients of the lines R_1 and R_2 (as depicted in Fig. 2).

$$v_{ij}^y \widehat{x}_{ij} - v_{ij}^x \widehat{y}_{ij} \leq 0, \quad \text{if } z_{ij} = 1, \quad \forall (i,j) \in \mathcal{P}, \tag{14a}$$

$$v_{ij}^y \widehat{x}_{ij} - v_{ij}^y \widehat{y}_{ij} \geq 0, \quad \text{if } z_{ij} = 0, \quad \forall (i,j) \in \mathcal{P}, \tag{14b}$$

$$v_{ij}^y \gamma_{ij}^l - v_{ij}^x \phi_{ij}^l \leq 0, \quad \text{if } z_{ij} = 1, \quad \forall (i,j) \in \mathcal{P}, \tag{14c}$$

$$v_{ij}^y \gamma_{ij}^u - v_{ij}^x \phi_{ij}^u \geq 0, \quad \text{if } z_{ij} = 0, \quad \forall (i,j) \in \mathcal{P}. \tag{14d}$$

These equations characterize two convex sub-regions, each representing a set of conflict-free trajectories. The expressions of these lines (represented by Eq. 14) depend on aircraft initial positions, i.e. $\widehat{x}_{ij}, \widehat{y}_{ij}$. As shown in Theorem 1 of [21], the set of equations described by Eq. (14) is equivalent to Eq. (11). We recall the formal definition of the conflict region from [21] and Fig. 3.

Definition 1 (*Conflict region*, [21]) Consider a pair of aircraft $(i, j) \in \mathcal{P}$. Let $\mathcal{C}$ be the subset of $\mathbb{R}^2$ defined as:

$$\mathcal{C} \equiv \left\{(v_{ij}^x, v_{ij}^y) \in \mathbb{R}^2 : v_{ij}^x \gamma_{ij}^l - v_{ij}^y \phi_{ij}^l \geq 0 \wedge v_{ij}^x \gamma_{ij}^u - v_{ij}^y \phi_{ij}^u \leq 0\right\}. \tag{15}$$

which means that $\mathcal{C}$ is the conflict region of $(i, j) \in \mathcal{P}$.

For each aircraft $i \in \mathcal{A}$, we assume that the speed rate variable is lower bounded by $\underline{q}_i$ and upper bounded by $\overline{q}_i$, i.e.:

$$\underline{q}_i \leq q_i \leq \overline{q}_i, \qquad \forall i \in \mathcal{A}. \tag{16}$$

We assume that the heading deviation is lower bounded by $\underline{\theta}_i$ and upper bounded by $\overline{\theta}_i$, i.e.:

$$\underline{\theta}_i \leq \theta_i \leq \overline{\theta}_i, \qquad \forall i \in \mathcal{A}. \tag{17}$$

To derive lower and upper bounds on relative velocity components v_{ij}^x and v_{ij}^y, we re-arrange Eq. (6) using trigonometric identities:

$$\begin{aligned} v_{ij}^x =& q_i \widehat{v}_i \cos(\widehat{\theta}_i)\cos(\theta_i) - q_i \widehat{v}_i \sin(\widehat{\theta}_i)\sin(\theta_i) \\ & - q_j \widehat{v}_j \cos(\widehat{\theta}_j)\cos(\theta_j) + q_j \widehat{v}_j \sin(\widehat{\theta}_j)\sin(\theta_j), \quad (18a) \\ v_{ij}^y =& q_i \widehat{v}_i \sin(\widehat{\theta}_i)\cos(\theta_i) + q_i \widehat{v}_i \cos(\widehat{\theta}_i)\sin(\theta_i) \\ & - q_j \widehat{v}_j \sin(\widehat{\theta}_j)\cos(\theta_j) - q_j \widehat{v}_j \cos(\widehat{\theta}_j)\sin(\theta_j). \quad (18b) \end{aligned}$$

Let $\underline{v}_{ij,x}$, $\overline{v}_{ij,x}$ and $\underline{v}_{ij,y}$, $\overline{v}_{ij,y}$ be the lower and upper bounds for v_{ij}^x and v_{ij}^y, respectively. These bounds can be determined using Eq. (18) and the bounds on speed and heading control provided in Eqs. (16) and (17). The derived bounds on the relative velocity components can be used to define a box in the plane $\{(v_{ij}^x, v_{ij}^y) \in \mathbb{R}^2\}$.

Definition 2 (*Relative velocity box* [21]) Consider a pair of aircraft $(i, j) \in \mathcal{P}$. Let $\mathcal{B}$ be the subset of $\mathbb{R}^2$ defined as

$$\mathcal{B} \equiv \left\{(v_{ij}^x, v_{ij}^y) \in \mathbb{R}^2 : \underline{v}_{ij,x} \leq v_{ij}^x \leq \overline{v}_{ij,x}, \underline{v}_{ij,y} \leq v_{ij}^y \leq \overline{v}_{ij,y}\right\}. \tag{19}$$

$\mathcal{B}$ is the relative velocity box of $(i, j) \in \mathcal{P}$.

The relative velocity box $\mathcal{B}$ characterizes all possible trajectories for the pair $(i, j) \in \mathcal{P}$ based on the available 2D deconfliction resources, i.e. speed and heading controls. To characterize the set of conflict-free trajectories of a pair of aircraft $(i, j) \in \mathcal{P}$, we evaluate if there is any intersection between the relative velocity box $\mathcal{B}$ with the conflict region of this pair of aircraft.

In conflict avoidance problems, a common objective is to minimize the combined deviations of all aircraft (based on q_i and θ_i). This may lead several aircraft to perform minimal conflict avoidance maneuvers, which may not be desirable from an operational perspective. As observed in [28], even small deviations may result in costly recovery maneuvers. In order to generate trajectories where fewer aircraft are controlled and those that are controlled have a reduced total cost, an additional binary variable is introduced. This variable determines whether an aircraft is controlled or not. In this formulation, the avoidance stage has the objective to minimize the total deviation and the number of aircraft that are controlled. Let $\zeta_i \in \{0, 1\}$ for $i \in \mathcal{A}$ represent the control variable: $\zeta_i = 1$ represents the case where aircraft i modifies its speed and/or heading; and $\zeta_i = 0$ represents the case where aircraft i does not perform any conflict avoidance maneuver. To link binary variable ζ_i with trajectory control variables, observe that if aircraft i does not perform any maneuver, its speed and heading control should remain unchanged. This can be expressed as:

$$\underline{q}_i \zeta_i + (1 - \zeta_i) \leq q_i \leq \overline{q}_i \zeta_i + (1 - \zeta_i), \qquad \forall i \in \mathcal{A}, \tag{20a}$$

$$\underline{\theta}_i \zeta_i \leq \theta_i \leq \overline{\theta}_i \zeta_i, \qquad \forall i \in \mathcal{A}. \tag{20b}$$

Let $\widehat{v}_{ij}^x$ and $\widehat{v}_{ij}^y$ denote the relative velocity components of aircraft pair (i, j) if both aircraft i and j follow their nominal trajectories, i.e. perform no deviation; and let $\widehat{t}_{ij}^{\min}$ be the corresponding time of minimal separation. Let $\mathcal{P}_0$ be the set of aircraft that are predicted to be in conflict if aircraft follow their nominal trajectory: $\mathcal{P}_0 \equiv \{(i, j) \in \mathcal{P} : g_{ij}(\widehat{v}_{ij}^x, \widehat{v}_{ij}^y) <$

$0 \wedge \widehat{t}_{ij}^{\min} > 0\}$. If a pair of aircraft is predicted to be in conflict if no action is taken, then at least one of these aircraft must perform avoidance maneuvers. Therefore, the following cuts are valid inequalities:

$$\zeta_i + \zeta_j \geq 1, \quad \forall (i,j) \in \mathcal{P}_0. \tag{21}$$

To incorporate the impact of controlling aircraft in the objective function, we propose to minimize a weighted sum of two terms: a fixed cost linked to control variables (ζ_i) and a variable cost linked to trajectory deviation variables (q_i, θ_i). We denote λ_ζ the weight representing the fixed cost of controlling an aircraft, and we denote w in [0, 1] the weight used to capture the trade-off between heading and speed control deviations.

$$\min \sum_{i \in \mathcal{A}} (w\theta_i^2 + (1-w)(1-q_i)^2 + \lambda_\zeta \zeta_i). \tag{22}$$

The conflict avoidance stage can be formulated as presented in [21] incorporating the penalty control variable ζ_i as described in Eq. (20) and constraint (21).

Model 1 (MINLP Formulation for Conflict Avoidance)

$$\textit{Minimize} \quad \sum_{i \in \mathcal{A}} (w\theta_i^2 + (1-w)(1-q_i)^2 + \lambda_\zeta \zeta_i),$$

Subject to:

$$v_{ij}^x = q_i \widehat{v}_i \cos(\widehat{\theta}_i + \theta_i) - q_j \widehat{v}_j \cos(\widehat{\theta}_j + \theta_j), \qquad \forall (i,j) \in \mathcal{P}, \tag{23a}$$

$$v_{ij}^y = q_i \widehat{v}_i \sin(\widehat{\theta}_i + \theta_i) - q_j \widehat{v}_j \sin(\widehat{\theta}_j + \theta_j), \qquad \forall (i,j) \in \mathcal{P}, \tag{23b}$$

$$v_{ij}^y \hat{x}_{ij} - v_{ij}^x \hat{y}_{ij} \leq 0, \quad \text{if } z_{ij} = 1, \qquad \forall (i,j) \in \mathcal{P}, \tag{23c}$$

$$v_{ij}^y \hat{x}_{ij} - v_{ij}^x \hat{y}_{ij} \geq 0, \quad \text{if } z_{ij} = 0, \qquad \forall (i,j) \in \mathcal{P}, \tag{23d}$$

$$v_{ij}^y \gamma_{ij}^l - v_{ij}^x \phi_{ij}^l \leq 0, \quad \text{if } z_{ij} = 1, \qquad \forall (i,j) \in \mathcal{P}, \tag{23e}$$

$$v_{ij}^y \gamma_{ij}^u - v_{ij}^x \phi_{ij}^u \geq 0, \quad \text{if } z_{ij} = 0, \qquad \forall (i,j) \in \mathcal{P}, \tag{23f}$$

$$\underline{q}_i \zeta_i + (1-\zeta_i) \leq q_i \leq \overline{q}_i \zeta_i + 1(1-\zeta_i), \qquad \forall_i \in \mathcal{A}, \tag{23g}$$

$$\underline{\theta}_i \zeta_i \leq \theta_i \leq \overline{\theta}_i \zeta_i, \qquad \forall_i \in \mathcal{A}, \tag{23h}$$

$$\underline{\theta}_i \leq \theta_i \leq \overline{\theta}_i, \qquad \forall i \in \mathcal{A}, \tag{23i}$$

$$v_{ij}^x, v_{ij}^y \in \mathcal{B}, \qquad \forall (i,j) \in \mathcal{P}, \tag{23j}$$

$$\zeta_i \in \{0,1\}, \qquad \forall i \in \mathcal{A}, \tag{23k}$$

$$z_{ij} \in \{0,1\}, \qquad \forall (i,j) \in \mathcal{P}. \tag{23l}$$

This formulation is nonconvex due to the velocity constraint, which is nonconvex quadratic (Eq. 6). This results in a MINLP formulation which is challenging to solve and does not scale easily. Coefficients γ_{ij}^l, ϕ_{ij}^l and γ_{ij}^u, ϕ_{ij}^u (present in Eq. 14) can be pre-processed based on the sign of $\widehat{x}_{ij}$ and $\widehat{y}_{ij}$. Finally, $\mathcal{B}$ represents the bounds for the velocity variables (v_{ij}^x, v_{ij}^y).

2.3 Trajectory recovery

In this study, we consider a simple trajectory recovery model in which deviated aircraft perform a second maneuver to change their heading or speed to move towards their target point. This trajectory recovery model was first introduced in [28], and we build on this

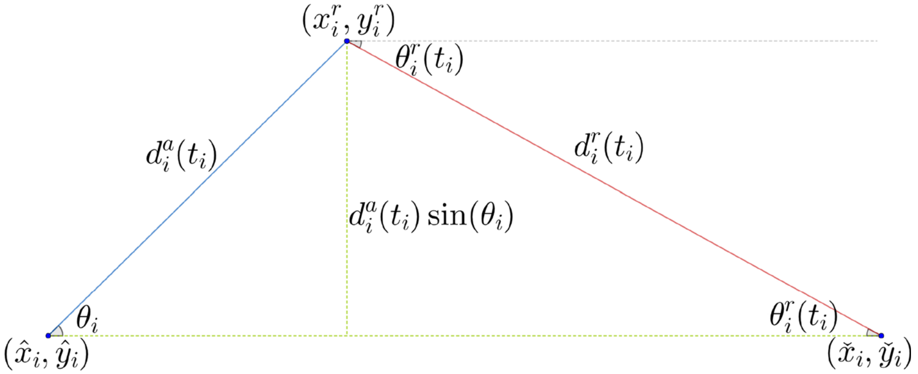


Fig. 4 Calculation of the recovery angle θ_i^r. The segment in blue corresponds to the avoidance stage, while the segment in red corresponds to the trajectory recovery stage. The segments in green are the projection of trigonometric functions in a right triangle. The initial position is denoted $(\hat{x}_i, \hat{y}_i)$, the point of recovery where the aircraft switches from its avoidance trajectory to its recovery trajectory is (x_i^r, y_i^r) and the aircraft target position is $(\check{x}_i, \check{y}_i)$. (Color figure online)

framework to develop a penalty-based approach. We next recall the main components of this trajectory recovery model.

Let $(\check{x}_i, \check{y}_i)$ be the coordinate of the target point of aircraft i, whose second maneuvers are performed to compensate the deviation performed during the avoidance stage. The speed maneuver q^r is simply defined such as the aircraft are returning their velocity magnitude to its nominal value, i.e. $q_i^r = 1$. For the heading changes, the deviation angle (also defined as the recovery angle, is the heading change maneuver required to compensate the initial deviation) is based on the time each aircraft moves from its avoidance trajectory towards its target point. For a given aircraft $i \in \mathcal{A}$, its recovery trajectory is defined as: $p_i(t) = [\widehat{x_i} + q_i \hat{v}_i \cos(\widehat{\theta_i} + \theta_i)t_i + \hat{v}_i \cos(\widehat{\theta_i})t, \widehat{y_i} + q_i \hat{v}_i \sin(\widehat{\theta_i} + \theta_i)t_i + \hat{v}_i \sin(\widehat{\theta_i})t]^T$, where t_i corresponds to the recovery time, i.e., the time in which the aircraft changes from its avoidance trajectory to its trajectory recovery (see Fig. 4). Hence, the deviation angle θ_i^r can be calculated as:

$$\theta_i^r(t_i) = \arcsin\left(\frac{d_i^a(t_i)\sin(\theta_i)}{d_i^r(t_i)}\right), \tag{24}$$

where the $d_i^a(t_i)$ (see Eq. (25)) corresponds to the distance flown during the conflict avoidance stage (from the initial position until the aircraft reaches t_i and changes to trajectory recovery) and $d_i^r(t_i)$ (see Eq. (26) to the distance flown during the recovery stage (from the time t_i where starts its trajectory until it reaches its destination point $(\check{x}_i, \check{y}_i)$) (see Fig. 4). If the aircraft did not change its heading angle, the recovery angle is not calculated.

$$d_i^a(t_i) = \sqrt{(\widehat{x_i} - x(t_i))^2 + (\widehat{y_i} - y(t_i))^2}, \quad \forall i \in \mathcal{A}, \tag{25}$$

$$d_i^r(t_i) = \sqrt{(x(t_i) - \check{x}_i)^2 + (y(t_i) - \check{y}_i)^2}, \quad \forall i \in \mathcal{A}. \tag{26}$$

In the avoidance stage, the core idea is to compare the distance between each pair of aircraft and derive time-independent expressions for all instances and impose them as constraints simultaneously. However, the same cannot be achieved when avoidance and recovery are calculated simultaneously. This is because the aircraft recovery trajectories are a function of the time when aircraft perform their recovery maneuver (t_i). It is evident that this expression is nonlinear with regard to speed and heading variables. Therefore, all the aircraft motion during trajectory recovery will depend explicitly on time, making it challenging to use the avoidance model described beforehand.

Given this context, it is expected that some simplification level is required to make any formulation using mathematical programming viable. One of these simplifications was implemented by [28], where the authors implemented a naive approach for trajectory recovery. It comprises a two-stage algorithm where avoidance and recovery are solved sequentially.

In this algorithm, the maneuvers determined during the avoidance stage are compensated in the recovery stage, which means that all maneuvers performed in the recovery are intended to redirect the aircraft towards its target destination. The algorithm preemptively uses the recovery stage to determines how costly the deviation is in terms of speed and angle applied. This way, the trajectory recovery can compensate for the cost during the avoidance stage. Another characteristic of this formulation is that recovery time is discrete. This reduces the feasible region and makes the options for trajectory recovery a limited, finite set. Based on those characteristics, this formulation can solve small to more significant instances (up to 30 aircraft) in a reasonable amount of time. More details can be found in [28].

The main drawback of this formulation is the lack of anticipation of the recovery costs at the avoidance stage. As both stages are solved separately, and each stage is solved only once, the solution obtained after solving both stages is myopic. This is not a significant issue for small instances due to the small amount of aircraft, but it becomes concerning in medium to large instances. As the number of aircraft and conflicts increase, the naive approach of [28] is expected to lead to weak solutions that do not anticipate recovery costs in designing aircraft trajectories. To explore the situation where the avoidance takes the recovery costs into account, we propose an alternative version of such an algorithm where the total cost is optimized throughout an iterative algorithm by projecting the cost of recovery operations into the avoidance stage to obtain non-trivial solutions (i.e. solutions that do not take into account any recovery cost. We next explain this algorithm in detail.

Each aircraft must perform opposing maneuvers during trajectory recovery to nullify any deviation that was performed during the avoidance stage. The goal is to guarantee that all pairs of aircraft are separated throughout the recovery stage. Since the separation condition in Eq. (14) is based on linear motion, we need to identify at which time each aircraft $i \in \mathcal{A}$ will perform its recovery maneuver, i.e. before and after its recovery time t_i. We denote A_i as the avoidance trajectory of aircraft i and R_i as its recovery trajectory. Given a pair (i, j) of aircraft, we need to ensure that aircraft are separated during all pairwise trajectory stages, denoted A_iA_j, A_iR_j, R_iA_j and R_iR_j. Observe that separation for the stage A_iA_j is already ensured by the solution of Model 1. If aircraft i and j recover at the same time, then aircraft would directly transition from A_iA_j to R_iR_j. Otherwise, if i (resp. j) recovers before j (resp. i), then A_iA_j will transition to R_iA_j (resp. A_iR_j) before transitioning to R_iR_j. We discretize aircraft recovery times to avoid a nonlinear model (due to trigonometric functions and quadratic terms) and, therefore, obtain a tractable formulation. Let $\mathcal{T}$ be the set of time periods available for recovery. We require:

$$t_i \in \{1\delta, 2\delta, \ldots, |\mathcal{T}|\delta\}, \tag{27}$$

where δ is the length of time periods. Abusing notation, we redefine the separation condition expressed in Eq. (11) as: $g_{ij}(m, n) \geq 0$ and $t_{ij}^{\min}(m, n) \leq 0$ where the pair (m, n) indicates the time period indices of recovery times t_i and t_j, respectively (for all cases $m \leq n$.). Let $\Omega_{X_iX_j}$ be the set of conflict-free pairs of recovery times for aircraft $i, j \in \mathcal{A}$ where X_i represents the state of the trajectory of aircraft i, i.e. A_i or R_i; and X_j represents the state of the trajectory of aircraft j, *i.e.* A_j or R_j. This set can be specified into three different sets corresponding to the three different states during the recovery stage. The set $\Omega_{R_iR_j}$ is

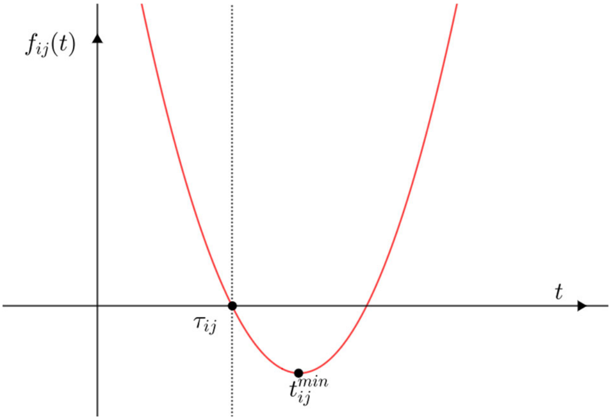


Fig. 5 Illustration of $f_{ij}(t)$ for a configuration with $g_{ij} < 0$ and $t_{ij}^{\min} > 0$. τ_{ij} represents the start time of the conflict

defined as:

$$\Omega_{R_iR_j} = \{(m, n) \in \mathcal{T}^2 : g_{R_iR_j}(m, n) \geq 0 \vee t_{R_iR_j}^{min}(m, n) \leq 0\}, \tag{28}$$

where $t_{R_iR_j}^{min}$ corresponds to the time of minimum separation between aircraft i and j in their recovery trajectory. For the states A_iR_j and R_iA_j an extra condition is required. Consider the state A_iR_j: if the segment of its trajectory corresponding to A_i and R_j are in conflict but aircraft i turns into recovery prior to the start of this conflict, then no conflict will occur. This is illustrated in Fig. 5 where $g_{A_iR_j} < 0$ and $t_{A_iR_j} > 0$. Let $\tau_{A_iR_j}(t_j)$ be the smallest root of $g_{A_iR_j} = 0$ if j recovers at time t_j. If aircraft i recovers prior to $\tau_{A_iR_j}(t_j)$, i.e. $t_i \leq \tau_{A_iR_j}(t_j)$, then the conflict will be avoided. Accordingly, we define:

$$\Omega_{A_iR_j} = \{(m, n) \in \mathcal{T}^2 : g_{A_iR_j}(n) \geq 0 \vee t_{A_iR_j}^{min}(n) \leq 0 \vee m \leq \tau_{A_iR_j}(n)\}, \tag{29a}$$

$$\Omega_{R_iA_j} = \{(m, n) \in \mathcal{T}^2 : g_{R_iA_j}(m) \geq 0 \vee t_{R_iA_j}^{min}(m) \leq 0 \vee n \leq \tau_{R_iA_j}(m)\}. \tag{29b}$$

Let ρ_{im} be a binary variable equal to 1 if aircraft $i \in \mathcal{A}$ recovers at time period $m \in \mathcal{T}$ and 0 otherwise. To track the states of aircraft pair (i, j) which are activated, we introduce two binary variables α_{ij} and β_{ij}. Those variables are used to identify whether $t_i < t_j$ $(\beta_{ij} = 1)$ which activates state R_iA_j, or if $t_i > t_j$ $(\alpha_{ij} = 1)$ which activates state A_iR_j. Variables α_{ij} and β_{ij} are defined via the constraints:

$$\alpha_{ij} \geq \frac{1}{|\mathcal{T}|}\left(\sum_{m\in\mathcal{T}} m\rho_{im} - \sum_{n\in\mathcal{T}} n\rho_{jn}\right), \qquad \forall (i, j) \in \mathcal{P}, \tag{30a}$$

$$\beta_{ij} \geq \frac{1}{|\mathcal{T}|}\left(\sum_{n\in\mathcal{T}} n\rho_{jn} - \sum_{m\in\mathcal{T}} m\rho_{im}\right), \qquad \forall (i, j) \in \mathcal{P}, \tag{30b}$$

$$\alpha_{ij} + \beta_{ij} \leq 1, \qquad \forall (i, j) \in \mathcal{P}. \tag{30c}$$

We use the following constraints to exclude conflicting trajectories from the feasible set. Observe that states A_iR_j and R_iA_j are conditional on the recovery times of t_i and t_j and thus the corresponding constraints are only active if i and j do not recover at the same time period. Specifically, if $t_i = t_j$, then the left-hand sides of constraints (30a) and (30b) are zero and the separation conditions (31a) and (31b) for the intermediate states $A_i R_j$ and $R_i A_j$ are relaxed since $\alpha_{ij} = \beta_{ij} = 0$ is feasible.

$$\rho_{im} + \rho_{jn} \le 2 - \beta_{ij}, \qquad \forall (i, j) \in \mathcal{P}, (m, n) \notin \Omega_{A_i R_j}, \tag{31a}$$

$$\rho_{im} + \rho_{jn} \le 2 - \alpha_{ij}, \qquad \forall (i, j) \in \mathcal{P}, (m, n) \notin \Omega_{R_i A_j}, \tag{31b}$$

$$\rho_{im} + \rho_{jn} \le 1, \qquad \forall (i, j) \in \mathcal{P}, (m, n) \notin \Omega_{R_i R_j}. \tag{31c}$$

Aircraft are assigned a recovery time via the constraint:

$$\sum_{m \in \mathcal{T}} \rho_{im} = 1, \qquad \forall i \in \mathcal{A}. \tag{32}$$

The second stage aims to identify the optimal time for aircraft to recover towards their target position. Different from [28], we adapted the objective function used in the avoidance model (in Model 1) as an aircraft-based coefficient a_i, for each $i \in \mathcal{A}$:

$$a_i = (1 - w)(1 - q_i)^2 + w\theta_i^2 + \lambda_\zeta \zeta_i, \quad \forall i \in \mathcal{A}, \tag{33}$$

For the recovery stage, we propose minimizing the total weighted recovery time, which accounts for maneuvers applied in the avoidance stage and aircraft recovery times:

$$\sum_{i \in \mathcal{A}} \sum_{m \in \mathcal{T}} a_i \rho_{im} t_m^2, \tag{34}$$

this expression comes from the region enclosed between the nominal trajectory and the deviating trajectory (as seen in Fig. 4) during the avoidance stage (i.e. the region formed by the lines describing the nominal and avoidance trajectories), and it is quadratic in t_i^2. The trajectory recovery formulation is summarized in Model 2, which is an MIQP (Mixed Integer Quadratic Programming).

Model 2 (Trajectory Recovery)

$$\begin{aligned}
&\textit{Minimize} \quad \sum_{i \in \mathcal{A}} \sum_{m \in \mathcal{T}} a_i \rho_{im} t_m^2,\\
&\textit{Subject to:}\\
&\alpha_{ij} \ge \frac{1}{|\mathcal{T}|}\left(\sum_{m \in \mathcal{T}} m\rho_{im} - \sum_{n \in \mathcal{T}} n\rho_{jn}\right), && \forall (i, j) \in \mathcal{P},\\
&\beta_{ij} \ge \frac{1}{|\mathcal{T}|}\left(\sum_{n \in \mathcal{T}} n\rho_{jn} - \sum_{m \in \mathcal{T}} m\rho_{im}\right), && \forall (i, j) \in \mathcal{P},\\
&\alpha_{ij} + \beta_{ij} \le 1, && \forall (i, j) \in \mathcal{P},\\
&\rho_{im} + \rho_{jn} \le 2 - \beta_{ij}, && \forall (i, j) \in \mathcal{P}, (m, n) \notin \Omega_{A_i R_j},\\
&\rho_{im} + \rho_{jn} \le 2 - \alpha_{ij}, && \forall (i, j) \in \mathcal{P}, (m, n) \notin \Omega_{R_i A_j},\\
&\rho_{im} + \rho_{jn} \le 1, && \forall (i, j) \in \mathcal{P}, (m, n) \notin \Omega_{R_i R_j},\\
&\sum_{m \in \mathcal{T}} \rho_{im} = 1, && \forall i \in \mathcal{A},\\
&\rho_{im} \in \{0, 1\}, && \forall i \in \mathcal{A}, m \in \mathcal{T},\\
&\alpha_{ij}, \beta_{ij} \in \{0, 1\}, && \forall (i, j) \in \mathcal{P}.
\end{aligned}$$

2.4 Penalty-based conflict resolution and trajectory recovery algorithm

The main idea behind the proposed penalty-based approach is to capture the recovery cost during the avoidance stage. Therefore, its goal is to preemptively consider the cost of the avoidance stage by anticipating the cost of trajectory recovery. It attempts to construct an optimized trajectory across both stages, representing a balance between optimizing avoidance and recovery. In this penalty-based approach, the solution of the recovery stage is used as a preemptive cost in the avoidance stage. In this case, the algorithm aims to find a trade-off between the deviation costs incurred during the avoidance and recovery stages. Let TC_i be the total cost per aircraft defined as the combined cost of avoidance and recovery.

$$TC_i = (1-w)(1-q_i)^2 + w\theta_i^2 + \lambda_\zeta \zeta_i + \lambda_t t_i^2, \qquad \forall i \in \mathcal{A}, \tag{35}$$

where λ_t (≥ 0) is the weight for the recovery time component.

Therefore, the objective function in the avoidance stage is modified to account for the anticipated cost of trajectory recovery. Let r_i be an algorithmic parameter used to account for the expected recovery cost of each aircraft $i \in \mathcal{A}$ at the avoidance stage. This parameter is updated before re-solving the avoidance problem based on the optimal solutions of the previous avoidance ($\boldsymbol{q}^\star$ and $\boldsymbol{\theta}^\star$) and recovery ($\boldsymbol{t}^\star$) stages. For each aircraft $i \in \mathcal{A}$, we define:

$$r_i = (1-w)(1-q_i^\star)^2 + w(\theta_i^\star)^2 + \lambda_t (t_i^\star)^2. \tag{36}$$

where $q^\star$,$\theta^\star$, and $t\star$ correspond to the optimal speed deviation, heading angle deviation and recovery time from a previous iteration.

By adding this expression into the avoidance stage, we aim to account for the impact of recovery costs within the avoidance stage. This approach aims to penalize the control of aircraft in the avoidance stage proportionally to their anticipated total trajectory deviation. Thus, the objective function in the avoidance stage can be rewritten as:

$$\min \sum_{i \in \mathcal{A}} w\theta_i^2 + (1-w)(1-q_i)^2 + \zeta_i(\lambda_\zeta + r_i). \tag{37}$$

The stopping criterion is based on the variation of the global total cost between two consecutive iterations, i.e.:

$$\Delta TC = \sum_{i \in \mathcal{A}} TC_i^n - TC_i^{n-1}, \tag{38}$$

where n corresponds to the index of the iteration. If the value of ΔTC is below a predefined threshold, the algorithm stops; otherwise, the recovery cost is updated, and the algorithm proceeds. This means that the algorithm reaches a compromised solution, where both avoidance and recovery are minimal, but not trivial, and both stages are equally considered in the optimization process. Instead of determining a myopic solution for the avoidance stage, i.e. where aircraft perform minimal deviations without considering trajectory recovery costs, the algorithm aims to find a trade-off such that recovery costs are anticipated at the avoidance stage.The optimal solution obtained using the algorithm is also a solution that the penalty-based approach cannot further improve.

The proposed iterative approach is summarized in Algorithm 1. The algorithm starts by initializing the algorithmic parameter $r_i = 0$ for all aircraft and solves the first-stage problem (conflict avoidance) using Model (1). The obtained first stage solution is then used to determine the sets $\Omega_{A_i R_j}$, $\Omega_{A_j R_i}$ and $\Omega_{R_i R_j}$ based on Eqs. (28) and (29). The second stage (trajectory recovery) is then solved to obtain aircraft recovery times using Model (2). The combination of the first- and second-stage solutions is used to evaluate the total cost of each aircraft (TC_i) using Eq. (35). The best feasible solution is updated if the corresponding global total cost $TC = \sum_{i \in \mathcal{A}} TC_i$ is lower than the incumbent. If the variation of the global total cost over two consecutive iterations ΔTC is less than a predefined threshold, the algorithm stops. Otherwise, the algorithmic parameter r_i is updated based on the last solutions of the first- and second-stage problems and the previous steps are repeated. If a time limit is attained while solving one of the stages, the best feasible solution is used to progress the iteration.

Algorithm 1 Penalty-based algorithm for the Conflict Resolution Problem with Trajectory Recovery

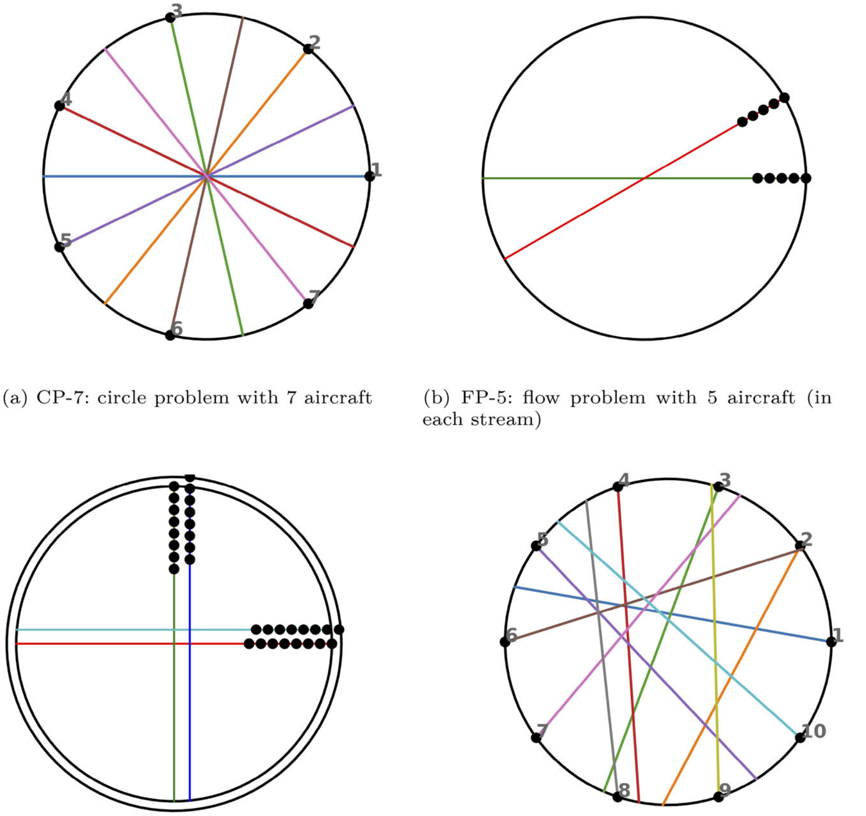
```
Require: $\mathcal{A}, \hat{\boldsymbol{\theta}}, \hat{\boldsymbol{v}}, \underline{\boldsymbol{q}}, \overline{\boldsymbol{q}}, \underline{\boldsymbol{\theta}}, \overline{\boldsymbol{\theta}}, \epsilon$
Ensure: $\boldsymbol{\zeta}^\star, \boldsymbol{q}^\star, \boldsymbol{\theta}^\star, \boldsymbol{t}^\star, TC$
$TC \leftarrow +\infty$
$r_i \leftarrow 0, \forall i \in \mathcal{A}$
CONVERGED $\leftarrow$ False
while CONVERGED = False do
    $\boldsymbol{q}, \boldsymbol{\theta}, \boldsymbol{\zeta} \leftarrow$ Solve Stage 1 - Avoidance using Model (1)
    if Infeasible then
        return INFEASIBLE
    end if
    Calculate Sets $\Omega_{A_iR_j}$, $\Omega_{AjR_i}$ and $\Omega_{R_iR_j}$ according to Eqs. (28)–(29)
    $\boldsymbol{t} \leftarrow$ Solve Stage 2 - Recovery using Model (2)
    if Infeasible then
        return INFEASIBLE
    else
        Update $TC$
        $\boldsymbol{\zeta}^\star \leftarrow \boldsymbol{\zeta}$
        $\boldsymbol{q}^\star \leftarrow \boldsymbol{q}$
        $\boldsymbol{\theta}^\star \leftarrow \boldsymbol{\theta}$
        $\boldsymbol{t}^\star \leftarrow \boldsymbol{t}$
        Calculate $\Delta TC$
    end if
    if $\Delta TC \leq \epsilon$ then
        CONVERGED $\leftarrow$ True
    else
        $r_i \leftarrow w\theta_i^2 + (1-w)(1-q_i)^2 + \lambda_t t_i^2$
    end if
end while
```

3 Numerical results

The experimental framework is introduced in Sect. 3.1. A detailed analysis of four instances is presented in Sect. 3.2. A sensitivity analysis on model parameters in conducted in Sect. 3.3. The computational performance of the proposed penalty-based approach is examined in Sect. 3.4.

3.1 Experimental framework

We test the performance of the proposed mixed-integer formulations and algorithm using four benchmark problems from the literature: the Circle Problem (CP), the Flow Problem (FP), the Grid Problem (GP) and the Random Circle Problem (RCP). The four types of benchmarking



(a) CP-7: circle problem with 7 aircraft

(b) FP-5: flow problem with 5 aircraft (in each stream)

(c) GP-8: grid problem with 8 aircraft (in each stream)

(d) RCP-10: random circle problem with 10 aircraft

Fig. 6 Examples of benchmarking instances for the Circle Problem (CP), Flow Problem (FP), Grid Problem (GP) and Random Circle Problem (RCP). All of the instances were also used in [21]

instances are illustrated in Fig. 6. The CP consists of a set of aircraft uniformly positioned on a circle heading towards its centre. Aircraft speeds are assumed to be identical; hence, the problem is highly symmetric (see Fig. 6a). The CP is notoriously difficult due to the geometry of initial aircraft configuration and has been widely used for benchmarking CD&R algorithms in the literature [13, 16, 20, 23, 29]. CP and RCP instances are named CP-N and RCP-N, respectively, where N is the total number of aircraft. While CP instances are highly symmetric, the RCP instances have each aircraft randomly and independently moving towards the centre of the circle with a unique initial heading and speed configuration. If a specific RCP instance is examined, it is referred to by its unique identifier ID and the name RCP-N-ID. For FP and GP, they are derived from [19], who formally introduced two additional structured problems which aim to represent more realistic air traffic configurations. The FP consists of two streams of aircraft separated by an angle α and anchored on the circumference of a circle. In each stream, aircraft are separated by at least 5 NM from each other. The GP consists of two FP instances separated by 15 NM diagonally. In our experiments, on each stream of aircraft in FP and GP instances, aircraft are organized in linear streams initially separated by 15 NM. FP and GP instances are named FP-N and GP-N, respectively, where N denotes the number of aircraft per stream. For all instances, we used $\alpha = \frac{\pi}{4}$. For reproducibility purposes, all formulations and data used for testing are made available online at the public repository https://github.com/acrp-lib/acrp-lib.

For all tests, a speed regulation range based on the subliminal speed control $[-6\%, +3\%]$ is used, and the typical heading control range $[-30°, +30°]$ is also used. For the weight, we use $w = 0.5$ and $\lambda_\zeta = 1$ in the objective in Eq. (33). This value was selected such that both heading and speed control terms were of a comparable order of magnitude, emphasizing penalizing heading control. For stage 2, a total of $|\mathcal{T}| = 15$ time periods are used, with a step of $\delta = 2$ min. To solve avoidance, the algorithm used was derived from the constraint generation algorithm implemented by [21] Disjunctive, and for the recovery stage, the model used is presented in Model 2. Both are implemented with CPLEX Python API and a time limit of 5 min per solving and 15 min per instance.

The proposed approach is compared to the algorithms presented by [28]. Those methods are respectively referred to as ExactNaive and GreedyNaive Recovery. The first method corresponds to a similar formulation of the method presented in this paper: it is the first iteration of Algorithm 1 where the avoidance and recovery stages are each solved only once. The second is a heuristic-based trajectory recovery procedure that iterates over all time steps and uses a priority list to decide which aircraft can be recovered at each time step. The priority list used is based on r_i values (36). The algorithm first sorts aircraft accordingly and iterates over time. At each time, the algorithm iterates over the sorted list of aircraft and checks if each aircraft can be recovered at the current time. The process is repeated until no aircraft can recover at the current time. The proposed algorithm has a worst-case time complexity of $\mathcal{O}(|\mathcal{T}||\mathcal{A}|^3)$. Details of our implementation of both benchmarking algorithms can be found in [28], and an example of their solution can be found in Fig. 7. Those models were slightly modified to be comparable to the algorithm presented in this paper. Specifically, the control variable ζ_i is introduced in the avoidance stage. Since the solution methods presented in [28] only solve each stage a single time, they are comparable to the first iteration of Algorithm (1) if all parameters r_i are initialized to zero.

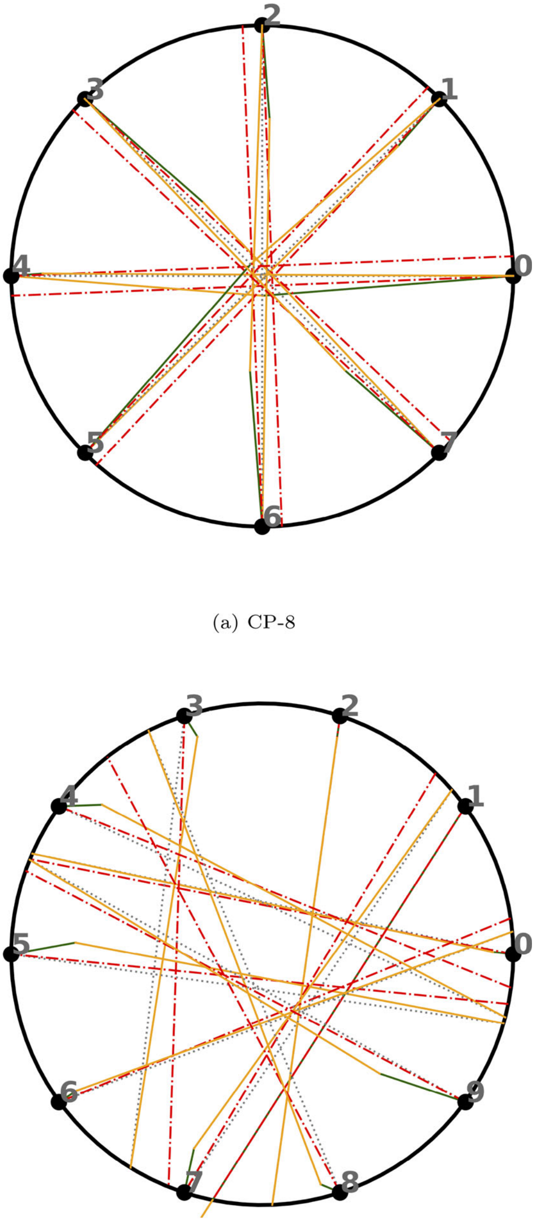
3.2 Illustration

In this section, we illustrate the proposed two-stage iterative algorithm. We display the optimal solution obtained for CP instances with eight aircraft and RCP instances with 20 aircraft. In Fig. 7 using Algorithm 1, dashed grey lines represent initial aircraft trajectories, green lines represent the avoidance trajectory of stage 1, and orange lines represent recovery trajectories of stage 2 using Model 2.

We compared the solutions from the penalty-based approach (in green and yellow) against the solution from the only avoidance separation (as presented by [21]) (in red). The figures show that the effect of the two computational stages imposes a larger deviation in the avoidance stage to guarantee a feasible trajectory recovery.

In CP-8, we observe that Algorithm 1 trades a larger deviation in the avoidance stage to obtain an earlier recovery time. In RCP-10-1, we can observe those deviations even more clearly. In addition, it is shown that only a few aircraft have their trajectories affected by minor deviations, which would be sufficient for guaranteeing separation conditions instead of deviating all aircraft (as shown in red). For RCP-20 and RCP-30 (see Fig. 10 in "Appendix A"), there are more aircraft to be altered, but overall, they show the same behaviour: more aircraft with higher avoidance costs and reduced recovery costs, with overall cost improved.

Fig. 7 Examples of two instances used to benchmark our formulation. For all figures, grey lines represent their nominal trajectory, orange lines represent the avoidance trajectory, and green trajectories correspond to their recovery. For those results are using special bounds for speed ([0.96,1.03]) and for heading angle([$\frac{-\pi}{3}$, $\frac{\pi}{3}$]). (Color figure online)



(a) CP-8

(b) RCP-10-1

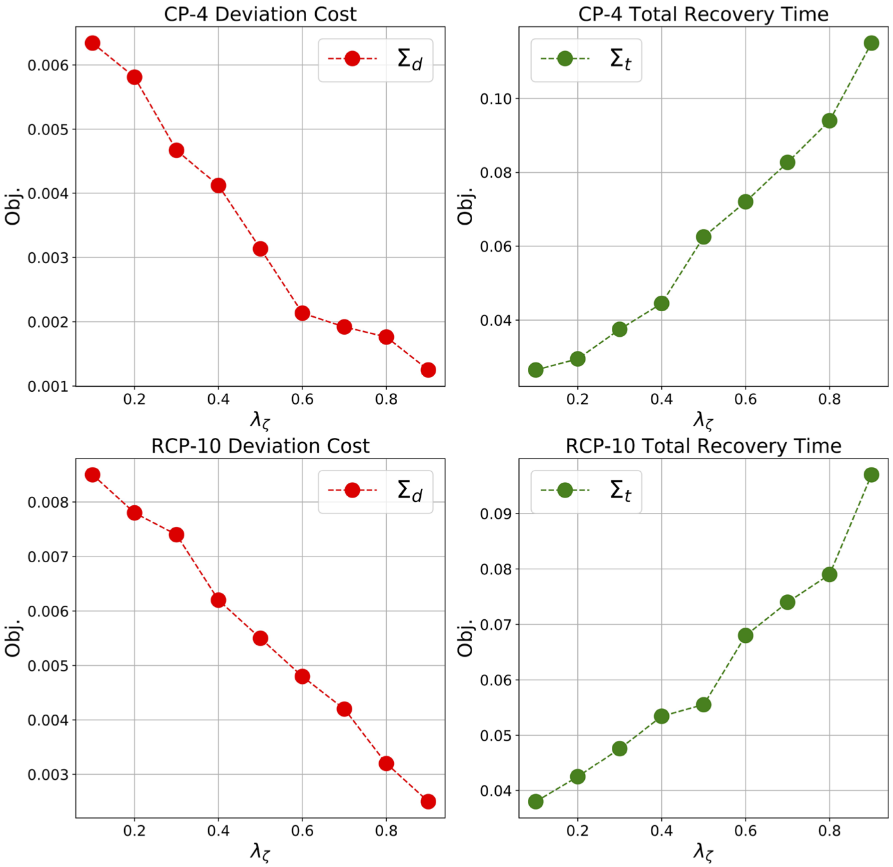


Fig. 8 Sensitivity analysis on the preference weight λ_ζ in the objective function Eq. (22). For all figures, Σ_d represents the total speed and heading deviation defined as $\sum_{i \in \mathcal{A}} (1 - q_i)^2 + \theta_i^2$ (in red) and Σ_t represents the total recovery time defined as $\sum_{i \in \mathcal{A}} t_i^2$ (in green). All instances tested here are CP-4 and RCP-10. (Color figure online)

3.3 Sensitivity analysis

To quantify the impact of the preference weights λ_t and λ_ζ in the proposed total cost function Eq. (35), we conduct numerical experiments on one instance of each of the four types of benchmarking instances for varying values of those parameters individually. This experiment focuses on the typical heading control range $[-30°, +30°]$. Our goal is to show that by varying those preference weights in $\lambda_t \in]0, 1[$ and $\lambda_\zeta \in]0, 1[$ the decision-maker can control the desired level of trade-off between the recovery time and total deviation. Recall that in the total cost function (37), λ_ζ is the coefficient of ζ_i and the total cost function has a minimal value when $\zeta_i = 0$; therefore, one can expect that increasing λ_ζ will tend to penalize the number of aircraft controlled and therefore decrease the recovery time. On the other hand, increasing the value of λ_t will tend to penalize the recovery time and, therefore, lead to potentially a greater total deviation of the aircraft.

This behaviour is confirmed in our numerical experiments. Specifically, we solve the instances CP-4 and RCP-10-3 for $\lambda_t, \lambda_\zeta = 0.1, \ldots, 0.9$ in steps of size 0.1, i.e. for a total of

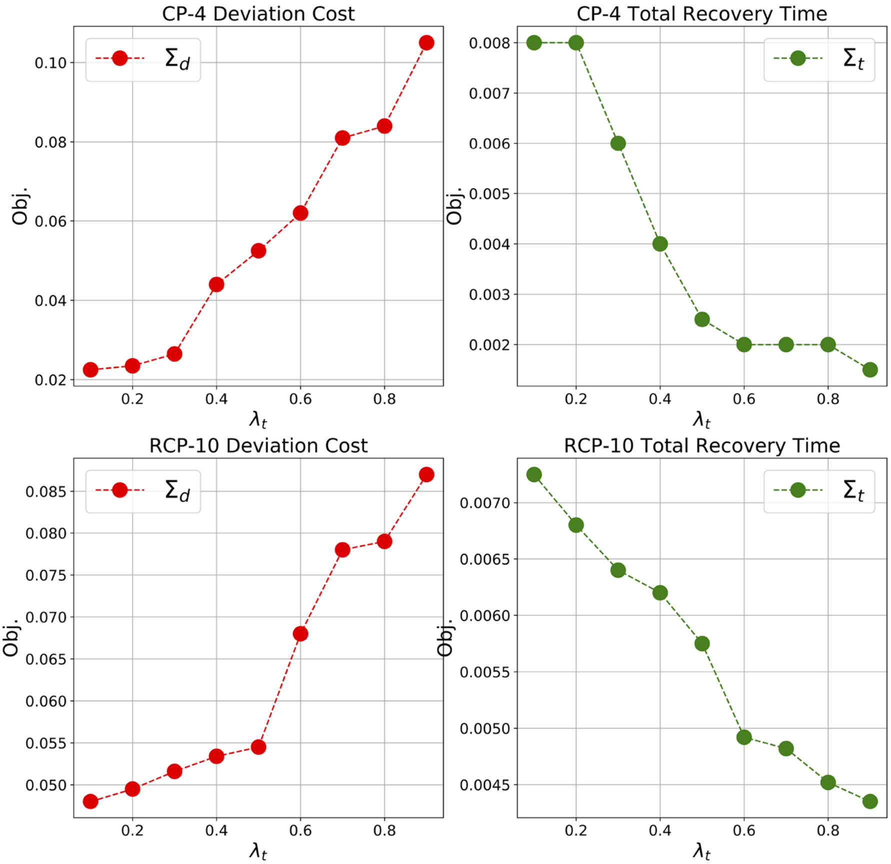


Fig. 9 Sensitivity analysis on the preference weight λ_t in the objective function Eq. (22). For all figures, Σ_d represents the total speed and heading deviation defined as $\sum_{i\in\mathcal{A}}(1-q_i)^2+\theta_i^2$ (in red) and Σ_t represents the total recovery time defined as $\sum_{i\in\mathcal{A}} t_i^2$ (in green). All instances tested here are CP-4 and RCP-10. (Color figure online)

9 values per instance. Each parameter is changed separately and independently, which means that when one parameter is changing, the other remains at a fixed value (in this case 0.5). All instances are solved using Algorithm 1. The change in the total deviation Σ_d and the total recovery time Σ_t are summarized in Figs. 8 and 9. Figure 8 shows the behaviour of CP-4 and RCP-10-3 when λ_ζ is changed: using the proposed objective function, the decision-maker can control which maneuver is prioritized by scaling up or down the preference weights λ_ζ accordingly. Higher values of λ_ζ will minimize the total deviation, while smaller values of λ_ζ will prioritize recovery time. We use $\lambda_\zeta = 0.25$ in the numerical experiments presented in the remainder of the paper. We find that increasing λ_ζ monotonically decreases the total deviation and monotonically increases the recovery time for all four instances tested. For λ_t (in Fig. 9), higher values lead to the minimization of recovery time and bigger deviation while increasing λ_t leads to smaller deviation in heading angle and speed while larger recovery time. We use $\lambda_t = 0.25$ in the numerical experiments presented in the remainder of the paper. For the parameter w, we are using the value $w = 0.5$ based on the sensitivity analysis presented in [21].

3.4 Performance of the penalty-based conflict resolution and trajectory recovery algorithm

The following tables represent the results for all instances. In the header, there are four groups of columns; the first group refers to the content of each instance: the first column $|\mathcal{A}|$ is the number of aircraft in each instance, and n_c is the number of conflicts. The second group refers to the avoidance stage: *Obj.* is the objective function; *Gap* (%) is the optimality gap; Time(s) is the runtime in seconds, followed by total deviation in terms of $|1 - q_i|$, $|\theta_i|$ and ζ_i, respectively. The third group contains the results from the recovery stage using Model 2. It is reported the average recovery time $\frac{1}{|A|}\sum_{i \in \mathcal{A}} t_i$ among all aircraft $\min\limits_{i \in \mathcal{A}} t_i$. All those values are reported after the last iteration. The fourth group reports the algorithm's overall performance: *Iter.* reports the number of iterations, and Time (s) reports the overall runtime in seconds (s). We also report the total cost in the first iteration TC^0 and in the last iteration TC^n. The gap tolerance was 5%. Table 1 summarizes the results for CP instances, while Tables 3, 5 and 7 for the FP, GP and RCP instances, respectively. For RCP instances, averages over 100 instances are provided, and the standard deviations are presented in parenthesis.

In the results for CP instances in Tables 1 and 2, it is expected that all the aircraft are moving towards the centre. In terms of the objective function, its value combines all three variables: speed change, heading angle and recovery time. However, due to their differences in magnitude, $\sum_{i \in \mathcal{A}} \zeta_i$ is the biggest component of the objective value.. It is consistently observed that one aircraft can remain in its initial condition in each instance. In most instances, we observed that the objective function is roughly equal to the number of aircraft present in that instance minus 1. This means that all other aircraft changed their trajectory from nominal while one remained as it was. This line up perfectly with the algorithm. It is more costly to alter all aircraft than only altering a few with larger deviations in each one. Such outcomes differ considerably from the results using naive avoidance, where the global solution would imply that all aircraft need to perform some maneuvers. In instances with ten or more aircraft, there are only slight deviations. Similarly, the variation in the heading angle is considered negligible for smaller instances. However, it does not increase proportionally with the number of conflicts. This observation shows that heading changes are used sporadically.

Moreover, the results for the penalty-based approach show that only a small amount of aircraft is required to be controlled, but the observed total deviations are more significant than in the naive approach. In terms of runtime, all instances with 12 aircraft can be solved within the time limit. In the recovery stage, the average recovery time does not increase with the number of aircraft (around 0.32 h on average), with the minimum recovery time around 5 min while the maximum peaks at almost 1 h. For all instances with 12 or more aircraft, the algorithm could not find a solution within the time limit. Most instances can be solved within the time limit for recovery time. However, for the denser instances, the time limit was reached during the avoidance stage, compromising the final solution after the recovery stage.

The algorithm only requires up to 3 iterations to obtain a solution within the convergence criterion used in terms of iterations. In comparison with the benchmark in Table 2, we can conclude by the TC value that the speed variation is relatively smaller and the angle deviation is overall larger compared to both ExactNaive (from [28]) and GreedyNaive (from [21]). This comparison can be verified by comparing the deviation values reported in Table 2 between all three models tested. This means that the recovery time is also considerably larger (in terms of value compared to those reported by the ExactNaive algorithm in Table 2). This is mainly due to the trivial solutions where the main focus is to solve the conflict as it comes without any consideration relating to the recovery (as it is done in the iterative algorithm). Another

Table 1 Summary of results for 2D CP instances with a speed control range of [−6%, +3%] and a heading control range of [−30°, +30°]

$\|A\|$	n_c	Avoidance						Recovery			Penalty-based algorithm			
		Obj	Gap (%)	Time (s)	$\sum_{i\in\mathcal{A}} \|1-q_i\|$	$\sum_{i\in\mathcal{A}} \|\theta_i\|$	$\sum_{i\in\mathcal{A}} \zeta_i$	Obj	Time (s)	$\frac{1}{\|A\|}\sum_{i\in\mathcal{A}} t_i$	TC^0	TC^n	Iter	Time (s)
4	6	3.00	1.79E−5	0.02	0.001	0.07	3	3.39E−4	0.23	0.36	3.07	2.83	2	0.25
5	10	4.00	9.93E−5	0.02	0.001	0.11	4	1.19E−4	0.30	0.11	4.11	3.91	2	0.31
6	15	5.00	9.94E−5	0.03	0.002	0.08	5	6.37E−4	0.66	0.35	5.08	4.57	3	0.69
7	21	6.00	9.88E−5	0.11	0.001	0.12	6	3.45E−4	3.85	0.26	6.20	5.58	3	3.96
8	28	7.00	9.94E−5	0.23	0.009	0.16	7	9.71E−4	1.86	0.30	7.17	6.74	3	2.09
9	36	8.00	1.00E−4	2.89	0.000	0.22	8	6.38E−4	15.1	0.20	8.22	7.40	3	18.0
10	45	9.01	9.99E−5	15.4	0.058	0.22	9	1.63E−3	10.6	0.30	9.29	8.36	3	26.1
11	55	10.01	1.00E−4	59.2	0.010	0.30	10	1.56E−3	29.6	0.27	10.3	9.60	2	355
12	66	11.01	1.00E−4	230	0.000	0.36	11	2.29E−3	34.2	0.27	11.3	10.4	2	264
13	78	12.01	3.59E−4	300	0.050	0.36	12	2.05E−3	55.9	0.24	12.4	11.4	2	355
14	91	13.01	5.83E−4	300	0.010	0.40	13	3.27E−3	52.3	0.27	13.4	12.3	2	352
15	105	14.02	6.80E−4	300	0.080	0.46	14	3.22E−3	300	0.25	14.5	13.3	2	600

All runtimes (Time) are reported in seconds while $\frac{1}{|A|}\sum_{i\in\mathcal{A}} t_i$ is in hours and Gap(%) expressed in percentage using the formulation applied in this papers

Table 2 Summary of comparison between the penalty-based algorithm and the exact-naive and greedy-naive presented in [28] using the TC as comparison key for the CP instances

$\lvert A\rvert$	Penalty-based			ExactNaive			GreedyNaive		
	$\sum_{i\in A} TC_i$	$\frac{1}{\lvert A\rvert}\sum_{i\in A} TC_i$	Time	$\sum_{i\in A} TC_i$	$\frac{1}{\lvert A\rvert}\sum_{i\in A} TC_i$	Time	$\sum_{i\in A} TC_i$	$\frac{1}{\lvert A\rvert}\sum_{i\in A} TC_i$	Time
4	2.83	0.71	0.25	3.07	0.77	0.21	3.12	0.78	0.11
5	3.91	0.78	0.31	4.11	0.82	0.26	4.56	0.91	0.13
6	4.57	0.76	0.69	5.08	0.85	0.59	5.43	0.92	0.25
7	5.58	0.80	3.96	6.20	0.89	3.37	6.91	0.98	1.51
8	6.74	0.84	2.09	7.17	0.90	1.78	7.42	0.92	0.82
9	7.40	0.82	18.0	8.22	0.91	15.3	8.71	0.96	6.85
10	8.36	0.84	26.1	9.29	0.93	22.1	9.35	0.93	9.92
11	9.60	0.87	355	10.3	0.94	301	10.8	0.98	56.1
12	10.40	0.87	264	11.3	0.94	224	12.1	1.08	97.1
13	11.40	0.88	355	12.4	0.95	301	13.9	1.06	135
14	12.30	0.88	352	13.4	0.96	299	14.5	1.03	164
15	13.30	0.89	600	14.5	0.97	600	15.3	1.02	270

Table 3 Summary of results for 2D FP instances with a speed control range of [− 6%, + 3%] and a heading control range of [− 30°, + 30°]

$\|A\|$	n_c	Avoidance						Recovery			Penalty-based algorithm			
		Obj	Gap (%)	Time (s)	$\sum_{i\in\mathcal{A}} \|1-q_i\|$	$\sum_{i\in\mathcal{A}} \|\theta_i\|$	$\sum_{i\in\mathcal{A}} \zeta_i$	Obj	Time (s)	$\frac{1}{\|A\|}\sum_{i\in\mathcal{A}} t_i$	TC^0	TC^n	Iter	Time(s)
4	8	6.00	0.00	0.25	0.001	0.08	6	3.65E−4	0.56	0.22	6.71	6.25	2	0.76
5	10	8.00	0.01	0.71	0.001	0.12	8	1.78E−4	0.93	0.28	8.43	8.31	2	1.21
6	12	9.00	0.03	1.01	0.005	0.21	9	6.68E−4	1.13	0.31	10.4	8.24	2	2.01
7	14	11.0	0.02	1.83	0.011	0.24	11	7.41E−4	2.14	0.46	11.8	11.1	3	2.48
8	16	13.0	0.03	2.52	0.031	0.27	13	4.95E−4	3.13	0.51	14.3	13.0	3	3.41
9	18	14.0	0.01	4.83	0.011	0.31	14	2.53E−3	4.05	0.54	15.9	14.1	3	4.14
10	20	17.0	0.02	6.01	0.031	0.35	17	4.55E−3	8.91	0.61	18.9	17.4	3	9.81
11	22	19.0	0.04	7.32	0.051	0.42	19	5.51E−3	14.8	0.64	20.1	19.0	3	36.5
12	24	21.0	0.01	8.13	0.033	0.13	21	4.81E−3	19.4	0.73	22.1	21.3	4	60.4
13	26	23.0	0.01	10.2	0.025	0.13	23	3.53E−2	26.1	0.78	24.3	22.9	4	137
14	28	24.0	0.01	21.3	0.011	0.24	24	1.53E−2	36.1	0.81	25.1	24.2	4	217
15	30	28.0	0.02	32.1	0.031	0.43	25	1.36E−2	50.4	0.88	29.3	28.1	4	300

All runtimes (time) are reported in seconds while $\frac{1}{|A|}\sum_{i\in\mathcal{A}} t_i$ is in hours and gap (%) expressed in percentage using the formulation applied in this papers

Table 4 Summary of comparison between the penalty-based algorithm and the exact-naive and greedy-naive presented in [28] using the TC as comparison key for the FP instances

$\lvert A\rvert$	Penalty-based			ExactNaive			GreedyNaive		
	$\sum_{i\in A} TC_i$	$\frac{1}{\lvert A\rvert}\sum_{i\in A} TC_i$	Time	$\sum_{i\in A} TC_i$	$\frac{1}{\lvert A\rvert}\sum_{i\in A} TC_i$	Time	$\sum_{i\in A} TC_i$	$\frac{1}{\lvert A\rvert}\sum_{i\in A} TC_i$	Time
4	6.25	0.78	0.76	6.98	0.87	0.65	7.15	0.89	0.51
5	8.31	0.83	1.21	8.81	0.88	1.35	8.62	0.86	1.32
6	8.24	0.68	2.01	9.18	0.76	2.15	10.4	0.87	2.13
7	11.1	0.79	2.48	12.4	0.88	3.37	14.1	1.00	3.55
8	13.0	0.81	3.41	14.7	0.91	4.31	15.9	0.99	4.13
9	14.1	0.78	4.14	16.8	0.93	5.19	17.8	0.98	5.93
10	17.4	0.87	9.81	19.2	0.96	10.3	20.4	1.02	10.5
11	19.0	0.86	36.5	22.4	1.02	40.2	24.2	1.10	41.2
12	21.3	0.88	60.4	23.9	0.99	71.4	25.5	1.06	72.3
13	22.9	0.88	137	23.3	0.89	141	26.1	1.00	132
14	24.2	0.86	217	27.8	0.99	266	28.3	1.01	239
15	28.1	0.93	300	29.9	0.98	300	30.7	1.02	300

Table 5 Summary of results for 2D GP instances with a speed control range of [$-$ 6%, + 3%] and a heading control range of [$-$ 30°, + 30°]

$\|A\|$	n_c	Avoidance						Recovery			Penalty-based algorithm			
		Obj	Gap (%)	Time (s)	$\sum_{i \in \mathcal{A}} \|1 - q_i\|$	$\sum_{i \in \mathcal{A}} \|\theta_i\|$	$\sum_{i \in \mathcal{A}} \zeta_i$	Obj	Time (s)	$\frac{1}{\|A\|} \sum_{i \in \mathcal{A}} t_i$	TC^0	TC^n	Iter	Time (s)
4	16	12.0	0.03	0.57	0.001	0.08	12	3.15E−4	3.57	0.42	12.71	12.0	2	1.64
5	20	16.0	0.12	1.45	0.005	0.32	16	2.78E−4	4.93	0.58	16.43	16.1	3	2.14
6	24	20.0	0.23	2.15	0.015	0.61	20	5.68E−4	6.31	0.31	21.12	20.0	3	4.11
7	28	25.0	0.55	3.78	0.041	0.84	25	6.41E−3	7.44	0.54	25.80	25.0	3	5.47
8	32	27.0	0.33	4.86	0.052	0.37	27	3.95E−3	8.32	0.42	28.19	27.0	3	8.19
9	36	31.0	0.31	4.67	0.024	0.51	31	7.83E−3	14.0	0.41	33.24	31.1	3	14.1
10	40	34.0	0.02	5.41	0.053	0.57	34	4.35E−3	18.6	0.47	36.21	34.0	4	19.8
11	44	40.0	0.54	7.32	0.081	0.92	40	5.71E−3	24.8	0.41	42.23	40.0	4	46.5
12	48	43.0	0.71	9.31	0.093	1.03	43	6.71E−3	39.4	0.87	44.64	43.1	4	65.4
13	52	48.0	0.81	14.2	0.085	1.32	48	5.13E−2	46.1	0.81	51.37	48.2	4	171
14	56	50.0	0.41	34.3	0.101	1.44	50	3.42E−2	76.1	0.69	55.36	50.2	4	275
15	60	53.0	0.21	46.1	0.231	2.43	53	2.34E−2	90.4	0.78	58.14	53.6	4	300

All runtimes (time) are reported in s while $\frac{1}{|A|} \sum_{i \in \mathcal{A}} t_i$ is in hours and gap (%) expressed in percentage using the formulation applied in this papers

Table 6 Summary of Comparison between the penalty-based algorithm and the exact-naive and greedy-naive presented in [28] using the TC as comparison key for the GP instances

$\|A\|$	Penalty-based			ExactNaive			GreedyNaive		
	$\sum_{i \in A} TC_i$	$\frac{1}{\|A\|} \sum_{i \in A} TC_i$	Time	$\sum_{i \in A} TC_i$	$\frac{1}{\|A\|} \sum_{i \in A} TC_i$	Time	$\sum_{i \in A} TC_i$	$\frac{1}{\|A\|} \sum_{i \in A} TC_i$	Time
4	12.0	0.75	1.64	13.1	0.81	1.44	15.1	0.94	1.39
5	16.1	0.80	2.14	17.8	0.89	2.11	18.3	0.90	2.14
6	20.0	0.83	4.11	22.7	0.94	3.62	24.2	1.00	3.50
7	25.0	0.89	9.47	27.4	0.97	8.32	28.9	1.03	7.12
8	27.0	0.84	14.1	29.1	0.90	12.5	30.1	0.94	10.9
9	31.1	0.86	34.1	33.4	0.92	33.6	35.5	0.98	25.3
10	34.0	0.85	59.8	36.1	0.90	47.3	37.2	0.93	35.7
11	40.0	0.90	76.5	43.8	0.99	65.2	44.9	1.02	56.3
12	43.1	0.89	105	45.9	0.95	98.1	46.8	0.97	79.3
13	48.2	0.92	171	50.2	0.96	142	53.1	1.02	121
14	50.2	0.89	275	53.6	0.95	241	55.7	0.99	232
15	53.6	0.91	300	57.2	0.95	300	59.1	0.98	300

Table 7 Summary of results for 2D RCP instances with a speed control range of [− 6%, + 3%] and a heading control range of [− 30°, + 30°]

$\|A\|$	n_c	Avoidance						Recovery			Penalty-based algorithm			
		Obj	Gap (%)	Time (s)	$\sum_{i\in\mathcal{A}} \|1-q_i\|$	$\sum_{i\in\mathcal{A}} \|\theta_i\|$	$\sum_{i\in\mathcal{A}} \zeta_i$	Obj	Time (s)	$\frac{1}{\|A\|}\sum_{i\in\mathcal{A}} t_i$	TC^0	TC^n	Iter	Time (s)
10	3.10 (1.5)	2.16 (0.93)	3.4E−5 (3.5E−5)	0.01 (0.00)	0.02 (0.03)	0.02 (0.02)	2.16 (0.93)	5.28E−5 (1.7E−4)	2.14 (3.04)	2.42 (2.26)	2.6 (1.60)	5.04 (1.01)	4.73 (0.95)	2.16 (3.05)
20	13.4 (3.4)	6.66 (1.19)	8.2E−5 (2.4E−5)	0.33 (0.31)	0.10 (0.08)	0.12 (0.06)	6.67 (1.19)	4.86E−4 (4.1E−4)	259 (68.8)	6.96 (1.87)	3.4 (0.50)	45.4 (1.43)	42.6 (1.34)	259 (61.1)
30	33.7 (5.7)	12.6 (1.46)	9.1E−5 (1.6E−5)	6.46 (9.02)	0.39 (0.18)	0.40 (0.15)	12.6 (1.47)	8.32E−3 (1.0E−2)	300	10.6 (2.53)	1.1 (0.21)	159 (2.21)	150 (2.08)	600

All runtimes (time) are reported in seconds while $\frac{1}{|A|}\sum_{i\in\mathcal{A}} t_i$ is in hours and gap (%) expressed in percentage using the formulation applied in this papers

Table 8 Summary of comparison between the penalty-based algorithm and the exact-naive and greedy-naive presented in [28] using the TC as comparison key for the RCP instances

$\|A\|$	Penalty-based			Exact Naive			Greedy Naive		
	$\sum_{i\in A} TC_i$	$\frac{1}{\|A\|}\sum_{i\in A} TC_i$	Time	$\sum_{i\in A} TC_i$	$\frac{1}{\|A\|}\sum_{i\in A} TC_i$	Time	$\sum_{i\in A} TC_i$	$\frac{1}{\|A\|}\sum_{i\in A} TC_i$	Time
10	4.73 (0.95)	0.47 (0.08)	2.16 (3.05)	5.07 (1.01)	0.50 (0.00)	1.16 (2.25)	7.41 (1.41)	0.74 (0.01)	0.78 (1.35)
20	42.6 (1.34)	2.17 (0.06)	250 (61.1)	45.4 (1.43)	2.13 (0.36)	127 (45.6)	63.5 (3.51)	3.17 (0.02)	2.35 (2.14)
30	150 (2.08)	5.0 (0.03)	600	159 (2.21)	5.12 (0.83)	356 (132)	204 (12.7)	6.81 (0.31)	5.71 (4.31)

observation (seen in Table 4) is that the recovery strategies presented in the ExactNaive and the GreedyNaive methods always provided a strategy where the recovery time was slightly higher for the latter and considerably higher for the former. This solidifies that using the recovery cost as a preemptive measure contributes to better performance and usage of the airspace configuration.

The results for FP instances are presented in Table 3. In the avoidance stage, the objective function is directly related to the number of conflicts in each instance, and it shows that it can be approximated by $|\mathcal{A}| - 2$. Similarly to the CP instances, the objective function values are a combination of the number of aircraft which performed a maneuver, speed and heading changes. For FP instances, these values largely amounted from the $\sum_{i \in \mathcal{A}} \zeta_i$. Speed change and angle deviation have a negligible contribution to those values. Due to the initial configuration of the aircraft in these instances, larger angle deviations are required to avoid conflict. This is justified by the higher values of $\sum_{i \in \mathcal{A}} |\theta_i|$ in comparison to $\sum_{i \in \mathcal{A}} |1 - q_i|$. The runtime for the avoidance stage is also quite small, with less than 40 s for any instance. In the recovery stage, the average recovery time increases with the number of aircraft (around 0.55 h on average), with the minimum recovery time around 20 min (as seen in Table 3, while the maximum peaks at almost 1 h. These values are justified by the fact that all aircraft must finish their routes at the same point (however, at different time intervals). In this case, aircraft in the leading position in each stream have small deviations, while any aircraft towards the end of the stream has more complex trajectories to avoid conflict with any aircraft ahead of them (it requires a combination of speed and angle in order to avoid them). For recovery time, most instances are solved within the time limit, but for the more prominent instances, the obtained solution in the avoidance stage and consequently in the recovery stage time out. The algorithm only requires up to 4 iterations to obtain a solution within the convergence criterion used in terms of iterations. In comparison with the benchmark in Table 4, we can conclude that the penalty-based offers the best cost (total and per aircraft) for all instances. However, it has a higher demand in terms of runtime. Because of the lack of symmetry, those instances do not allow a more balanced arrangement of maneuvers among the aircraft, which leads to higher averages of TC values (compared to CP instances). In addition, the iterative algorithm allows for those maneuvers to be refined further from the original configuration, which is unavailable to the other two strategies. By each iteration, each maneuver (either speed or heading changes and recovery time) can be recalculated in order to find a solution that minimizes total costs. Another observation is that the recovery strategies presented in the ExactNaive and the GreedyNaive methods always provided a strategy where the recovery time was slightly higher for the latter and considerably higher for the former.

In Tables 5 and 6, all the remarks made about FP instances can be further extended to GP instances. Especially since the trends observed in the former are even more evident in the latter. One important observation is that even though GP instances have a considerably larger number of conflicts, their runtime and total cost do not increase monotonically with the number of aircraft and the number of conflicts.

The results for RCP instances are presented in Table 7. For the avoidance stage, the objective function reflects the number of aircraft required to be separated. As indicated by the objective function values, which are at 2.16 for RCP-10, 6.66 for RCP-20 and 12.6 for RCP-30, which values are very close to the values obtained by $\sum_{i \in \mathcal{A}} \zeta_i$. Regarding speed, most aircraft do not perform relatively large deviations. The values are close to the nominal value, with only 0.02 for RCP-10, 0.10 for RCP-20 and 0.39 for RCP-30. For heading changes, the values are small, reflecting that most aircraft do not perform any deviation in heading either: 0.02 for RCP-10, 0.12 for RCP-20 and 0.40 for RCP-30. However, compared to the heading deviation obtained using the naive approach, it is clear that the heading deviation in that model is considerably smaller and that most of the total deviation is caused by speed changes. The opposite behaviour is observed here. The runtime for those instances is reasonably short, and instances with up to 30 aircraft can be solved in less than 10 s. Compared to the naive approach, which provided balanced solutions with equal contribution from speed and heading control, the results from the penalty-based approach state that speed and heading control can be kept to their minimum.

Finally, the optimality gap is negligible in all instances. In the recovery stage, the runtime increases considerably with the number of aircraft given that with more aircraft, the number of alternative routes to choose an optimal from increases drastically: up to 2.14 s for RCP-10, 259 s for RCP-20 and time out for 100% of the instances under 5 min of the time limit. Regarding the number of iterations, RCP-10 instances and RCP-20 instances require up to 4 iterations to achieve convergence while solving RCP-30 instances; only up to two iterations are executed due to the time limit. Compared with the benchmark methods (i.e. naive approach), the results in Table 8 show that the recovery time is significantly larger than the values obtained via the penalty-based algorithm. On the other hand, the runtime observed by the naive approach is relatively small, which highlights the computational trade-off at stake.

4 Conclusion

Findings are summarized in Sect. 4.1 and future research directions are discussed in Sect. 4.2.

4.1 Summary of findings

A new mixed-integer formulation and a penalty-based algorithm for the aircraft conflict resolution problem with trajectory recovery are proposed. We solve aircraft collision avoidance considering speed and heading controls, as well as binary aircraft selection variables. This is incorporated into an iterative two-stage algorithm that incorporates the projected cost of recovering aircraft to their target destination into the avoidance stage. The performance of this approach revealed that by echoing the cost associated with trajectory recovery, the avoidance stage decisions could be modified to minimize preemptively the overall cost of aircraft trajectory deviations. On average, a few iterations are sufficient to obtain improvements compared to naive approaches where more significant deviations in the avoidance stage are required but compensated with an earlier recovery time in the recovery stage. This reveals a trade-off between collision avoidance and trajectory recovery operations. The performance of the Penalty-based algorithm shows that the number of aircraft required to be maneuvered is reduced compared that of naive approaches. While the naive approaches provide more balanced solutions in terms of maneuvers across aircraft, they are also more demanding, i.e. most, if not all, aircraft are required to perform at least one maneuver. The Penalty-based algorithm developed in this study presents an alternative solution where fewer aircraft are controlled with higher deviations, which is expected to facilitate air traffic control operations.

4.2 Future research and perspectives

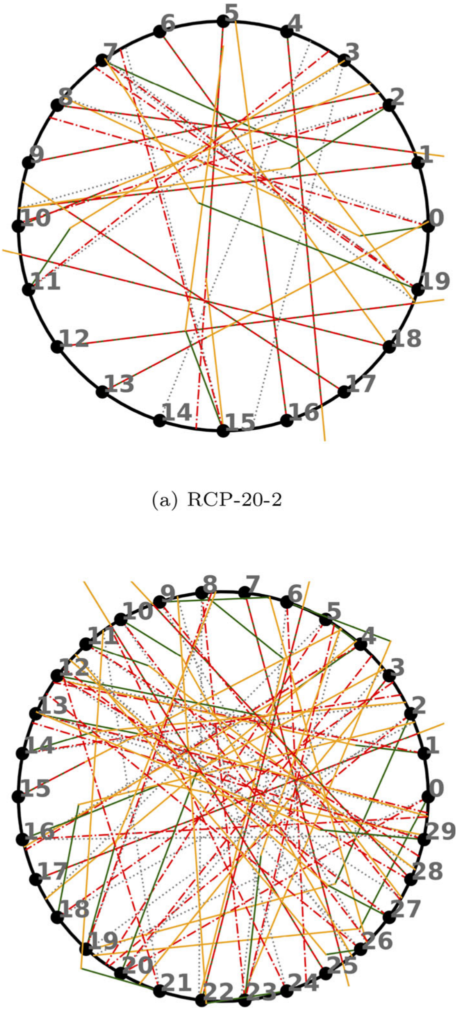
The most significant limitation in the iterative two-stage algorithm is conflict avoidance and trajectory recovery decomposition. Although this is a common practice in mathematical programming, there is no guarantee of the quality of the solution. Ideally, a global solution should be created by jointly optimizing both stages in a unified formulation. The nonlinearity and complexity of such formulations heavily challenge this. Therefore, further research is needed to model this problem more simply and efficiently. In addition, the cost of recovery is projected into the avoidance stage throughout the iterations. Alternatively, stochastic optimization methods can be used to determine the expected cost of recovery maneuvers already incorporated in the avoidance stage. Finally, the discretization of any variable is a limitation and modelling recovery time as a continuous variable may help reduce the total cost of trajectories.

Further testing is required to fully assess the impact of the control variables on solution quality. Notably, quantifying the impact of first-stage formulations with non-discretized heading changes and second-stage formulations continuously is critical to evaluate the cost of maneuver discretization. Both formulations presented here are deterministic, where no uncertainty is considered. An alternative formulation could consider the aircraft conflict and resolution problem under uncertainty where weather effects and measurement errors may affect the aircraft's speed. Such trajectory prediction uncertainty would also affect trajectory recovery operations, thus leading to a richer modelling framework capable of accounting for perturbations and measurement errors.

A Extra figures

See Fig. 10.

Fig. 10 Examples of two more instances used to benchmark our formulation. For all figures, grey lines represent their nominal trajectory, orange lines represent the avoidance trajectory, and green trajectories correspond to their recovery. For those results are using special bounds for speed ([0.96,1.03]) and for heading angle ($[\frac{-\pi}{3}, \frac{\pi}{3}]$). (Color figure online)



(a) RCP-20-2

(b) RCP-30-2

Funding Open Access funding provided by Aalto University.

References

1. Ibáñez-Gijón, J., Travieso, D., Navia, J.A., Montes, A., Jacobs, D.M., Frutos, P.L.: Experimental validation of COMETA model of mental workload in air traffic control. J. Air Transp. Manag. **108**, 102378 (2023)
2. Kuchar, J.K., Yang, L.C.: A review of conflict detection and resolution modeling methods. IEEE Trans. Intell. Transp. Syst. **1**(4), 179–189 (2000)
3. Pelegrín, M., d'Ambrosio, C.: Aircraft deconfliction via mathematical programming: review and insights. Transp. Sci. **56**(1), 118–140 (2022)
4. ICAO, A.: To the convention on international civil aviation. Montreal, Canada, vol. 1 (2010)
5. Pallottino, L., Feron, E.M., Bicchi, A.: Conflict resolution problems for air traffic management systems solved with mixed integer programming. IEEE Trans. Intell. Transp. Syst. **3**(1), 3–11 (2002)
6. Bilimoria, K.: A geometric optimization approach to aircraft conflict resolution. In: 18th Applied Aerodynamics Conference, p. 4265 (2000)
7. Frazzoli, E., Mao, Z.-H., Oh, J.-H., Feron, E.: Resolution of conflicts involving many aircraft via semidefinite programming. J. Guid. Control Dyn. **24**(1), 79–86 (2001)
8. Vela, A.E., Solak, S., Clarke, J.-P.B., Singhose, W.E., Barnes, E.R., Johnson, E.L.: Near real-time fuel-optimal en route conflict resolution. IEEE Trans. Intell. Transp. Syst. **11**(4), 826–837 (2010)
9. Vela, A.E., Salaun, E., Solak, S., Feron, E., Singhose, W., Clarke, J.-P.: A two-stage stochastic optimization model for air traffic conflict resolution under wind uncertainty. In: 2009 IEEE/AIAA 28th Digital Avionics Systems Conference, p. 2. IEEE (2009)
10. Alonso-Ayuso, A., Escudero, L.F., Martín-Campo, F.J.: Collision avoidance in air traffic management: a mixed-integer linear optimization approach. IEEE Trans. Intell. Transp. Syst. **12**(1), 47–57 (2011)
11. Alonso-Ayuso, A., Escudero, L.F., Martín-Campo, F.J.: Exact and approximate solving of the aircraft collision resolution problem via turn changes. Transp. Sci. **50**(1), 263–274 (2014)
12. Rey, D., Rapine, C., Fondacci, R., El Faouzi, N.-E.: Subliminal speed control in air traffic management: optimization and simulation. Transp. Sci. **50**(1), 240–262 (2016)
13. Cafieri, S., Rey, D.: Maximizing the number of conflict-free aircraft using mixed-integer nonlinear programming. Comput. Oper. Res. **80**, 147–158 (2017)
14. Cafieri, S., Durand, N.: Aircraft deconfliction with speed regulation: new models from mixed-integer optimization. J. Glob. Optim. **58**, 613–629 (2014)
15. Omer, J., Farges, J.-L.: Hybridization of nonlinear and mixed-integer linear programming for aircraft separation with trajectory recovery. IEEE Trans. Intell. Transp. Syst. **14**(3), 1218–1230 (2013)
16. Cafieri, S., Omheni, R.: Mixed-integer nonlinear programming for aircraft conflict avoidance by sequentially applying velocity and heading angle changes. Eur. J. Oper. Res. **260**(1), 283–290 (2017)
17. Cafieri, S., d'Ambrosio, C.: Feasibility pump for aircraft deconfliction with speed regulation. J. Glob. Optim. **71**(3), 501–515 (2018)
18. Cerulli, M., d'Ambrosio, C., Liberti, L., Pelegrín, M.: Detecting and solving aircraft conflicts using bilevel programming. J. Glob. Optim. **81**(2), 529–557 (2021)
19. Lehouillier, T., Omer, J., Soumis, F., Desaulniers, G.: Two decomposition algorithms for solving a minimum weight maximum clique model for the air conflict resolution problem. Eur. J. Oper. Res. **256**(3), 696–712 (2017)
20. Rey, D., Hijazi, H.: Complex number formulation and convex relaxations for aircraft conflict resolution. In: 2017 IEEE 56th Annual Conference on Decision and Control (CDC), pp. 88–93. IEEE (2017)
21. Dias, F.H., Hijazi, H., Rey, D.: Disjunctive linear separation conditions and mixed-integer formulations for aircraft conflict resolution. Eur. J. Oper. Res. **296**(2), 520–538 (2022)
22. Durand, N., Alliot, J.-M., Chansou, O.: Optimal resolution of en route conflicts. In: Procceedings of the 1rst USA/Europe Seminar (1997)
23. Durand, N., Alliot, J.-M.: Ant colony optimization for air traffic conflict resolution. In: Atm Seminar 2009, 8th USA/Europe Air Traffic Management Research and Development Seminar (2009)
24. Durand, N., Alliot, J.-M., Noailles, J.: Automatic aircraft conflict resolution using genetic algorithms. In: Proceedings of the 1996 ACM Symposium on Applied Computing, pp. 289–298 (1996)
25. Dougui, N., Delahaye, D., Puechmorel, S., Mongeau, M.: A light-propagation model for aircraft trajectory planning. J. Glob. Optim. **56**(3), 873 (2013)
26. Peyronne, C., Conn, A.R., Mongeau, M., Delahaye, D.: Solving air traffic conflict problems via local continuous optimization. Eur. J. Oper. Res. **241**(2), 502–512 (2015)
27. Omer, J.: A space-discretized mixed-integer linear model for air-conflict resolution with speed and heading maneuvers. Comput. Oper. Res. **58**, 75–86 (2015)
28. Dias, F.H., Rahme, S., Rey, D.: A two-stage algorithm for aircraft conflict resolution with trajectory recovery. In: ICRAT 2020 Ninth International Conference on Research in Air Transportation. ICRAT (2020)

29. Rey, D., Rapine, C., Dixit, V.V., Waller, S.T.: Equity-oriented aircraft collision avoidance model. IEEE Trans. Intell. Transp. Syst. **16**(1), 172–183 (2015)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.