Detecting and solving aircraft conflicts using bilevel programming

Martina Cerulli, Claudia d'Ambrosio, Leo Liberti, Mercedes Pelegrín

Noname manuscript No.
(will be inserted by the editor)

Detecting and solving aircraft conflicts using bilevel programming

**Martina Cerulli, Claudia D'Ambrosio,
Leo Liberti, Mercedes Pelegrín**

the date of receipt and acceptance should be inserted later

Abstract We present two bilevel programming formulations for the aircraft deconfliction problem: one based on speed regulation in k dimensions, the other on heading angle changes in 2 dimensions. We propose three reformulations of each problem based on KKT conditions and on two different duals of the lower-level subproblems. We also propose a cut generation algorithm to solve the bilevel formulations. Finally, we present computational results on a variety of instances.

Keywords mathematical programming · deconfliction · cutting plane · dominance · bilevel programming

1 Introduction

Two aircraft are said to be in conflict if their relative distance is less than a given safety threshold. By *aircraft deconfliction* we mean the set of strategies for detecting and solving conflicts among flying aircraft. A growing effort is dedicated to automate its optimal management, which is currently still widely performed on the ground by (human) Air Traffic Controllers (ATC). The development of urban air mobility will also rely on such decision-making support tools. Looking at a certain restricted airspace on a radar screen, the human controllers give real-time instructions to the pilots, based essentially on the change of their trajectories.

This research was partly funded by the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement n. 764759 ETN "MINOA".
This publication was supported by the Chair "Integrated Urban Mobility", backed by L'X - École Polytechnique and La Fondation de l'École polytechnique and sponsored by Uber.

Martina Cerulli · Claudia D'Ambrosio · Leo Liberti · Mercedes Pelegrín
LIX CNRS, École Polytechnique, Institut Polytechnique de Paris, 91128 Palaiseau France
E-mail: {mcerulli,dambrosio,liberti,pelegringarcia}@lix.polytechnique.fr

Many strategies can be used to avoid conflicts. In this paper, we focus on heading angle and speed changes. While heading angle changes (HAC) are often used by ATC to prevent collisions, speed changes are almost never performed in practice because of the tight speed modification restrictions imposed by air travel regulations. There are several reasons for the strict bounds, which include aircraft dynamics, passengers' comfort and the real-time nature of the decision process needed to make this maneuver efficient. In 2004, however, the concept of *Subliminal Control* was introduced in the context of the European project ERASMUS [1]. Subliminal speed control consists in allowing minor speed adjustments that have to be small enough to remain imperceptible to ATC, thus reducing their workloads. During the ERASMUS project, the efficiency of this method was validated through human-in-the-loop experiments by Drogoul et al. [2], who proposed two speed modulation ranges: a weak one from -6% to +3% and a strong one from -12% to +6%. Nowadays, the European Union is working towards implementing these types of approaches through the Single European Sky ATM Research (SESAR) project [3].

When we consider aircraft moving in a three-dimensional space, the need for subliminal speed changes becomes less relevant: speed regulation (SR) is not realistically performed while changing altitude, but only when aircraft are flying within a fixed altitude layer. This does not apply to Unmanned Aerial Vehicles (UAVs), however, which have different dynamics. As discussed in [4], the minimum distance between UAVs can be guaranteed by modifying their 3D trajectories, including by SR methods.

In this paper, we present different formulations of the Aircraft Deconfliction Problem (ADP) based on both HAC and SR. We present different bilevel programming formulations for each case, with several corresponding reformulations. Our formulations can also apply to other problems, such as coordinating a fleet of robots or unmanned vehicles in a complex obstacle-ridden region [5]. This includes, among others, the scenario of automated material handling vehicles moving on fixed routes in warehouses or production plants, to transport raw materials or perform tasks in production processes [6].

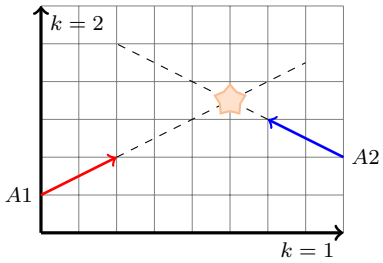

Fig. 1: Two conflicting aircraft in 2 dimensions

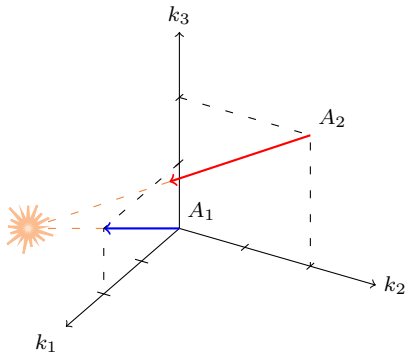

Fig. 2: Two conflicting aircraft in 3 dimensions

This paper considerably extends [7], where SR was applied to aircraft at the same flight level. This implies that the optimization takes place in the plane (see Figure 1). Here, we propose a more general approach by modeling the ADP through SR in a k-dimensional airspace (see Figure 2), an additional single-level reformulation based on the Wolfe dual, as well as a new bilevel formulation of the ADP based on the HAC strategy in the plane. Moreover, we extend the Cut Generation (CG) algorithm in [7] to the HAC based formulation, and provide a new analysis thereof.

The rest of the paper is organized as follows. We review the relevant literature in Sect. 2. In Sect. 3 we introduce our SR based ADP (SRADP) formulations: a natural formulation and a bilevel formulation, with three single-level reformulations. In Sect. 4 we introduce a bilevel formulation of the HAC based ADP (HACADP), and its reformulations. A CG solution algorithm for both SRADP and HACADP is presented in Sect. 5. In Sect. 6 we discuss computational results. Some concluding comments are given in Sect. 7.

2 Literature review

There exists a wide range of approaches for modeling and solving the ADP. Here, we give a non-exhaustive review on those related works proposing mathematical programming formulations based on SR, HAC, or both.

SR is one of the most common strategies for aircraft deconfliction by Mathematical Programming. In [8], speed is converted to travel time in order to minimize the total cost of all potential conflicts: the cost of a conflict depends on the time two aircraft spend travelling at a distance below the security threshold, since this time is proportional to the ATC monitoring and conflict solution effort. The decision variables of the proposed Mixed Integer Linear Program (MILP) are the arrival times at the different intersection points of the trajectories in the considered time horizon. An equity-oriented conflict resolution (ECR) model, based on these same variables, is introduced in [9]. It

proposes an innovative aircraft collision avoidance model promoting equitable solutions (airlines are equally affected by the trajectory adjustments). The ECR model combines three optimization stages, which can be formulated as MILP and attempt respectively to: maximize the number of solved conflicts; resolve conflicts in the fairest way; reduce the delay induced by the trajectory changes. A different kind of approach is proposed in [10,11], where Mixed Integer Nonlinear Programming (MINLP) formulations for the SRADP in the plane are considered. Specifically, [10] proposes a heuristic algorithm that decomposes the problem into smaller subproblems. The exact solutions of the subproblems are then combined to form a globally feasible but possibly sub-optimal solution of the original problem. A feasibility pump heuristic is proposed in [11]. This algorithm builds two sequences of points: one consisting of points that are feasible w.r.t. nonlinear constraints, and the other consisting of points satisfying the integrality conditions. The algorithm iterates until the two sequences converge to a feasible solution of the MINLP.

SR fails to solve frontal conflicts; moreover, it may not be sufficient to ensure safety if speed bounds are tight. Consequently, it is usually combined with other maneuvers, such as flight level reallocation. For instance, the authors of [12] present a MILP formulation where conflict situations are avoided by performing both speed and altitude changes over predefined routes. The objective is to minimize the expected fuel costs of the aircraft. Binary variables are used to assign flight levels, which indicate whether two aircraft fly at different altitudes, as well as allowing deconfliction of aircraft traversing the same flight level. A multi-objective MILP approach in a similar vein, based on both maneuver types, and aiming at an equitable distribution of the maneuvers over the aircraft, is proposed in [13]. In [14], two disjunctive formulations are proposed for the ADP based on speed and altitude changes. Their objective functions penalize the number of changes linearly or quadratically, giving rise to a MILP or a Mixed Integer Quadratic Program, respectively.

A part of the literature focuses on the geometric characterization of conflicts. These are then used in SR or HAC based models. This is, for example, the case of [15], where the geometric characteristics of aircraft trajectories are used in order to obtain closed-form expressions for single planar conflicts, based on SR and HAC alone, as well as closed-form expressions yielding minimum deviations from the original trajectories with combined SR and HAC. The authors of [16] present a geometric analysis of the conflicts leading to two MILPs: one for SR and another for HAC. The resulting separation constraints are linear on speeds and heading angles, respectively.

Other works interpret maneuvers as a combination of SR and HAC. In [17], SR and HAC are applied sequentially. First, a MINLP formulation minimizing HAC and subject to the safety separation condition is presented. Then, another MINLP is proposed which maximizes the number of collisions avoided by SR. These two models are solved sequentially. Then, using a two-step methodology, the solution of the SR MINLP is used as a pre-processing step for the HAC MINLP. SR and HAC are sometimes combined in the same formulation. In [18], the planar ADP is formulated as a nonconvex Quadratically Con-

strained Quadratic Program (QCQP) where the objective function minimizes the deviations from the original velocity vector. If the solution of the "natural" Semidefinite Programming relaxation of this QCQP has rank one, then the problem is solved; otherwise, a locally optimal and conflict-free solution with a certain crossing pattern can be obtained via a stochastic rank reduction procedure. A different approach, which also combines SR and HAC to find optimal aircraft maneuvers, is proposed in [19]. In this case, a formulation in complex numbers with disjunctive constraints is introduced; speed bounds are translated into nonconvex quadratic constraints by considering the Euclidean norm of the vectors of velocities; different relaxations of the resulting MINLP are then proposed, solved, and compared.

In this paper, we formulate the SRADP in k dimensions and the HACADP in two dimensions via bilevel programming. To the best of our knowledge, this is the first time that a bilevel approach is used to model the ADP. Our objective is to minimize the changes w.r.t. the original flight plan while still satisfying the safety distance on aircraft pairs. The terminology and symbols, as well as the formulæ aircraft separation, are taken from [10,11,17].

3 Aircraft deconfliction via speed regulation

The goal of the approach presented in this section is to minimize the total speed changes needed to satisfy the minimum safety distance for each pair of aircraft in a given time horizon. An important assumption is that changes occur instantaneously and that the new speeds remain constant in the time horizon. Specifically, given a constant speed for every aircraft, our formulation decides new optimal constant speeds satisfying the safety constraints. The sets, parameters, and variables used in all the mathematical formulations in this section are listed below.

- **Sets:**
  - $A = \{1, \ldots, i, \ldots, n\}$ is the set of aircraft flying in a shared airspace;
  - $K = \{1, \ldots, k_{\max}\}$ is the set of dimension indices.
- **Parameters:**
  - T is the length of the time horizon [hours];
  - d is the safety distance between aircraft [Nautical Miles NM] [1];
  - x^0_{ik} is the k-th component of the initial position of aircraft i;
  - v_i is the initially planned speed of aircraft i [NM/h];
  - u_{ik} is the k-th component of the direction of aircraft i;
  - q_i^{min} and q_i^{max} define the feasible range of the speed modification ratios of aircraft i s.t. $q_i^{\mathsf{min}} < 1 < q_i^{\mathsf{max}}$.
- **Variables:** q_i is the ratio of the implemented speed w.r.t. the initially planned speed of aircraft i: $q_i = 1$ if the speed is equal to the initially planned one, $q_i > 1$ if it is increased, $q_i < 1$ if it is decreased.

[1] 1 NM = 1852 m

3.1 Natural problem formulation

The following provides a "natural" way to formulate SRADP:

$$\min_{q} \quad \sum_{i \in A} (q_i - 1)^2 \tag{1a}$$

$$\forall i \in A \quad q_i^{\mathsf{min}} \leq q_i \leq q_i^{\mathsf{max}} \tag{1b}$$

$$\forall i < j \in A,\ t \in [0, T] \quad \sum_{k \in K} \left[(x_{ik}^0 - x_{jk}^0) + t(q_i v_i u_{ik} - q_j v_j u_{jk}) \right]^2 \geq d^2. \tag{1c}$$

Formulation (1a)–(1c) is a semi-infinite program, where the last line (Eq. (1c)) contains uncountably many constraints, which ensure aircraft separation. Specifically, Eq. (1c) requires the squared Euclidean distance between each two aircraft i and j to be greater than or equal to d^2 at each instant t in the time window $[0, T]$. The (convex) objective function is the sum of squared aircraft speed changes. This is equivalent to finding the feasible solution with the minimum speed change, which must be in $[q_i^{\mathsf{min}}, q_i^{\mathsf{max}}]$ for every aircraft i. As mentioned earlier, each aircraft i will start flying with the implemented speed, which is equal to $v_i q_i$.

3.2 Bilevel formulation of the problem

Since it is difficult to deal with the uncountably many constraints (1c) of the natural formulation, we propose a bilevel reformulation of SRADP with multiple lower-level subproblems. More details on the connections of semi-infinite and bilevel programming can be found in [20]. For an introduction to bilevel programming, see, for instance, [21].

The reformulation technique between semi-infinite and bilevel programs, used in the proof of Prop. 1 below, is well known but provides an introduction to some important concepts in bilevel programming.

Proposition 1 *The semi-infinite formulation in Eq.* (1a)-(1c) *can be reformulated exactly to the following bilevel formulation:*

$$\min_{q,t} \quad \sum_{i \in A} (q_i - 1)^2 \tag{2a}$$

$$\forall i \in A \quad q_i^{\mathsf{min}} \leq q_i \leq q_i^{\mathsf{max}} \tag{2b}$$

$$\forall i < j \in A \quad \min_{t_{ij} \in [0,T]} \sum_{k \in K} \left[(x_{ik}^0 - x_{jk}^0) + t_{ij}(q_i v_i u_{ik} - q_j v_j u_{jk}) \right]^2 \geq d^2. \tag{2c}$$

Proof We first introduce a bilevel formulation with a lower-level subproblem for each pair of aircraft $i < j \in A$:

$$\min_{q,\tau} \quad \sum_{i\in A}(q_i - 1)^2 \tag{3a}$$

$$\forall i \in A \quad q_i^{\mathsf{min}} \leq q_i \leq q_i^{\mathsf{max}} \tag{3b}$$

$$\forall i < j \in A \quad \tau_{ij} \in \operatorname*{arg\,min}_{t_{ij}\in[0,T]} \sum_{k\in K} \left[(x_{ik}^0 - x_{jk}^0) + t_{ij}(q_i v_i u_{ik} - q_j v_j u_{jk})\right]^2 \tag{3c}$$

$$\forall i < j \in A \quad \sum_{k\in K} \left[(x_{ik}^0 - x_{jk}^0) + \tau_{ij}(q_i v_i u_{ik} - q_j v_j u_{jk})\right]^2 \geq d^2. \tag{3d}$$

An optimal solution of each lower level subproblem, denoted by τ_{ij}, corresponds to the time instant at which aircraft i and j are closest. Note that each lower-level subproblem is an optimization problem on the lower-level variables t_{ij}, parametrized by the upper-level variables q_i and q_j. Next, we reformulate Eq. (3a)-(3d) by means of the *optimal value function* of each lower-level subproblem:

$$\varphi_{ij}(q) := \min_{t_{ij}\in[0,T]} \sum_{k\in K} \left[(x_{ik}^0 - x_{jk}^0) + t_{ij}(q_i v_i u_{ik} - q_j v_j u_{jk})\right]^2, \tag{4}$$

which yields the so-called (and equivalent) *optimal value formulation*:

$$\min_{q,\tau} \quad \sum_{i\in A}(q_i - 1)^2 \tag{5a}$$

$$\forall i \in A \quad q_i^{\mathsf{min}} \leq q_i \leq q_i^{\mathsf{max}} \tag{5b}$$

$$\forall i < j \in A \quad \sum_{k\in K} \left[(x_{ik}^0 - x_{jk}^0) + \tau_{ij}(q_i v_i u_{ik} - q_j v_j u_{jk})\right]^2 \leq \varphi_{ij}(q) \tag{5c}$$

$$\forall i < j \in A \quad \sum_{k\in K} \left[(x_{ik}^0 - x_{jk}^0) + \tau_{ij}(q_i v_i u_{ik} - q_j v_j u_{jk})\right]^2 \geq d^2. \tag{5d}$$

By inspection, Eq. (5c)-(5d) can be replaced by

$$\forall i < j \in A \qquad \varphi_{ij}(q) \geq d^2.$$

By Eq. (4), we obtain Eq. (2a)-(2c) as claimed. □

3.3 KKT reformulation

Certain bilevel programs, notably those having a convex lower-level subproblem, can be easily reformulated to single-level by replacing the lower-level subproblem by its KKT conditions [21, Sec. 3.5]. Assuming some regularity condition (e.g. Slater's condition) holds, this yields a single-level mathematical program with complementarity constraints. Given the KKT multipliers μ_{ij} and λ_{ij}

defined for each pair of lower-level subproblem constraints $-t_{ij} \leq 0$ and $t_{ij} \leq T$ respectively, we have:

$$\min_{q,t,\mu,\lambda} \quad \sum_{i \in A} (q_i - 1)^2 \tag{6a}$$

$$\forall i \in A \quad q_i^{\mathsf{min}} \leq q_i \leq q_i^{\mathsf{max}} \tag{6b}$$

$$\forall i < j \in A \quad \sum_{k \in K} \left[2t_{ij}\psi_{ijk}^2 + 2(x_{ik}^0 - x_{jk}^0)\psi_{ijk} - \mu_{ij} + \lambda_{ij} \right] = 0 \tag{6c}$$

$$\forall i < j \in A \quad \mu_{ij},\ \lambda_{ij} \geq 0 \tag{6d}$$

$$\forall i < j \in A \quad \mu_{ij}\, t_{ij} = 0 \tag{6e}$$

$$\forall i < j \in A \quad \lambda_{ij}\, t_{ij} - \lambda_{ij}\, T = 0 \tag{6f}$$

$$\forall i < j \in A \quad 0 \leq t_{ij} \leq T \tag{6g}$$

$$\forall i < j \in A \quad \sum_{k \in K} \left[(x_{ik}^0 - x_{jk}^0) + t_{ij}\psi_{ijk} \right]^2 \geq d^2 \tag{6h}$$

where the symbol ψ_{ijk} appearing in Eq. (6c) and (6h), and defined as:

$$\psi_{ijk} := q_i v_i u_{ik} - q_j v_j u_{jk}, \tag{7}$$

is used throughout the paper as short-hand for its definition in the right hand side.

Constraints (6c) (setting the gradient of the lower-level objective function equal to zero) correspond to the stationary condition, Eq. (6d) and (6g) to dual and primal feasibility conditions respectively, and Eq. (6e)-(6f) to complementary slackness. Eq. (6h) enforce the safety distance.

We remark that the complementarity constraints Eq. (6e)-(6f) involve products of continuous decision variables, and, therefore, define nonconvex feasible sets in general. A possible reformulation based on MILP modeling may define mixed-integer linear feasible sets instead, but also requires the determination of some big-M constant providing a valid bound to μ, λ. This problem, however, is **NP**-hard [22]. This particular reformulation, moreover, would not dispose of the nonconvexities in constraints Eq. (6c) and (6h). We therefore propose to solve the formulation above by means of global optimization techniques (see Sec. 6).

3.4 Dual reformulations

We propose another reformulation of the bilevel problem (2a)–(2c), which arises because the lower-level subproblems in (2c) are convex Quadratic Programs (QPs) occurring in constraints having general form:

$$\min_y \{ \frac{1}{2}\, y^\top Q_x y + p_x^\top y + c_x \mid Ay \geq b \} \geq \mathsf{const} \tag{8}$$

with Q_x positive semidefinite.

In general, given the dual variable z, by strong duality we have:

$$\begin{aligned} &\max_z\{\mathsf{LowerDualObj}(x,z) \mid \mathsf{LowerDualConstr}(x,z)\} \\ &= \min_y\{\frac{1}{2}\, y^\top Q_x y + p_x^\top y + c_x \mid Ay \geq b\}, \end{aligned} \tag{9}$$

where $\mathsf{LowerDualObj}(x,z)$ and $\mathsf{LowerDualConstr}(x,z)$ denote the objective function and the constraints of the dual problem of the left hand side of Eq. (8), respectively. If we impose the following inequality:

$$\max_z\{\mathsf{LowerDualObj}(x,z) \mid \mathsf{LowerDualConstr}(x,z)\} \geq \mathsf{const} \tag{10}$$

then, of course, Eq. (8) will also hold due to Eq. (9). The two constraints Eq. (8) and (10) are then equivalent. The following proposition, first proved in [7] w.r.t. Dorn's dual problem, can be easily generalized as follows.

Proposition 2 *Assume Eq.* (8) *occurs in a bilevel formulation. Replacing Eq.* (8) *by the constraint set:*

$$\left.\begin{array}{r} \mathsf{LowerDualObj}(x,z) \geq \mathsf{const} \\ \mathsf{LowerDualConstr}(x,z) \end{array}\right\} \tag{11}$$

yields a formulation with the same optima.

Proof We can replace Eq. (8) by the equivalent Eq. (10). If Eq. (10) is active (i.e. it is satisfied as an equality), then the maximum objective function value of the dual lower-level subproblem is const. Because of the max operator, its objective function cannot be greater than this value. This means that Eq. (11) can only be feasible when $\mathsf{LowerDualObj}(x,y)$ attains its maximum over its feasible region, defined by $\mathsf{LowerDualConstr}(x,z)$. If Eq. (10) is inactive, it has no effect on the optimum. Since Eq. (11) is a relaxation of Eq. (10), the same holds. □

Proposition 2 provides us with a scheme for reformulating (2a)–(2c), since, as already mentioned, Eq. (2c) is of the form (8). We observe that we can consider two different duals of the lower-level subproblems: Dorn's dual [23,24] (as we did for $K = \{1,2\}$ in [7]), and Wolfe's dual [25].

3.4.1 Dorn's dual reformulation

Given the dual variables g_{ij} and z_{ij} of each lower-level subproblem in the left hand side of Eq. (2c) (defined for constraints $-t_{ij} \leq 0$ and $t_{ij} \leq T$ respectively),

using Dorn's dual [23,24] and Prop. 2, the following reformulation of (2a)–(2c) follows:

$$\min_{q,g,z} \quad \sum_{i\in A}(q_i-1)^2 \tag{12a}$$

$$\forall i \in A \quad q_i^{\mathsf{min}} \le q_i \le q_i^{\mathsf{max}} \tag{12b}$$

$$\forall i<j\in A \quad -\sum_{k\in K}\psi_{ijk}^2\, g_{ij}^2 - T z_{ij} \ge d^2 - \sum_{k\in K}(x_{ik}^0 - x_{jk}^0)^2 \tag{12c}$$

$$\forall i<j\in A \quad -\frac{z_{ij}}{2} - \sum_{k\in K}\psi_{ijk}^2\, g_{ij} \le \sum_{k\in K}(x_{ik}^0 - x_{jk}^0)\psi_{ijk} \tag{12d}$$

$$\forall i<j\in A \quad z_{ij} \ge 0, \tag{12e}$$

obtained by the application of Prop. 2 to replace the lower-level subproblems of Eq. (2a)-(2c) by their Dorn duals in the variables g_{ij}, z_{ij} for each aircraft pair $i<j\in A$. This yields Eq. (12c)-(12d). Note that the primal lower-level variable t_{ij} does not appear in (12a)–(12e). This is not an issue because we just want to know the new aircraft speeds such that each potential conflict is avoided.

Proposition 3 *Eq.* (12a)–(12e) *is an exact reformulation of* (2a)–(2c).

Proof By Dorn's duality theory [23], (D) is a dual problem of (P):

$$\left.\begin{array}{rl} \min\limits_{y} & \frac{1}{2}y^\top Q y + p^\top y \\ & Ay \ge b \\ & y \ge 0 \end{array}\right\} \quad (P) \qquad \left.\begin{array}{rl} \max\limits_{g,z} & -\frac{1}{2}g^\top Q g + b^\top z \\ & A^\top z - Qg \le p \\ & z \ge 0 \end{array}\right\} \quad (D)$$

In our case, we have:

- $y := t_{ij}$,
- $Q := 2\sum\limits_{k\in K}\psi_{ijk}^2$,
- $p := 2\sum\limits_{k\in K}(x_{ik}^0 - x_{jk}^0)\psi_{ijk}$,
- $A := -1$,
- $b := -T$.

Recall that ψ_{ijk} is constant in the lower level since, by Eq. (7), it only depends on the upper-level variables q_i and q_j. By easy replacements and Prop. 2, with $\mathsf{const} = d^2 - \sum\limits_{k\in K}(x_{ik}^0 - x_{jk}^0)^2$ in Eq. (11), the formulation Eq. (12a)-(12e) follows. □

3.4.2 Wolfe's dual reformulation

Another single-level reformulation can be obtained using Prop. 2 and Wolfe's dual [25] of the convex lower-level subproblems in Eq. (2c). The lower-level

dual objective function is the Lagrangian of the lower-level primal problem in Eq. (2c)

$$\sum_{k\in K}\left[(x^0_{ik}-x^0_{jk})+t_{ij}\psi_{ijk}\right]^2+\alpha_{ij}(t_{ij}-T)-\beta_{ij}t_{ij},$$

where α_{ij} and β_{ij} are the Lagrangian multipliers associated to the constraints $-t_{ij}\leq 0$ and $t_{ij}\leq T$, respectively. Therefore, by Prop. 2, we obtain the following reformulation of Eq. (2a)-(2c):

$$\min_{q,t,\alpha,\beta} \quad \sum_{i\in A}(q_i-1)^2 \tag{13a}$$

$$\forall i\in A \quad q_i^{\mathsf{min}}\leq q_i\leq q_i^{\mathsf{max}} \tag{13b}$$

$$\forall i<j\in A \quad \sum_{k\in K}\left[(x^0_{ik}-x^0_{jk})+t_{ij}\psi_{ijk}\right]^2+\alpha_{ij}(t_{ij}-T)-\beta_{ij}t_{ij}\geq d^2 \tag{13c}$$

$$\forall i<j\in A \quad \sum_{k\in K}\left[2t_{ij}\psi^2_{ijk}+2(x^0_{ik}-x^0_{jk})\psi_{ijk}+\alpha_{ij}-\beta_{ij}\right]=0 \tag{13d}$$

$$\forall i<j\in A \quad \alpha_{ij},\beta_{ij}\geq 0. \tag{13e}$$

We note that the single-level reformulation presented above involves some of the KKT conditions as constraints: the stationarity condition Eq. (13d) and the nonnegativity of the Lagrangian multipliers Eq. (13e). The (nonlinear) complementarity constraints, however, are not needed in Wolfe's duality [25]. The obtained reformulation Eq. (13a)-(13e) is exact.

Proposition 4 *Eq.* (13a)-(13e) *is an exact reformulation of Eq.* (2a)-(2c).

Proof By Wolfe's duality theory [25], (D) is a dual problem of (P):

$$\left.\begin{array}{rl}\min\limits_{y} & \frac{1}{2}y^\top Qy+p^\top y+c\\ & Ay\geq b\\ & y\geq 0\end{array}\right\}\quad(P)\qquad\left.\begin{array}{rl}\max\limits_{g,z} & \mathcal{L}(y,\alpha,\beta)\\ & \frac{\partial\mathcal{L}}{\partial y}=0\\ & \alpha,\beta\geq 0\end{array}\right\}\quad(D)$$

with:

$$\mathcal{L}(y,\alpha,\beta)=\frac{1}{2}y^\top Qy+p^\top y+c+\alpha(b-Ay)-\beta y,$$

and

$$\frac{\partial\mathcal{L}}{\partial y}=Qy+p+\alpha-\beta.$$

In our case, we have:

- $y:=t_{ij}$,
- $Q:=2\sum\limits_{k\in K}\psi^2_{ijk}$,
- $p:=2\sum\limits_{k\in K}(x^0_{ik}-x^0_{jk})\psi_{ijk}$,
- $c:=\sum\limits_{k\in K}(x^0_{ik}-x^0_{jk})^2$,

- $A := -1$,
- $b := -T$.

Again, we recall that ψ_{ijk} is constant in the lower level because, by Eq. (7), it only depends on the upper level variables q_i and q_j. By easy replacements and Prop. 2, with $\mathsf{const} = d^2$ in Eq. (11), the formulation Eq. (13a)-(13e) follows. □

4 Aircraft deconfliction via heading angle changes

In this section, we present several formulations to model the HACADP. The goal is again to satisfy the minimum safety distance for each pair of aircraft while minimizing the total deviations with respect to the original flight plan. The outcome of the HACADP will be the set of new heading angles of the aircraft. After collision avoidance, the routes are corrected (as a post-processing step) in order to return each aircraft to its original trajectory. The new parameters and variables used in the mathematical formulations introduced in this section are listed below.

- **Parameters:**
  - ϕ_i is the initial heading angle of aircraft i
  - x_i^0 is the first component of the initial position of aircraft i
  - y_i^0 is the second component of the initial position of aircraft i
  - θ_i^{min} and θ_i^{max} are the bounds on heading angle variation, with $\theta_i^{\mathsf{min}} < 0$ and $\theta_i^{\mathsf{max}} > 0$
- **Variables:** θ_i is the heading angle variation for each aircraft i

4.1 Bilevel formulation of the problem

We introduce a bilevel formulation where the upper-level decision variables are the heading angle variations θ_i (for all $i \in A$) and the lower-level decision variables are t_{ij} (for all $i < j \in A$). Each lower-level subproblem is parametrized by the upper-level variables θ_i, θ_j.

$$\min_{\theta,t} \quad \sum_{i\in A} \theta_i^2 \tag{14a}$$

$$\forall i \in A \quad \theta_i^{\mathsf{min}} \le \theta_i \le \theta_i^{\mathsf{max}} \tag{14b}$$

$$\forall i < j \in A \quad \min_{t_{ij}\in[0,T]} \left[(x_i^0 - x_j^0) + t_{ij}(\cos(\phi_i + \theta_i)v_i - \cos(\phi_j + \theta_j)v_j)\right]^2 + \left[(y_i^0 - y_j^0) + t_{ij}(\sin(\phi_i + \theta_i)v_i - \sin(\phi_j + \theta_j)v_j)\right]^2 \ge d^2. \tag{14c}$$

The convex objective function of the upper level is the sum of squared heading angle changes, which are bounded by $[\theta_i^{\mathsf{min}}, \theta_i^{\mathsf{max}}]$. The objective of

each lower-level subproblem is to minimize the squared Euclidean distance between aircraft i and j over $t_{ij} \in [0, T]$. Its expression is obtained by defining the position $(x_i(t), y_i(t))$ of aircraft i at time t as

$$x_i(t) = x_i^0 + \cos(\phi_i + \theta_i) v_i t \quad \text{and} \quad y_i(t) = y_i^0 + \sin(\phi_i + \theta_i) v_i t.$$

Note that the lower-level objective function is also convex in t_{ij}. Similarly to Eq. (2c) of previous section, Eq. (14c) guarantees that the minimum squared distance between each pair within the time horizon is at least d^2.

The optimal heading angle change $\theta_i^\star$ for each aircraft i is obtained by solving Eq. (14a)-(14c). The trajectory deviation is followed until necessary to guarantee the safety distance, then the aircraft must return to their initial trajectories. Following what is done in [26,17], for each pair of aircraft the convex unconstrained QP Eq. (15) is solved as a post-processing step to return each aircraft to its original flight plan as soon as possible after conflict resolution:

$$\min_{t_{ij}} \left\| \begin{pmatrix} (x_i^0 - x_j^0) + t_{ij}(\cos(\phi_i + \theta_i^\star) v_i - \cos(\phi_j + \theta_j^\star) v_j) \\ (y_i^0 - y_j^0) + t_{ij}(\sin(\phi_i + \theta_i^\star) v_i - \sin(\phi_j + \theta_j^\star) v_j) \end{pmatrix} \right\|^2. \tag{15}$$

The objective function of the problem Eq. (15) is the relative squared Euclidean distance between aircraft, which is computed using the optimal heading angles of the proposed bilevel problem Eq. (14a)–(14c).

Once the optimal solution $\tau_{ij}^\star$ for problem Eq. (15) is found, we compute

$$T_i^\star := \max_{j : i \neq j} \tau_{ij}^\star \tag{16}$$

as the optimal time for which aircraft i can return to its initial trajectory after the deconfliction (there will be, for each i, a different τ_{ij} for every pair of aircraft (i, j)).

Knowing $(x_i(T_i^\star), y_i(T_i^\star))$ and the exit point from the air sector, it is easy to determine the new trajectory each aircraft has to follow in order to go back to its initial trajectory, as shown in Fig. 3.

As clarified in [26], when the aircraft are returning to the initial trajectories, new conflicts may occur. In order to ensure a conflict-free situation, the HACADP must be solved again. Sometimes, the maneuver to return to the initial trajectory must start when the aircraft is already close to the boundary of the air sector. This could lead to an angle variation exceeding the bounds. In this case, [26] proposes turning at the maximum bound and sending a warning message to the air traffic controllers of the following sector notifying that the aircraft is arriving in that sector at a different entry point w.r.t. the scheduled one.

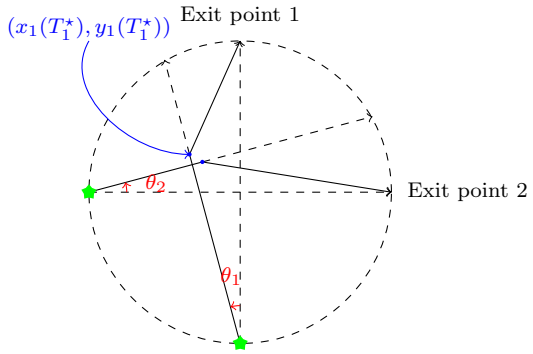

Fig. 3: New trajectories of two aircraft that, after conflict resolution, return to their initial trajectories (dashed lines)

4.2 KKT reformulation

We derive the KKT reformulation of Eq. (14a)–(14c), based on KKT multipliers λ_{ij} (resp. μ_{ij}) associated to constraints $t_{ij} \leq T$ (resp. $-t_{ij} \leq 0$). Assuming some regularity conditions hold, the reformulation is exact since the lower level is convex in the variable t_{ij} [21, Sec. 3.5].

$$\min_{\theta,t,\lambda,\mu} \quad \sum_{i\in A} \theta_i^2 \tag{17a}$$

$$\forall i \in A \quad \theta_i^{\mathsf{min}} \leq \theta_i \leq \theta_i^{\mathsf{max}} \tag{17b}$$

$$\forall i < j \in A \quad 2t_{ij}(c_{ij}^2 + s_{ij}^2) + 2(x_i^0 - x_j^0)c_{ij} + 2(y_i^0 - y_j^0)s_{ij} + \lambda_{ij} - \mu_{ij} = 0 \tag{17c}$$

$$\forall i < j \in A \quad \lambda_{ij},\ \mu_{ij} \geq 0 \tag{17d}$$

$$\forall i < j \in A \quad \lambda_{ij}\, t_{ij} - \lambda_{ij}\, T = 0 \tag{17e}$$

$$\forall i < j \in A \quad \mu_{ij}\, t_{ij} = 0 \tag{17f}$$

$$\forall i < j \in A \quad 0 \leq t_{ij} \leq T \tag{17g}$$

$$\forall i < j \in A \quad \left[(x_i^0 - x_j^0) + t_{ij}c_{ij}\right]^2 + \left[(y_i^0 - y_j^0) + t_{ij}s_{ij}\right]^2 \geq d^2, \tag{17h}$$

where the symbols c_{ij} and s_{ij} are shorthand for the following nonlinear expressions:

$$c_{ij} := \cos(\phi_i + \theta_i)v_i - \cos(\phi_j + \theta_j)v_j \tag{18}$$

$$s_{ij} := \sin(\phi_i + \theta_i)v_i - \sin(\phi_j + \theta_j)v_j. \tag{19}$$

The formulation in Eq. (17a)–(17h) is a single-level Nonlinear Programming (NLP) problem in the variables θ, t, λ, and μ. Constraints Eq. (17c) correspond to stationarity conditions of the lower-level subproblems, Eq. (17g) to primal

feasibility, Eq. (17d) to dual feasibility, Eq. (17e) and Eq. (17f) to complementarity conditions. We, again, require that the safety distance is satisfied for each pair of aircraft with constraints (17h).

4.3 Dual reformulations

We follow the procedure discussed in Sect. 3.4 in order to obtain two *dual reformulations* of Eq. (14a)-(14c). The first involves Dorn's dual of the lower-level subproblems, while the second involves Wolfe's dual.

4.3.1 Dorn's dual reformulation

Given the dual variables g_{ij} and z_{ij} of the lower-level subproblems, using Dorn's dual and Prop. 2, the following reformulation of Eq. (14a)-(14c) is obtained:

$$\min_{\theta,g,z} \quad \sum_{i\in A} \theta_i^2 \tag{20a}$$

$$\forall i \in A \quad \theta_i^{\mathsf{min}} \leq \theta_i \leq \theta_i^{\mathsf{max}} \tag{20b}$$

$$\forall i < j \in A \quad -g_{ij}^2(c_{ij}^2 + s_{ij}^2) - Tz_{ij} \geq d^2 - (x_i^0 - x_j^0)^2 - (y_i^0 - y_j^0)^2 \tag{20c}$$

$$\forall i < j \in A \quad -\frac{z_{ij}}{2} - (c_{ij}^2 + s_{ij}^2)g_{ij} \leq (x_i^0 - x_j^0)c_{ij} + (y_i^0 - y_j^0)s_{ij} \tag{20d}$$

$$\forall i < j \in A \quad z_{ij} \geq 0. \tag{20e}$$

The formulation in Eq. (20a)–(20e) is a single-level problem in the variables θ, g, and z, the exactness of which is proved below. Note that the primal lower-level variable t_{ij} does not appear in(20a)–(20e). This is not an issue because we just want to know the new heading angles such that each potential conflict is avoided.

Proposition 5 *Eq.* (20a)-(20e) *is an exact reformulation of Eq.* (14a)-(14c).

Proof By Dorn's duality theory [23], (D) is a dual problem of (P):

$$\left.\begin{array}{rl} \min\limits_{y} & \frac{1}{2}y^\top Qy + p^\top y \\ & Ay \geq b \\ & y \geq 0 \end{array}\right\} \quad (P) \qquad \left.\begin{array}{rl} \max\limits_{g,z} & -\frac{1}{2}g^\top Qg + b^\top z \\ & A^\top z - Qg \leq p \\ & z \geq 0 \end{array}\right\} \quad (D)$$

In our case, we have:

- $y := t_{ij}$,
- $Q := 2(c_{ij}^2 + s_{ij}^2)$,
- $p := 2(x_i^0 - x_j^0)c_{ij} + 2(y_i^0 - y_j^0)s_{ij}$,
- $A := -1$,
- $b := -T$.

We recall that c_{ij} and s_{ij} are constant in the lower level because, by Eq. (18)-(19), they only depend on the upper-level variables θ_i and θ_j. By easy replacements and Prop. 2, with $\mathsf{const} = d^2 - (x_i^0 - x_j^0)^2 - (y_i^0 - y_j^0)^2$ in Eq. (11), Eq. (20a)-(20e) follow. □

4.3.2 Wolfe's dual reformulation

Using Prop. 2 and Wolfe's dual of each lower-level subproblem in the variables α_{ij} and β_{ij}, we obtain the following single-level reformulation of of Eq. (14a)-(14c):

$$\min_{\theta,t,\alpha,\beta} \quad \sum_{i\in A} \theta_i^2 \tag{21a}$$

$$\forall i \in A \quad \theta_i^{\mathsf{min}} \leq \theta_i \leq \theta_i^{\mathsf{max}} \tag{21b}$$

$$\forall i < j \in A \quad \left[(x_i^0 - x_j^0) + t_{ij}c_{ij}\right]^2 + \left[(y_i^0 - y_j^0) + t_{ij}s_{ij}\right]^2 + \alpha_{ij}(t_{ij} - T) - \beta_{ij}t_{ij} \geq d^2 \tag{21c}$$

$$\forall i < j \in A \quad 2t_{ij}(c_{ij}^2 + s_{ij}^2) + 2(x_i^0 - x_j^0)c_{ij} + 2(y_i^0 - y_j^0)s_{ij} + \alpha_{ij} - \beta_{ij} = 0 \tag{21d}$$

$$\forall i < j \in A \quad \alpha_{ij}, \beta_{ij} \geq 0. \tag{21e}$$

With Eq. (21c), the Lagrangian of each lower-level subproblem is required to exceed the minimum required safety distance. The stationarity KKT condition (gradient of the Lagrangian equal to zero) corresponds to (21d). Constraints (21e) impose the nonnegativity of the dual variables α_{ij} and β_{ij}. The exactness of formulation (21a)-(21e) is proved below.

Proposition 6 *Eq.* (21a)-(21e) *is an exact reformulation of Eq.* (14a)-(14c).

Proof By Wolfe's duality theory [25], (D) is a dual problem of (P):

$$\left.\begin{array}{rl} \min\limits_{y} & \frac{1}{2}y^\top Qy + p^\top y + c \\ & Ay \geq b \\ & y \geq 0 \end{array}\right\} \quad (P) \qquad \left.\begin{array}{rl} \max\limits_{g,z} & \mathcal{L}(y,\alpha,\beta) \\ & \frac{\partial \mathcal{L}}{\partial y} = 0 \\ & \alpha, \beta \geq 0 \end{array}\right\} \quad (D)$$

with:

$$\mathcal{L}(y,\alpha,\beta) = \frac{1}{2}y^\top Qy + p^\top y + c + \alpha(b - Ay) - \beta y,$$

and

$$\frac{\partial \mathcal{L}}{\partial y} = Qy + p + \alpha - \beta.$$

In our case, we have:

- $y := t_{ij}$,
- $Q := 2(c_{ij}^2 + s_{ij}^2)$,
- $p := 2(x_i^0 - x_j^0)c_{ij} + 2(y_i^0 - y_j^0)s_{ij}$,

- $c := (x_i^0 - x_j^0)^2 + (y_i^0 - y_j^0)^2$,
- $A := -1$,
- $b := -T$.

Again we recall that c_{ij} and s_{ij} are constant in the lower level because, by Eq. (18)-(19), they only depend on the upper-level variables θ_i and θ_j. By easy replacements and Prop. 2, with $\mathsf{const} = d^2$ in Eq. (11), Eq. (21a)-(21e) follow. □

5 Cut generation algorithm

Cutting-plane approaches are one of the major techniques used for solving linear, quadratic [27], and convex semi-infinite programs. In [7], we proposed a tailored CG algorithm for the bilevel formulation Eq. (2a)–(2c) in two dimensions. We recall our CG algorithm in Sec. 5.1 for SRADP in k dimensions. Then, we discuss the possibility of exploiting some dominance relationships among cuts in order to speed up the CG algorithm. In Sec. 5.2, we tailor the CG algorithm for the bilevel formulation Eq. (14a)–(14c) of HACADP.

The problem solved at each iteration of the CG the algorithm is nonconvex. In our implementation, its solution is obtained either with global solvers or, in the interest of efficiency, by executing a local NLP solver several times within a multistart procedure that starts from randomly chosen points.

5.1 Cut generation algorithm for ADP via speed regulation

Algorithm 1 is a solution algorithm for the bilevel formulation (2a)-(2c), which iteratively defines the feasible set of the upper-level problem by means of quadratic cuts in the upper-level variables q. At each iteration h, the relaxation R_h of the original bilevel problem, obtained by considering the upper-level problem together with the cuts added in previous iterations, is solved. At the outset, R_1 is:

$$\begin{aligned} \min_{q} \quad & \sum_{i \in A} (q_i - 1)^2 \\ \forall i \in A \quad & q_i^{\mathsf{min}} \leq q_i \leq q_i^{\mathsf{max}}. \end{aligned}$$

Algorithm 1 CG algorithm for SRADP

1: Let $h = 1$. Initialize the relaxation R_h of the bilevel program, obtained by considering the upper-level problem only.
2: **while** true **do**
3: Solve R_h to obtain the optimal solution q^*.
4: For each aircraft pair (i, j), compute the instant $\tau_{ij}^h \in [0, T]$ as

$$\tau_{ij}^h = -\frac{\sum_{k\in K}(x_{ik}^0 - x_{jk}^0)(q_i^* v_i u_{ik} - q_j^* v_j u_{jk})}{\sum_{k\in K}(q_i^* v_i u_{ik} - q_j^* v_j u_{jk})^2}. \tag{22}$$

5: **if** $\sum_{k\in K}((x_{ik}^0 - x_{jk}^0) + \tau_{ij}^h(q_i^* v_i u_{ik} - q_j^* v_j u_{jk}))^2 \geq d^2 \quad \forall i < j \in A$ **then**
6: The algorithm terminates and q^* is the optimal solution of the bilevel formulation.
7: **else**
8: For each pair (i, j) violating the inequality, define R_{h+1} as R_h with the adjoined inequality:

$$\sum_{k\in K}((x_{ik}^0 - x_{jk}^0) + \tau_{ij}^h(q_i v_i u_{ik} - q_j v_j u_{jk}))^2 \geq d^2. \tag{23}$$

9: $h := h + 1$
10: **end if**
11: **end while**

The problem R_h, solved at each iteration of Algorithm 1, is nonconvex since constraints (23) are of the form $f(q_i, q_j) \geq d^2$ with $f(q_i, q_j)$ convex. Therefore, in order to find global optima of R_h, a global optimization algorithm should be employed. This, however, would make the CG algorithm excessively slow. In our implementation (see Sect. 6) we chose to heuristically solve R_h using a multistart algorithm calling a local NLP solver, from randomly chosen starting points, when global optimization solvers are too slow.

Note that, in Step 4, τ_{ij}^h, easily computed in closed form, is the instant for which the distance between i and j is minimum. If this distance is greater than or equal to the safety value for each pair of aircraft, the algorithm terminates at Step 6, as q^* must be an optimal solution of the bilevel formulation.

5.1.1 Cut dominance

The time per iteration taken by Algorithm 1 increases with the number of cuts added to the formulation. In fact, while solutions of the lower-level subproblems are easily computed in closed form at Step 4, increasing the number of quadratic constraints (23) yields a time increase when solving R_h. It would therefore be desirable to remove as many unnecessary cuts as possible. Consequently, we consider dominance relationships between cuts.

In particular, at each iteration $h > 1$, given a pair (i, j), we only wish to consider non-dominated cuts from Eq. (23) as constraints of our relaxed problem R_h. Let us define, for any $h' \in \{1, \ldots, h-1\}$,

$$m := \mathsf{max}\{\tau_{ij}^h, \tau_{ij}^{h'}\}, \quad p := \mathsf{min}\{\tau_{ij}^h, \tau_{ij}^{h'}\},$$

with $m \geq p \geq 0$. We aim at finding a dominance relationship among the following two cuts:

$$\sum_{k\in K} \left[(x^0_{ik} - x^0_{jk})^2 + m^2\psi^2_{ijk} + 2m(x^0_{ik} - x^0_{jk})\psi_{ijk}\right]$$
$$\sum_{k\in K} \left[(x^0_{ik} - x^0_{jk})^2 + p^2\psi^2_{ijk} + 2p(x^0_{ik} - x^0_{jk})\psi_{ijk}\right].$$

First, we remove the nonnegative term $\sum\limits_{k\in K}(x^0_{ik} - x^0_{jk})^2$ from both cuts and just compare

$$\sum_{k\in K} \left[m^2\psi^2_{ijk} + 2m(x^0_{ik} - x^0_{jk})\psi_{ijk}\right], \quad \text{and} \quad \sum_{k\in K} \left[p^2\psi^2_{ijk} + 2p(x^0_{ik} - x^0_{jk})\psi_{ijk}\right].$$

Moreover, notice that

$$\sum_{k\in K} m^2\psi^2_{ijk} \geq \sum_{k\in K} p^2\psi^2_{ijk}$$

holds, what with m^2 and p^2 being both multiplied by the same nonnegative term $\sum_{k\in K}\psi^2_{ijk}$.

Different scenarios arise depending on the relationship among the terms of the cut expressions. Either the cut involving p dominates the cut involving m, namely

$$\sum_{k\in K} \left[m^2\psi^2_{ijk} + 2m(x^0_{ik} - x^0_{jk})\psi_{ijk}\right] \geq \sum_{k\in K} \left[p^2\psi^2_{ijk} + 2p(x^0_{ik} - x^0_{jk})\psi_{ijk}\right],$$

or the cut involving m dominates the cut involving p, namely

$$\sum_{k\in K} \left[m^2\psi^2_{ijk} + 2m(x^0_{ik} - x^0_{jk})\psi_{ijk}\right] \leq \sum_{k\in K} \left[p^2\psi^2_{ijk} + 2p(x^0_{ik} - x^0_{jk})\psi_{ijk}\right].$$

If, for each feasible q_i and q_j,

$$\sum_{k\in K}(x^0_{ik} - x^0_{jk})\psi_{ijk} \geq 0 \tag{24}$$

then

$$\sum_{k\in K} 2m(x^0_{ik} - x^0_{jk})\psi_{ijk} \geq \sum_{k\in K} 2p(x^0_{ik} - x^0_{jk})\psi_{ijk}$$

also holds for each $q_i, q_j \in [q^{\mathsf{min}}, q^{\mathsf{max}}]$, since $m \geq p$ by definition.

However, if Eq. (24) is true for all $q_i, q_j \in [q^{\mathsf{min}}, q^{\mathsf{max}}]$ (i.e., the cut involving p dominates the one involving m), then $\tau^h_{ij} \leq 0$ for all h. In fact, τ^h_{ij} has a nonpositive numerator in Eq. (22) when Eq. (24) holds. This corresponds to instances where aircraft i and j do not collide in the time horizon $(0, T]$, which obviously makes this dominance rule moot.

If, for each feasible q_i and q_j,

$$\sum_{k\in K}(x^0_{ik}-x^0_{jk})\psi_{ijk}<0, \tag{25}$$

we can analyse the relationship between cuts by comparing the positive term $(m^2-p^2)\sum_{k\in K}\psi^2_{ijk}$, and the negative term $2(m-p)\sum_{k\in K}(x^0_{ik}-x^0_{jk})\psi_{ijk}$. In general we cannot infer anything about the above mentioned relationship, unless we consider specific subclasses of the problem. Therefore, although τ^h_{ij} is nonnegative in this case, it is not possible to detect the existence of dominances among cuts.

5.2 Cut generation algorithm for ADP via HAC

We propose a tailored version of the CG algorithm for the bilevel formulation Eq. (14a)-(14c), which models the HACADP. In this case, the nonconvex problem R_1 solved at the first iteration is

$$\begin{aligned}\min_{\theta}\quad & \sum_{i\in A}\theta_i^2\\ \forall i\in A\quad & \theta_i^{\mathsf{min}}\le\theta_i\le\theta_i^{\mathsf{max}}.\end{aligned}$$

Algorithm 2 CG algorithm for HACADP

Let $h=1$. Initialize the relaxation R_h of the bilevel program, obtained by considering the upper-level problem only.
while true **do**
  Solve R_h to obtain the optimal solution θ^*.
  For each aircraft pair (i,j), compute the instant $\tau^h_{ij}\in[0,T]$ as

$$\tau^h_{ij}=-\frac{(x^0_i-x^0_j)c^*_{ij}+(y^0_i-y^0_j)s^*_{ij}}{(c^*_{ij})^2+(s^*_{ij})^2}, \tag{26}$$

  with $c^*_{ij}=\cos(\phi_i+\theta^*_i)-\cos(\phi_j+\theta^*_j)$ and $s^*_{ij}=\sin(\phi_i+\theta^*_i)-\sin(\phi_j+\theta^*_j)$.
  if $\left[(x^0_i-x^0_j)+\tau^h_{ij}c^*_{ij}\right]^2+\left[(y^0_i-y^0_j)+\tau^h_{ij}s^*_{ij}\right]^2\ge d^2\quad\forall i<j\in A$ **then**
    The algorithm terminates and θ^* is the optimal solution of the bilevel formulation.
  else
    For each pair (i,j) violating the inequality, define R_{h+1} as R_h with the adjoined inequality:

$$\begin{aligned}&\left[(x^0_i-x^0_j)+\tau^h_{ij}(\cos(\phi_i+\theta_i)v_i-\cos(\phi_j+\theta_j)v_j)\right]^2\\ &+\left[(y^0_i-y^0_j)+\tau^h_{ij}(\sin(\phi_i+\theta_i)v_i-\sin(\phi_j+\theta_j)v_j)\right]^2\ge d^2.\end{aligned} \tag{27}$$

    $h:=h+1$
  end if
end while

Again, the problem R_h, solved at each iteration of the algorithm, is non-convex since the constraints (27) are of the form $f(\theta_i, \theta_j) \geq d^2$ with $f(\theta_i, \theta_j)$ convex. We find θ^* in Step 3 using a global NLP solver or, when the time limit is exceeded, with a local NLP solver within a multistart procedure from randomly chosen starting points.

As in Algorithm 1, τ_{ij}^h indicates when the distance between i and j is minimized and it is always computed in closed form in Step 4. If this distance satisfies the safety threshold for each pair of aircraft, the algorithm terminates at Step 6.

The considerations carried out in Sec. 5.1.1 also apply to Algorithm 2: we can prove the existence of a dominance relation on the cuts only for those pairs that do not collide in the time horizon $(0, T]$.

6 Computational experiments

For the SRADP in k dimensions, we use a 3D generalization of the 2D instances tested in [7] (named *sphere instances* in Tab 1), where n aircraft are placed on a sphere of a given radius r — see Figure 4. We consider also instances in which aircraft move along straight 3D trajectories (named *non-sphere instances* in Tab 1), which intersect in at least $\frac{n}{2}$ conflict points.

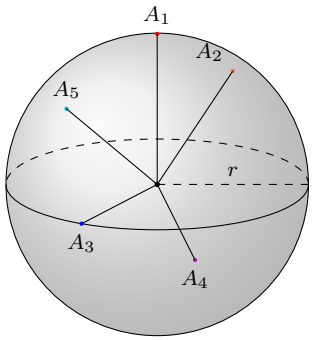

Fig. 4: n conflicting aircraft flying towards the center of a sphere

A trajectory is defined by two angles: the so-called pitch angle γ_i (angle that the vector of the direction u_i forms with the axis k_3) and the heading angle ϕ_i (angle between the projection of u_i onto the k_1k_2-plane and the axis k_1) — see Figure 5.

We test our approaches for the HACADP using the set of instances proposed in [17], where n aircraft are randomly placed on a circle of a given radius r. All aircraft speeds are initially set to the same value, and their trajectories are such that the aircraft fly exactly or almost exactly towards the center of the

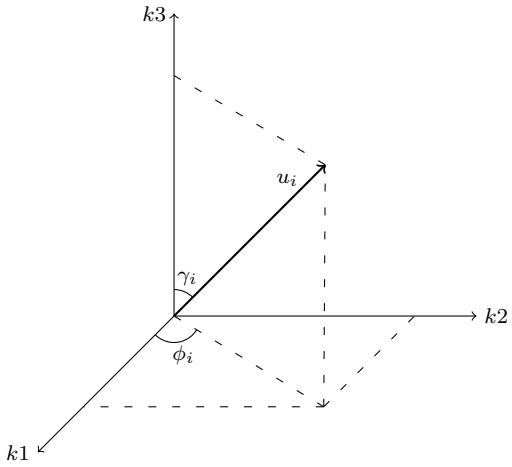

Fig. 5: The 3-dimensional airspace

circle — see Figure 6. These problems, characterized by an unrealistic highly symmetric configuration, are known in literature as *circle problems*.

We also consider instances, always from [17], in which aircraft are placed around a circle and have trajectories with a starting heading angle ϕ_i randomly chosen in $[-\frac{\pi}{6}, \frac{\pi}{6}]$ with respect to the diameter of the circle. The end point of each trajectory belongs to the circle as well. Note that these problems, named *random circle instances* in Tab 2, are more realistic than *circle problems* without deviation.

Moreover, we test some *non-circle instances* from [11] in which aircraft move along straight trajectories intersecting in n_c conflict points.

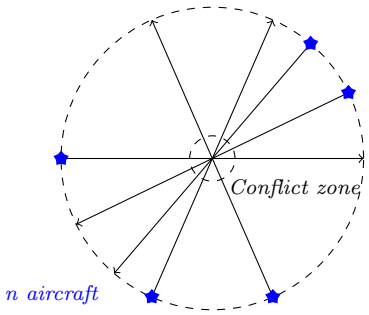

Fig. 6: n conflicting aircraft flying towards the center of a circle

In all the experiments, we consider a time horizon of $T = 2$ hours and standard safety distance $d = 5\,\text{NM}$. All the solvers are run with their default

settings. The tests are performed on a 2.53GHz Intel(R) Xeon(R) CPU with 49.4 GB RAM.

6.1 ADP in 3 dimensions via speed regulation

For the *sphere instances* the initial speed is $v_i = 400\,\text{NM/h}$ for each $i \in A$ and the angles γ_i and ϕ_i are randomly generated within $[0, \frac{\pi}{2}]$ and parameters x^0_{ik} and u_{ik} are given by

$$u_{i1} = \cos(\phi_i)\sin(\gamma_i), \quad u_{i2} = \sin(\phi_i)\sin(\gamma_i), \quad u_{i3} = \cos(\gamma_i), \quad x^0_{ik} = -r\,u_{ik}$$

where the sphere radius r is chosen in $\{100, 200, \ldots, 700\}$. The bounds q_i^{min} and q_i^{max} are set to 0.94 and 1.03 respectively, following the weaker bounds proposed by the ERASMUS project.

We implement the single-level formulations using the AMPL modeling language [28] and solve them with the global optimization solver Baron [29] (B in the Table 1). When Baron exceeds the time-limit (set to 3600 seconds), we use a Multistart algorithm (MS in Table 1), which performs 1000 calls to the SNOPT [30] local NLP solver from randomly sampled starting points.

We solve the bilevel formulation using the CG algorithm in Sect. 5 (CG in Table 1) with maximum iteration number set to 1000; at each iteration we solve the relaxed formulation R_h using the SNOPT [30] local NLP solver called 50 times within a Multistart procedure from randomly chosen points.

All the results are reported in Table 1. The headings are the following: n number of aircraft; r radius of the sphere in NM; *obj* best objective value found by each model; *cpu* computing time in seconds; *slv* solver used (for the CG algorithm we always use a Multistart method calling SNOPT to solve the inner problem R_h).

The value of the objective function is always very small, given the nature of the problem (q must be in $[0.94, 1.03]$). The tight speed variation bounds imposed by ERASMUS project lead to an additional complication since instances are not guaranteed to be feasible. Best objective values and minimum required time are reported in bold for each instance. The best formulation in terms of solution quality is the one in Eq. (6a)-(6h) based on KKT conditions of the lower-level subproblems. In terms of computational efficiency, for most of the instances the CG Alg. 1 is the best.

6.2 ADP in 2 dimensions via heading angles changes

As mentioned above, the HACADP instances are taken from [17,11]. The authors set $v_i = 400\,\text{NM/h}$ for each $i \in A$ for all the instances. For the *circle instances* the angles ϕ_i are randomly generated and parameters x^0_i and y^0_i are given by

$$x^0_i = -r\cos(\phi_i), \quad y^0_i = -r\sin(\phi_i).$$

For the *random circle instances* both the angles ϕ_i and the parameters x_i^0 and y_i^0 are randomly generated. The bounds θ_i^{min} and θ_i^{max} are set to $-\pi/6$ and $\pi/6$ respectively.

We again implement the formulations using the AMPL modeling language [28] and solve them with the global optimization solver Couenne [31] (C in the Table 2). We do not use Baron because it cannot handle the trigonometric functions sine and cosine. When Couenne exceeds the time-limit (set to 3600 seconds), we use a Multistart algorithm (MS in Table 2). It performs 1000 calls to SNOPT [30] for KKT reformulation Eq. (17a)–(17h) and Wolfe's dual reformulation Eq. (21a)–(21e), and 1000 calls to IPOPT [32] in the case of Dorn's dual reformulation Eq. (20a)–(20e). In all of the calls, starting points are randomly chosen.

We solve the bilevel formulation using the CG algorithm in Sect. 5 (CG in Table 2) with maximum iteration number set to 1000; at each iteration we solve the relaxed formulation R_h using Couenne, or, when the CG exceed the time-limit of 3600 seconds, the SNOPT [30] local NLP solver called 50 times within a Multistart procedure from randomly chosen points.

Our results are reported in Table 2. The headings are the following: name of the instance; n number of aircraft; n_c number of potential conflicts; *obj* best objective value found by each model; *cpu* computing time in seconds; *slv* solver used (in the last column, the solver used to solve the inner problem R_h of the CG algorithm). In Table 3 the results on *circle* and *random circle* instances are compared with those that are obtained using HAC only (without pre-processing) in [17]. Best objective values and minimum required time are reported in bold for each instance.

As shown in Table 2, among the models proposed in this paper, for most of the instances the CG algorithm is the best in terms of objective function and

Table 1: Results obtained solving 4 different formulations of SRADP

Instances		KKT reformulation			Dorn's dual reformul.			Wolfe's dual reformul.			CG	
n	r	*obj*	*cpu*	slv	*obj*	*cpu*	slv	*obj*	*cpu*	slv	*obj*	*cpu*
Sphere												
2	100	**0.0022197**	0.63	B	0.0022226	0.86	B	0.0022253	0.62	B	0.0022268	**0.16**
3	200	0.0014057	9.64	B	**0.0014036**	5.86	B	0.0014066	11.4	B	0.0014077	**0.61**
4	200	0.0037129	312	B	**0.0037034**	270	B	0.0037091	3582	B	0.0037142	**0.90**
5	300	0.0029590	5.90	MS	0.0029588	4.53	MS	0.0029590	11.7	MS	**0.0029586**	**1.67**
6	300	**0.0058469**	9.69	MS	**0.0058469**	7.41	MS	0.0058472	12.6	MS	0.0058750	**4.41**
7	500	**0.0028553**	15.3	MS	0.0028563	20.2	MS	0.0028563	26.4	MS	0.0028916	**10.2**
8	500	**0.0045492**	**16.8**	MS	0.00460517	35.6	MS	0.0045718	23.1	MS	0.0045989	22.7
9	500	**0.0069865**	**20.4**	MS	0.0072103	53.6	MS	0.0071089	28.1	MS	0.0071280	61.2
10	600	**0.0064098**	**32.2**	MS	0.0065400	80.9	MS	0.0064131	44.2	MS	0.0065062	242
12	700	0.0085109	79.4	MS	0.00868476	**57.4**	MS	**0.0084041**	81.4	MS	0.0088769	1063
Non-sphere												
2	-	**0.0003047**	**0.14**	B	0.0003049	0.17	B	0.0003049	0.26	B	0.0003049	0.44
4	-	**0.0032779**	54.9	B	0.0032827	5.19	MS	0.0037091	3584	B	0.0032821	**0.69**
6	-	0.0060038	14.5	MS	0.0060038	10.2	MS	0.0060038	16.2	MS	**0.0060032**	**0.75**
8	-	0.011705	19.0	MS	0.011936	28.5	MS	0.011705	17.9	MS	**0.011704**	**0.54**
10	-	**0.015025**	34.5	MS	**0.015025**	67.3	MS	**0.015025**	30.0	MS	**0.015025**	**2.30**

computational time, even when the inner problem R_h is solved using Couenne. Looking at the comparison of our results with the ones obtained in [17] in Table 3, it appears that they are comparable. Again the best solutions are obtained for most of the instances by the CG algorithm.

7 Conclusions

We propose bilevel programming as a suitable approach to model the well-known aircraft deconfliction problem or ADP. In particular, we present two bilevel formulations of the ADP: one based on speed regulation in k dimensions and another where potential conflicts are avoided via heading angle changes in two dimensions. In both cases, the convexity of the lower-level subproblems allows us to derive three different single-level problems respectively, using KKT conditions, Dorn's duality, and Wolfe's duality.

The single level reformulations of both problems are solved by using state-of-the-art solvers, which provide good solutions in reasonable computing time. Alternatively, we propose a cut generation algorithm to solve the bilevel problems, and theoretically discuss the existence of dominance relations among cuts that could improve the algorithm's performance. The proposed cut generation algorithm, compared with state-of-the-art solvers, obtains the best results for most of the tested instances in few seconds. Numerical results, when compared with other approaches in the literature, are encouraging and stress the potential of the proposed approach.

Table 2: Results obtained solving 4 different formulations of HACADP

Instances			KKT reformul.			Dorn's dual reformul.			Wolfe's dual reformul.			CG		
Name	n	n_c	*obj*	*cpu*	*slv*	*obj*	*cpu*	*slv*	*obj*	*cpu*	*slv*	*obj*	*cpu*	*slv*
Circle														
pb_n2	2	1	**0.001250**	0.15	C	**0.001250**	**0.14**	C	**0.001250**	0.18	C	**0.001250**	0.23	C
pb_n3_1	3	3	0.002501	0.83	C	0.002501	1.27	C	0.002501	0.83	C	**0.002500**	**0.12**	C
pb_n3_2	3	3	0.006672	7.83	C	0.006672	**5.82**	C	0.006657	7.89	C	**0.006650**	20.0	C
pb_n3_3	3	3	**0.000950**	**1.03**	C	**0.000950**	1.16	C	**0.000950**	1.26	C	**0.000950**	1.11	C
pb_n4_1	4	6	0.007240	**416**	C	0.007240	529	C	0.007228	1410	C	**0.007217**	661	C
pb_n4_2	4	6	**0.017061**	2497	C	0.017094	38.2	MS	0.017095	21.8	MS	0.017098	**1.18**	MS
pb_n4_3	4	6	0.001318	**25.3**	C	0.001318	85.2	C	0.001318	33.5	C	**0.001317**	36.9	C
pb_n5_1	5	10	0.011629	**25.3**	MS	0.011629	38.4	MS	0.011629	27.4	MS	**0.011591**	938	C
pb_n5_2	5	10	**0.018467**	28.1	MS	**0.018467**	37.7	MS	**0.018467**	30.0	MS	0.018482	**1.81**	MS
pb_n5_3	5	10	0.017122	23.6	MS	0.017121	40.8	MS	0.017122	27.7	MS	**0.017103**	**2.07**	MS
pb_n5_4	5	10	0.014750	22.9	MS	0.014750	40.5	MS	0.014750	27.6	MS	**0.014717**	**2.84**	MS
pb_n5_5	5	10	0.012163	**33.0**	MS	0.012163	53.4	MS	0.012163	35.0	MS	**0.012119**	1777	C
pb_n5_6	5	10	0.011235	36.7	MS	0.011235	111	MS	0.011235	38.2	MS	**0.011225**	**1.18**	MS
pb_n5_7	5	10	0.012273	31.6	MS	0.012273	42.3	MS	0.012274	37.2	MS	**0.012248**	**1.54**	MS
pb_n5_8	5	10	0.017556	36.0	MS	0.017556	42.0	MS	0.017556	39.5	MS	**0.017548**	**2.33**	MS
pb_n5_9	5	10	0.019140	29.1	MS	0.019140	46.2	MS	0.019140	32.9	MS	**0.019088**	**3.38**	MS
pb_n5_10	5	10	**0.026022**	20.9	MS	**0.026022**	42.3	MS	**0.026022**	23.6	MS	0.026122	**3.78**	MS
pb_n5_11	5	10	0.011202	42.4	MS	0.011202	38.2	MS	0.011202	37.4	MS	**0.011179**	**1.15**	MS
pb_n5_12	5	10	0.012122	23.7	MS	0.012121	39.0	MS	0.012122	25.3	MS	**0.012113**	**1.48**	MS
pb_n5_13	5	10	**0.023265**	32.1	MS	**0.023265**	43.9	MS	**0.023265**	33.4	MS	0.023269	**1.55**	MS
pb_n5_14	5	10	0.013789	28.1	MS	0.013789	40.6	MS	0.013789	32.7	MS	**0.013779**	**1.53**	MS
pb_n5_15	5	10	0.014578	32.9	MS	0.014578	51.4	MS	0.014578	35.3	MS	**0.014543**	**1.22**	MS
pb_n5_16	5	10	0.010378	**31.9**	MS	0.010378	93.8	MS	0.010378	35.8	MS	**0.010344**	2370	C
pb_n5_17	5	10	**0.011956**	34.9	MS	**0.011956**	38.8	MS	**0.011956**	39.2	MS	0.011985	**1.18**	MS
pb_n5_18	5	10	0.011167	**24.6**	MS	0.011167	32.1	MS	0.011167	27.5	MS	**0.011130**	473	C
pb_n5_19	5	10	0.009920	2498	C	0.009920	1613	C	0.009920	**36.9**	MS	**0.009892**	119	C
pb_n5_20	5	10	**0.019774**	27.6	MS	**0.019774**	40.7	MS	**0.019774**	29.2	MS	0.019834	**1.49**	MS
pb_n5_21	5	10	0.009050	1402	C	0.009050	1503	C	0.009051	61.8	MS	**0.009034**	**41.3**	C
pb_n5_22	5	10	0.030611	30.5	MS	**0.030610**	49.3	MS	0.030611	33.0	MS	0.030833	**4.32**	MS
pb_n5_23	5	10	0.001543	**27.4**	MS	0.001543	38.9	MS	0.001543	40.1	MS	**0.001541**	1831	C
pb_n6_1	6	15	0.001667	49.9	MS	0.001667	43.2	MS	0.001667	71.3	MS	**0.001664**	**1.86**	MS
pb_n6_2	6	15	0.001667	51.5	MS	0.001667	44.0	MS	0.001667	75.8	MS	**0.001664**	**1.31**	MS
Random Circle														
rpb2_1	2	1	**0.000141**	0.07	C	**0.000141**	**0.06**	C	**0.000141**	**0.06**	C	**0.000141**	0.07	C
rpb2_2	2	1	0.000795	0.07	C	0.000795	**0.05**	C	0.000795	0.06	C	**0.000793**	0.07	C
rpb3_1	3	2	**0.000078**	0.39	C	**0.000078**	2946	C	**0.000078**	0.29	C	**0.000078**	**0.18**	C
rpb3_2	3	2	0.000515	0.63	C	**0.000509**	9.9	C	0.000515	0.85	C	0.000514	**0.60**	C
rpb3_3	3	1	**0.000114**	0.11	C	**0.000114**	234	C	**0.000114**	0.11	C	**0.000114**	**0.06**	C
rpb4_1	4	1	**0.000156**	4.36	C	**0.000156**	4.70	C	**0.000156**	3.65	C	**0.000156**	**0.06**	C
rpb4_2	4	3	0.001177	13.3	C	0.001177	24.7	MS	0.001177	28.6	C	**0.001174**	**0.82**	C
rpb4_3	4	2	**0.000202**	1.55	C	**0.000202**	3.39	C	**0.000202**	6.39	C	**0.000202**	**0.17**	C
rpb5_1	5	6	0.000413	356	C	0.000413	39.3	MS	0.000413	319	C	**0.000412**	**2.07**	C
rpb5_2	5	3	0.000454	1323	C	0.000454	40.7	MS	0.000454	36.4	MS	**0.000452**	**4.54**	C
rpb5_3	5	8	0.000612	213	C	0.000612	39.2	MS	0.000612	171	C	**0.000611**	**4.58**	C
rpb6_1	6	5	0.000965	**44.2**	MS	0.000965	48.1	MS	0.000965	58.3	MS	**0.000962**	283	C
rpb6_2	6	9	**0.000858**	50.7	MS	**0.000858**	45.4	MS	**0.000858**	58.8	MS	**0.000858**	**5.68**	C
rpb6_3	6	4	0.000694	44.0	MS	0.000694	49.5	MS	0.000694	57.9	MS	**0.000693**	**34.0**	C
rpb7_1	7	2	0.002199	50.7	MS	**0.000213**	60.6	MS	**0.000213**	96.0	MS	**0.000213**	**0.39**	C
rpb7_2	7	7	0.001161	64.5	MS	0.001161	64.8	MS	0.001161	89.0	MS	**0.001159**	1629	C
rpb7_3	7	9	0.002794	66.4	MS	0.001914	58.1	MS	0.001914	99.8	MS	**0.001913**	**1.61**	MS
rpb8_1	8	12	0.001174	99.1	MS	0.001174	82.9	MS	0.001174	93.8	MS	**0.001173**	**2.23**	MS
rpb8_2	8	5	0.000370	102	MS	0.000370	**79.6**	MS	0.000370	97.2	MS	**0.000369**	93.5	C
rpb8_3	8	10	**0.001006**	97.8	MS	0.001007	84.3	MS	**0.001006**	98.7	MS	0.001529	**3.94**	MS
Non-Circle														
pb6_5	6	5	0.128250	59.3	MS	0.128249	**40.4**	MS	0.128250	146	MS	**0.127742**	171	C
pb7_4	7	4	0.001628	219	MS	0.001602	71.7	MS	0.001602	327	MS	**0.001599**	**0.86**	MS
pb7_6	7	6	0.001567	222	MS	0.001567	75.6	MS	0.001567	324	MS	**0.001565**	**0.60**	MS
pb8_4	8	4	0.002401	108	MS	0.002384	81.6	MS	0.002384	261	MS	**0.002382**	**0.50**	MS
pb10_10	10	10	0.127695	166	MS	0.127693	90.9	MS	0.127695	270	MS	**0.127525**	**2.08**	C

Table 3: Results obtained solving 4 different formulations of HACADP compared with those obtained in [17]

Instances			*obj*				
Name	n	n_c	KKT reformul.	Dorn's dual reformul.	Wolfe's dual reformul.	CG	[17]
Circle							
pb_n2	2	1	**0.001250**	**0.001250**	**0.001250**	**0.001250**	**0.001250**
pb_n3_1	3	3	0.002501	0.002501	0.002501	**0.002500**	0.002501
pb_n3_2	3	3	0.006672	0.006672	0.006657	**0.006650**	0.006665
pb_n3_3	3	3	**0.000950**	**0.000950**	**0.000950**	**0.000950**	**0.000950**
pb_n4_1	4	6	0.007240	0.007240	0.007228	**0.007217**	0.007240
pb_n4_2	4	6	**0.017061**	0.017094	0.017095	0.017098	0.017065
pb_n4_3	4	6	0.001318	0.001318	0.001318	**0.001317**	0.001318
pb_n5_1	5	10	0.011629	0.011629	0.011629	**0.011591**	0.011629
pb_n5_2	5	10	**0.018467**	**0.018467**	**0.018467**	0.018482	0.018468
pb_n5_3	5	10	0.017122	0.017121	0.017122	0.017103	**0.017100**
pb_n5_4	5	10	0.014750	0.014750	0.014750	**0.014717**	0.014750
pb_n5_5	5	10	0.012163	0.012163	0.012163	**0.012119**	0.012149
pb_n5_6	5	10	0.011235	0.011235	0.011235	**0.011225**	**0.011225**
pb_n5_7	5	10	0.012273	0.012273	0.012274	**0.012248**	0.012262
pb_n5_8	5	10	0.017556	0.017556	0.017556	**0.017548**	0.017556
pb_n5_9	5	10	0.019140	0.019140	0.019140	**0.019088**	0.019119
pb_n5_10	5	10	0.026022	0.026022	0.026022	0.026122	**0.025960**
pb_n5_11	5	10	0.011202	0.011202	0.011202	**0.011179**	0.011190
pb_n5_12	5	10	0.012122	0.012121	0.012122	0.012113	**0.012111**
pb_n5_13	5	10	**0.023265**	**0.023265**	**0.023265**	0.023269	**0.023265**
pb_n5_14	5	10	0.013789	0.013789	0.013789	**0.013779**	0.013790
pb_n5_15	5	10	0.014578	0.014578	0.014578	**0.014543**	0.014551
pb_n5_16	5	10	0.010378	0.010378	0.010378	**0.010344**	0.010367
pb_n5_17	5	10	0.011956	0.011956	0.011956	0.011985	**0.011940**
pb_n5_18	5	10	0.011167	0.011167	0.011167	**0.011130**	0.011153
pb_n5_19	5	10	0.009920	0.009920	0.009920	**0.009892**	0.009920
pb_n5_20	5	10	0.019774	0.019774	0.019774	0.019834	**0.019739**
pb_n5_21	5	10	0.009050	0.009050	0.009051	**0.009034**	0.009051
pb_n5_22	5	10	0.030611	0.030610	0.030611	0.030833	**0.030577**
pb_n5_23	5	10	0.001543	0.001543	0.001543	**0.001541**	0.001543
pb_n6_1	6	15	0.001667	0.001667	0.001667	0.001664	**0.001649**
pb_n6_2	6	15	0.001667	0.001667	0.001667	0.001664	**0.001661**
Random Circle							
rpb2_1	2	1	**0.000141**	**0.000141**	**0.000141**	**0.000141**	**0.000141**
rpb2_2	2	1	0.000795	0.000795	0.000795	**0.000793**	0.000795
rpb3_1	3	2	**0.000078**	**0.000078**	**0.000078**	**0.000078**	**0.000078**
rpb3_2	3	2	0.000515	**0.000509**	0.000515	0.000514	0.000513
rpb3_3	3	1	0.000114	0.000114	0.000114	0.000114	**0.000113**
rpb4_1	4	1	**0.000156**	**0.000156**	**0.000156**	**0.000156**	**0.000156**
rpb4_2	4	3	0.001177	0.001177	0.001177	**0.001174**	0.001175
rpb4_3	4	2	**0.000202**	**0.000202**	**0.000202**	**0.000202**	**0.000202**
rpb5_1	5	6	0.000413	0.000413	0.000413	0.000412	**0.000408**
rpb5_2	5	3	0.000454	0.000454	0.000454	0.000452	**0.000450**
rpb5_3	5	8	0.000612	0.000612	0.000612	**0.000611**	0.000613
rpb6_1	6	5	0.000965	0.000965	0.000965	0.000962	**0.000955**
rpb6_2	6	9	0.000858	0.000858	0.000858	0.000858	**0.000855**
rpb6_3	6	4	0.000694	0.000694	0.000694	**0.000693**	**0.000693**
rpb7_1	7	2	0.002199	0.000213	0.000213	0.000213	**0.000210**
rpb7_2	7	7	0.001161	0.001161	0.001161	**0.001159**	0.001162
rpb7_3	7	9	0.002794	0.001914	0.001914	**0.001913**	0.002637
rpb8_1	8	12	0.001174	0.001174	0.001174	**0.001173**	0.001189
rpb8_2	8	5	0.000370	0.000370	0.000370	**0.000369**	0.000373
rpb8_3	8	10	**0.001006**	0.001007	**0.001006**	0.001529	0.001019

References

1. J. Villiers: Automatisation du controÎle de la circulation aèrienne: "ERASMUS", une voie conviviale pour franchir le mur de la capacitè, *Institut du Transport Aèrien* **58** (2004)
2. F. Drogoul, P. Averty, R. Weber: ERASMUS Strategic Deconfliction to Benefit SESAR, *Proc. 8th USA/Europe Air Traffic Management Research and Development Seminar, Napa, USA*, 1–10 (2009)
3. https://ec.europa.eu/transport/modes/air/sesar_en
4. M. Lao, J. Tang: Cooperative Multi-UAV Collision Avoidance Based on Distributed Dynamic Optimization and Causal Analysis, *Applied Sciences* **7(1)**, 83 (2017)
5. D. Claes, K. Tuyls: Multi robot collision avoidance in a shared workspace, *Autonomous Robots* **42(8)**, 1749–1770 (2018)
6. C. G. Co, J.M.A. Tanchoco, A review of research on AGVS vehicle management, *Engineering Costs and Production Economics* **21(1)**, 35–42 (1991)
7. M. Cerulli, C. D'Ambrosio, L. Liberti: On aircraft deconfliction by bilevel programming, *Advances in Optimization and Decision Science for Society, Services and Enterprises: ODS, Genoa, Italy, September 4-7, 2019* – AIRO Springer Serie **3**, 197–206 (2019)
8. D. Rey, C. Rapine, S. Constans, R. Fondacci: A Mixed Integer Linear Model for Potential Conflict Minimization by Speed Modulations, *ICRAT 2010. Fourth International Conference on Research in Air Transportation* (2010)
9. D. Rey, C. Rapine, V.V. Dixit, S. T. Waller: Equity-Oriented Aircraft Collision Avoidance Model, *IEEE Transactions on Intelligent Transportation Systems* **16(1)**, 172–183 (2015)
10. S. Cafieri, N. Durand: Aircraft Deconfliction with Speed Regulation: New Models from Mixed-integer Optimization, *Journal of Global Optimization* **58(4)**, 613–629 (2014)
11. S. Cafieri, C. D'Ambrosio: Feasibility pump for aircraft deconfliction with speed regulation, *Journal of Global Optimization* **71(3)**, 501–515 (2018)
12. A. Vela, S. Solak, W. Singhose and J. Clarke: A mixed integer program for flight-level assignment and speed control for conflict resolution, *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference, Shanghai*, 5219–5226 (2009)
13. A. Alonso-Ayuso, L. F. Escudero, F. J. Martín-Campo: Collision avoidance in air traffic management: A mixed-integer linear optimization approach, *IEEE Transactions on Intelligent Transportation Systems* **12(1)**, 47–57 (2010)
14. F.H. Dias, H. Hijazi, D. Rey: Disjunctive linear separation conditions and mixed-integer formulations for aircraft conflict resolution by speed and altitude control, *arXiv preprint arXiv:1911.12997* (2019)
15. K. D. Bilimoria: A Geometric Optimization Approach to Aircraft Conflict Resolution, *18th Applied Aerodynamics Conference*, (2000)
16. L. Pallottino, E. Feron, A. Bicchi: Conflict resolution problems for air traffic management systems solved with mixed integer programming, *IEEE transactions on intelligent transportation systems* **3(1)**, 3–11 (2002)
17. S. Cafieri, R. Omheni: Mixed-integer nonlinear programming for aircraft conflict avoidance by sequentially applying velocity and heading angle changes, *European Journal of Operational Research* **260(1)**, 283–290 (2017)
18. E. Frazzoli, Z.-H. Mao, J.-H. Oh, E. Feron: Resolution of Conflicts Involving Many Aircraft via Semidefinite Programming, *Journal of Guidance, Control and Dynamic* **24(1)**, 79–86 (2001)
19. D. Rey, H. Hijazi: Complex number formulation and convex relaxations for aircraft conflict resolution, *IEEE 56th Annual Conference on Decision and Control*, 88–93 (2017)
20. O. Stein: How to solve a semi-infinite optimization problem, *European Journal of Operational Research* **223**, 312—320 (2012)
21. S. Dempe, V. Kalashnikov, G.A. Prez-Valds, N. Kalashnykova: Bilevel Programming Problems: Theory, Algorithms and Applications to Energy Networks – Springer, Berlin (2015)
22. T. Kleinert, M. Labbé, F. Plein, M. Schmidt: There's No Free Lunch: On the Hardness of Choosing a Correct Big-M in Bilevel Optimization, *Optimization online* **7172**, (2019)

23. W. S. Dorn: Self-Dual Quadratic Programs, *Journal of the Society for Industrial and Applied Mathematics* **9(1)**, 51–54 (1961)
24. W. S. Dorn: Duality in quadratic programming, *Quarterly of Applied Mathematics* **18(2)**, 155–162 (1960)
25. P. Wolfe: A duality theorem for non-linear programming, *Quarterly of Applied Mathematics* **19**, 239–244 (1961)
26. A. Alonso-Ayuso, L. F. Escudero, F. J. Martín-Campo: Exact and Approximate Solving of the Aircraft Collision Resolution Problem via Turn Changes, *Transportation Science* **50**, 263–274 (2016)
27. S. Fang, C. Lin, S. Wu: Solving quadratic semi-infinite programming problems by using relaxed cutting-plane scheme, *Journal of Computational and Applied Mathematics* **129**, 89—104 (2001)
28. R. Fourer, D. M. Gay, B. W. Kernighan: AMPL: A Modeling Language for Mathematical Programming, *Cengage Learning* (2002)
29. N.V. Sahinidis and M. Tawarmalani: *BARON 7.2.5: Global Optimization of Mixed-Integer Nonlinear Programs*, User's Manual (2005)
30. P.E. Gill: *User's guide for SNOPT version 7.2* (2006)
31. P. Belotti, J. Lee, L. Liberti and A. Wächter: Branching and bounds tightening techniques for non-convex MINLP, *Optimization Methods and Software* **24(4-5)**, 597–634, (2009)
32. A. Wächter and L. Biegler: On the Implementation of a Primal-Dual Interior Point Filter Line Search Algorithm for Large-Scale Nonlinear Programming, *Mathematical Programming* (2006)