# Detecting and solving aircraft conflicts using bilevel programming <br> Martina Cerulli, Claudia d'Ambrosio, Leo Liberti, Mercedes Pelegrín 

## To cite this version:

Martina Cerulli, Claudia d'Ambrosio, Leo Liberti, Mercedes Pelegrín. Detecting and solving aircraft conflicts using bilevel programming. 2020. hal-02869699v1

HAL Id: hal-02869699 https://hal.science/hal-02869699v1<br>Preprint submitted on 16 Jun 2020 (v1), last revised 9 Dec 2020 (v3)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# 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.

[^0]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 11). 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 singlelevel 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;
$-K=\left\{1, \ldots, k_{\max }\right\}$ 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] ${ }^{1}$,
- $x_{i k}^{0}$ is the $k$-th component of the initial position of aircraft $i$;
$-v_{i}$ is the initially planned speed of aircraft $i[\mathrm{NM} / \mathrm{h}]$;
- $u_{i k}$ 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. $q_{i}^{\min }<1<q_{i}^{\max }$.
- 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]3.1 Natural problem formulation

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

$$
\begin{align*}
\min _{q} & \sum_{i \in A}\left(q_{i}-1\right)^{2}  \tag{1a}\\
\forall i \in A & q_{i}^{\min } \leq q_{i} \leq q_{i}^{\max }  \tag{1b}\\
\forall i<j \in A, t \in[0, T] & \sum_{k \in K}\left[\left(x_{i k}^{0}-x_{j k}^{0}\right)+t\left(q_{i} v_{i} u_{i k}-q_{j} v_{j} u_{j k}\right)\right]^{2} \geq d^{2} . \tag{1c}
\end{align*}
$$

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 $\left[q_{i}^{\min }, q_{i}^{\max }\right]$ 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 semiinfinite 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:

$$
\begin{align*}
\min _{q, t} & \sum_{i \in A}\left(q_{i}-1\right)^{2}  \tag{2a}\\
\forall i \in A & q_{i}^{\min } \leq q_{i} \leq q_{i}^{\max }  \tag{2b}\\
\forall i<j \in A & \min _{t_{i j} \in[0, T]} \sum_{k \in K}\left[\left(x_{i k}^{0}-x_{j k}^{0}\right)+t_{i j}\left(q_{i} v_{i} u_{i k}-q_{j} v_{j} u_{j k}\right)\right]^{2} \geq d^{2} . \tag{2c}
\end{align*}
$$

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

$$
\begin{align*}
\min _{q, \tau} & \sum_{i \in A}\left(q_{i}-1\right)^{2}  \tag{3a}\\
\forall i \in A & q_{i}^{\min } \leq q_{i} \leq q_{i}^{\max }  \tag{3b}\\
\forall i<j \in A & \tau_{i j} \in \underset{t_{i j} \in[0, T]}{\arg \min } \sum_{k \in K}\left[\left(x_{i k}^{0}-x_{j k}^{0}\right)+t_{i j}\left(q_{i} v_{i} u_{i k}-q_{j} v_{j} u_{j k}\right)\right]^{2}  \tag{3c}\\
\forall i<j \in A & \sum_{k \in K}\left[\left(x_{i k}^{0}-x_{j k}^{0}\right)+\tau_{i j}\left(q_{i} v_{i} u_{i k}-q_{j} v_{j} u_{j k}\right)\right]^{2} \geq d^{2} \tag{3~d}
\end{align*}
$$

An optimal solution of each lower level subproblem, denoted by $\tau_{i j}$, corresponds to the time instant at which aircraft $i$ and $j$ are closest. Note that each lowerlevel subproblem is an optimization problem on the lower-level variables $t_{i j}$, 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:

$$
\begin{equation*}
\varphi_{i j}(q):=\min _{t_{i j} \in[0, T]} \sum_{k \in K}\left[\left(x_{i k}^{0}-x_{j k}^{0}\right)+t_{i j}\left(q_{i} v_{i} u_{i k}-q_{j} v_{j} u_{j k}\right)\right]^{2} \tag{4}
\end{equation*}
$$

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

$$
\begin{align*}
\min _{q, \tau} & \sum_{i \in A}\left(q_{i}-1\right)^{2}  \tag{5a}\\
\forall i \in A & q_{i}^{\min } \leq q_{i} \leq q_{i}^{\max }  \tag{5b}\\
\forall i<j \in A & \sum_{k \in K}\left[\left(x_{i k}^{0}-x_{j k}^{0}\right)+\tau_{i j}\left(q_{i} v_{i} u_{i k}-q_{j} v_{j} u_{j k}\right)\right]^{2} \leq \varphi_{i j}(q)  \tag{5c}\\
\forall i<j \in A & \sum_{k \in K}\left[\left(x_{i k}^{0}-x_{j k}^{0}\right)+\tau_{i j}\left(q_{i} v_{i} u_{i k}-q_{j} v_{j} u_{j k}\right)\right]^{2} \geq d^{2} . \tag{5~d}
\end{align*}
$$

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

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

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

$$
\begin{array}{rl}
\min _{q, t, \mu, \lambda} & \sum_{i \in A}\left(q_{i}-1\right)^{2} \\
\forall i \in A & q_{i}^{\min } \leq q_{i} \leq q_{i}^{\max } \\
\forall i<j \in A & \sum_{k \in K}\left[2 t_{i j} \psi_{i j k}^{2}+2\left(x_{i k}^{0}-x_{j k}^{0}\right) \psi_{i j k}-\mu_{i j}+\lambda_{i j}\right]=0 \\
\forall i<j \in A & \mu_{i j}, \lambda_{i j} \geq 0 \\
\forall i<j \in A & \mu_{i j} t_{i j}=0 \\
\forall i<j \in A & \lambda_{i j} t_{i j}-\lambda_{i j} T=0 \\
\forall i<j \in A & 0 \leq t_{i j} \leq T \\
\forall i<j \in A & \sum_{k \in K}\left[\left(x_{i k}^{0}-x_{j k}^{0}\right)+t_{i j} \psi_{i j k}\right]^{2} \geq d^{2} \tag{6h}
\end{array}
$$

where the symbol $\psi_{i j k}$ appearing in Eq. (6c) and 6h), and defined as:

$$
\begin{equation*}
\psi_{i j k}:=q_{i} v_{i} u_{i k}-q_{j} v_{j} u_{j k} \tag{7}
\end{equation*}
$$

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. 6 d and 6 g ) 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 $\mu, \lambda$. 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:

$$
\begin{equation*}
\min _{y}\left\{\left.\frac{1}{2} y^{\top} Q_{x} y+p_{x}^{\top} y+c_{x} \right\rvert\, A y \geq b\right\} \geq \text { const } \tag{8}
\end{equation*}
$$

with $Q_{x}$ positive semidefinite.

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

$$
\begin{align*}
& \max _{z}\{\operatorname{LowerDualObj}(x, z) \mid \text { LowerDualConstr }(x, z)\} \\
= & \min _{y}\left\{\left.\frac{1}{2} y^{\top} Q_{x} y+p_{x}^{\top} y+c_{x} \right\rvert\, A y \geq b\right\}, \tag{9}
\end{align*}
$$

where LowerDualObj $(x, z)$ and 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:

$$
\begin{equation*}
\max _{z}\{\operatorname{LowerDualObj}(x, z) \mid \text { LowerDualConstr }(x, z)\} \geq \text { const } \tag{10}
\end{equation*}
$$

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:

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

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 $\operatorname{LowerDual} \operatorname{Obj}(x, y)$ attains its maximum over its feasible region, defined by 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) $-(2 \mathrm{c})$, 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 $g_{i j}$ and $z_{i j}$ of each lower-level subproblem in the left hand side of Eq. 2c) (defined for constraints $-t_{i j} \leq 0$ and $t_{i j} \leq T$ respectively),
using Dorn's dual [23, 24] and Prop. 2, the following reformulation of (2a)-2c) follows:

$$
\begin{align*}
\min _{q, g, z} & \sum_{i \in A}\left(q_{i}-1\right)^{2}  \tag{12a}\\
\forall i \in A & q_{i}^{\min } \leq q_{i} \leq q_{i}^{\max }  \tag{12b}\\
\forall i<j \in A & -\sum_{k \in K} \psi_{i j k}^{2} g_{i j}^{2}-T z_{i j} \geq d^{2}-\sum_{k \in K}\left(x_{i k}^{0}-x_{j k}^{0}\right)^{2}  \tag{12c}\\
\forall i<j \in A & -\frac{z_{i j}}{2}-\sum_{k \in K} \psi_{i j k}^{2} g_{i j} \leq \sum_{k \in K}\left(x_{i k}^{0}-x_{j k}^{0}\right) \psi_{i j k}  \tag{12~d}\\
\forall i<j \in A & z_{i j} \geq 0, \tag{12e}
\end{align*}
$$

obtained by the application of Prop. 2 to replace the lower-level subproblems of Eq. (2a)-(2c) by their Dorn duals in the variables $g_{i j}, z_{i j}$ for each aircraft pair $i<j \in A$. This yields Eq. 12 c$)-(12 \mathrm{~d})$. Note that the primal lower-level variable $t_{i j}$ 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 $(P)$ :

$$
\left.\left.\begin{array}{rlrl}
\min _{y} \frac{1}{2} y^{\top} Q y+p^{\top} y  \tag{D}\\
A y & \geq b \\
y & \geq 0
\end{array}\right\} \quad(P) \quad \begin{array}{cr}
\max _{g, z}-\frac{1}{2} g^{\top} Q g+b^{\top} z \\
A^{\top} z-Q g \leq p \\
z & \geq 0
\end{array}\right\}
$$

In our case, we have:

- $y:=t_{i j}$,
- $Q:=2 \sum_{k \in K} \psi_{i j k}^{2}$,
- $p:=2 \sum_{k \in K}\left(x_{i k}^{0}-x_{j k}^{0}\right) \psi_{i j k}$,
- $A:=-1$,
- $b:=-T$.

Recall that $\psi_{i j k}$ is constant in the lower level since, by Eq. 77, it only depends on the upper-level variables $q_{i}$ and $q_{j}$. By easy replacements and Prop. 2 , with const $=d^{2}-\sum_{k \in K}\left(x_{i k}^{0}-x_{j k}^{0}\right)^{2}$ 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)

$$
\sum_{k \in K}\left[\left(x_{i k}^{0}-x_{j k}^{0}\right)+t_{i j} \psi_{i j k}\right]^{2}+\alpha_{i j}\left(t_{i j}-T\right)-\beta_{i j} t_{i j}
$$

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

$$
\begin{align*}
\min _{q, t, \alpha, \beta} & \sum_{i \in A}\left(q_{i}-1\right)^{2}  \tag{13a}\\
\forall i \in A & q_{i}^{\min } \leq q_{i} \leq q_{i}^{\max }  \tag{13b}\\
\forall i<j \in A & \sum_{k \in K}\left[\left(x_{i k}^{0}-x_{j k}^{0}\right)+t_{i j} \psi_{i j k}\right]^{2}+\alpha_{i j}\left(t_{i j}-T\right)-\beta_{i j} t_{i j} \geq d^{2}  \tag{13c}\\
\forall i<j \in A & \sum_{k \in K}\left[2 t_{i j} \psi_{i j k}^{2}+2\left(x_{i k}^{0}-x_{j k}^{0}\right) \psi_{i j k}+\alpha_{i j}-\beta_{i j}\right]=0  \tag{13d}\\
\forall i<j \in A & \alpha_{i j}, \beta_{i j} \geq 0 . \tag{13e}
\end{align*}
$$

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 $(P)$ :

$$
\left.\left.\begin{array}{rlr}
\min _{y} \frac{1}{2} y^{\top} Q y+p^{\top} y+c  \tag{D}\\
A y \geq b \\
y \geq 0
\end{array}\right\} \quad(P) \quad \begin{array}{rr}
\max _{g, z} \mathcal{L}(y, \alpha, \beta) \\
\frac{\partial \mathcal{L}}{\partial y}=0 \\
\alpha, \beta \geq 0
\end{array}\right\}
$$

with:

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

and

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

In our case, we have:

- $y:=t_{i j}$,
- $Q:=2 \sum_{k \in K} \psi_{i j k}^{2}$,
- $p:=2 \sum_{k \in K}\left(x_{i k}^{0}-x_{j k}^{0}\right) \psi_{i j k}$,
- $c:=\sum_{k \in K}\left(x_{i k}^{0}-x_{j k}^{0}\right)^{2}$,
- $A:=-1$,
- $b:=-T$.

Again, we recall that $\psi_{i j k}$ 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 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:

- $\phi_{i}$ 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 $i$
$-\theta_{i}^{\min }$ and $\theta_{i}^{\max }$ are the bounds on heading angle variation, with $\theta_{i}^{\min }<0$ and $\theta_{i}^{\max }>0$
- Variables: $\theta_{i}$ 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 $\theta_{i}$ (for all $i \in A$ ) and the lower-level decision variables are $t_{i j}$ (for all $i<j \in A$ ). Each lower-level subproblem is parametrized by the upper-level variables $\theta_{i}, \theta_{j}$.

$$
\begin{align*}
\min _{\theta, t} & \sum_{i \in A} \theta_{i}^{2}  \tag{14a}\\
\forall i \in A & \theta_{i}^{\min } \leq \theta_{i} \leq \theta_{i}^{\max }  \tag{14b}\\
\forall i<j \in A & \min _{t_{i j} \in[0, T]}\left[\left(x_{i}^{0}-x_{j}^{0}\right)+t_{i j}\left(\cos \left(\phi_{i}+\theta_{i}\right) v_{i}-\cos \left(\phi_{j}+\theta_{j}\right) v_{j}\right)\right]^{2} \\
& \quad+\left[\left(y_{i}^{0}-y_{j}^{0}\right)+t_{i j}\left(\sin \left(\phi_{i}+\theta_{i}\right) v_{i}-\sin \left(\phi_{j}+\theta_{j}\right) v_{j}\right)\right]^{2} \geq d^{2} . \tag{14c}
\end{align*}
$$

The convex objective function of the upper level is the sum of squared heading angle changes, which are bounded by $\left[\theta_{i}^{\min }, \theta_{i}^{\max }\right]$. The objective of
each lower-level subproblem is to minimize the squared Euclidean distance between aircraft $i$ and $j$ over $t_{i j} \in[0, T]$. Its expression is obtained by defining the position $\left(x_{i}(t), y_{i}(t)\right)$ of aircraft $i$ at time $t$ as

$$
x_{i}(t)=x_{i}^{0}+\cos \left(\phi_{i}+\theta_{i}\right) v_{i} t \quad \text { and } \quad y_{i}(t)=y_{i}^{0}+\sin \left(\phi_{i}+\theta_{i}\right) v_{i} t .
$$

Note that the lower-level objective function is also convex in $t_{i j}$. 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. 14 a$)-(14 \mathrm{c})$. 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:

$$
\begin{equation*}
\min _{t_{i j}}\left\|\binom{\left(x_{i}^{0}-x_{j}^{0}\right)+t_{i j}\left(\cos \left(\phi_{i}+\theta_{i}^{\star}\right) v_{i}-\cos \left(\phi_{j}+\theta_{j}^{\star}\right) v_{j}\right)}{\left(y_{i}^{0}-y_{j}^{0}\right)+t_{i j}\left(\sin \left(\phi_{i}+\theta_{i}^{\star}\right) v_{i}-\sin \left(\phi_{j}+\theta_{j}^{\star}\right) v_{j}\right)}\right\|^{2} . \tag{15}
\end{equation*}
$$

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. 114 a$)-(14 \mathrm{c})$.

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

$$
\begin{equation*}
T_{i}^{\star}:=\max _{j: i \neq j} \tau_{i j}^{\star} \tag{16}
\end{equation*}
$$

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 $\tau_{i j}$ for every pair of aircraft $(i, j))$.

Knowing $\left(x_{i}\left(T_{i}^{\star}\right), y_{i}\left(T_{i}^{\star}\right)\right)$ 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 $\lambda_{i j}$ (resp. $\mu_{i j}$ ) associated to constraints $t_{i j} \leq T$ (resp. $-t_{i j} \leq 0$ ). Assuming some regularity conditions hold, the reformulation is exact since the lower level is convex in the variable $t_{i j}$ [21, Sec. 3.5].

$$
\begin{array}{rl}
\min _{\theta, t, \lambda, \mu} & \sum_{i \in A} \theta_{i}^{2} \\
\forall i \in A & \theta_{i}^{\min } \leq \theta_{i} \leq \theta_{i}^{\max } \\
\forall i<j \in A & 2 t_{i j}\left(c_{i j}^{2}+s_{i j}^{2}\right)+2\left(x_{i}^{0}-x_{j}^{0}\right) c_{i j}+2\left(y_{i}^{0}-y_{j}^{0}\right) s_{i j} \\
& +\lambda_{i j}-\mu_{i j}=0 \\
\forall i<j \in A & \lambda_{i j}, \mu_{i j} \geq 0 \\
\forall i<j \in A & \lambda_{i j} t_{i j}-\lambda_{i j} T=0 \\
\forall i<j \in A & \mu_{i j} t_{i j}=0 \\
\forall i<j \in A & 0 \leq t_{i j} \leq T \\
\forall i<j \in A & {\left[\left(x_{i}^{0}-x_{j}^{0}\right)+t_{i j} c_{i j}\right]^{2}+\left[\left(y_{i}^{0}-y_{j}^{0}\right)+t_{i j} s_{i j}\right]^{2} \geq d^{2},} \tag{17h}
\end{array}
$$

where the symbols $c_{i j}$ and $s_{i j}$ are shorthand for the following nonlinear expressions:

$$
\begin{align*}
c_{i j} & :=\cos \left(\phi_{i}+\theta_{i}\right) v_{i}-\cos \left(\phi_{j}+\theta_{j}\right) v_{j}  \tag{18}\\
s_{i j} & :=\sin \left(\phi_{i}+\theta_{i}\right) v_{i}-\sin \left(\phi_{j}+\theta_{j}\right) v_{j} . \tag{19}
\end{align*}
$$

The formulation in Eq. $17 \mathrm{a}-\sqrt{17 \mathrm{~h}}$ is a single-level Nonlinear Programming (NLP) problem in the variables $\theta, t, \lambda$, and $\mu$. Constraints Eq. 17 c correspond to stationarity conditions of the lower-level subproblems, Eq. (17g) to primal
feasibility, Eq. $(17 \mathrm{~d})$ to dual feasibility, Eq. 17 e and Eq. 17 f$)$ to complementarity conditions. We, again, require that the safety distance is satisfied for each pair of aircraft with constraints 17 h .

### 4.3 Dual reformulations

We follow the procedure discussed in Sect. 3.4 in order to obtain two dual reformulations of Eq. 114 a$)-(14 \mathrm{c})$. The first involves Dorn's dual of the lowerlevel subproblems, while the second involves Wolfe's dual.

### 4.3.1 Dorn's dual reformulation

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

$$
\begin{align*}
\min _{\theta, g, z} & \sum_{i \in A} \theta_{i}^{2}  \tag{20a}\\
\forall i \in A & \theta_{i}^{\min } \leq \theta_{i} \leq \theta_{i}^{\max }  \tag{20b}\\
\forall i<j \in A & -g_{i j}^{2}\left(c_{i j}^{2}+s_{i j}^{2}\right)-T z_{i j} \geq d^{2}-\left(x_{i}^{0}-x_{j}^{0}\right)^{2}-\left(y_{i}^{0}-y_{j}^{0}\right)^{2}  \tag{20c}\\
\forall i<j \in A & -\frac{z_{i j}}{2}-\left(c_{i j}^{2}+s_{i j}^{2}\right) g_{i j} \leq\left(x_{i}^{0}-x_{j}^{0}\right) c_{i j}+\left(y_{i}^{0}-y_{j}^{0}\right) s_{i j}  \tag{20d}\\
\forall i<j \in A & z_{i j} \geq 0 . \tag{20e}
\end{align*}
$$

The formulation in Eq. 20a -20e) is a single-level problem in the variables $\theta, g$, and $z$, the exactness of which is proved below. Note that the primal lower-level variable $t_{i j}$ 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 $(P)$ :

In our case, we have:

- $y:=t_{i j}$,
- $Q:=2\left(c_{i j}^{2}+s_{i j}^{2}\right)$,
- $p:=2\left(x_{i}^{0}-x_{j}^{0}\right) c_{i j}+2\left(y_{i}^{0}-y_{j}^{0}\right) s_{i j}$,
- $A:=-1$,
- $b:=-T$.

We recall that $c_{i j}$ and $s_{i j}$ are constant in the lower level because, by Eq. (18)(19), they only depend on the upper-level variables $\theta_{i}$ and $\theta_{j}$. By easy replacements and Prop. 2 with const $=d^{2}-\left(x_{i}^{0}-x_{j}^{0}\right)^{2}-\left(y_{i}^{0}-y_{j}^{0}\right)^{2}$ 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 $\alpha_{i j}$ and $\beta_{i j}$, we obtain the following single-level reformulation of of Eq. 14a)14c:

$$
\begin{array}{rl}
\min _{\theta, t, \alpha, \beta} & \sum_{i \in A} \theta_{i}^{2} \\
\forall i \in A & \theta_{i}^{\min } \leq \theta_{i} \leq \theta_{i}^{\max } \\
\forall i<j \in A & {\left[\left(x_{i}^{0}-x_{j}^{0}\right)+t_{i j} c_{i j}\right]^{2}+\left[\left(y_{i}^{0}-y_{j}^{0}\right)+t_{i j} s_{i j}\right]^{2}} \\
& +\alpha_{i j}\left(t_{i j}-T\right)-\beta_{i j} t_{i j} \geq d^{2} \\
\forall i<j \in A & 2 t_{i j}\left(c_{i j}^{2}+s_{i j}^{2}\right)+2\left(x_{i}^{0}-x_{j}^{0}\right) c_{i j}+2\left(y_{i}^{0}-y_{j}^{0}\right) s_{i j} \\
& +\alpha_{i j}-\beta_{i j}=0 \\
\forall i<j \in A & \alpha_{i j}, \beta_{i j} \geq 0 . \tag{21e}
\end{array}
$$

With Eq. 221 c ), 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 $\alpha_{i j}$ and $\beta_{i j}$. The exactness of formulation 21 a$)-21 \mathrm{e}$ 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 $(P)$ :

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

with:

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

and

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

In our case, we have:

- $y:=t_{i j}$,
- $Q:=2\left(c_{i j}^{2}+s_{i j}^{2}\right)$,
- $p:=2\left(x_{i}^{0}-x_{j}^{0}\right) c_{i j}+2\left(y_{i}^{0}-y_{j}^{0}\right) s_{i j}$,
- $c:=\left(x_{i}^{0}-x_{j}^{0}\right)^{2}+\left(y_{i}^{0}-y_{j}^{0}\right)^{2}$,
- $A:=-1$,
- $b:=-T$.

Again we recall that $c_{i j}$ and $s_{i j}$ are constant in the lower level because, by Eq. 18)-(19), they only depend on the upper-level variables $\theta_{i}$ and $\theta_{j}$. By easy replacements and Prop. 2, with 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:

$$
\begin{array}{ll}
\min _{q} & \sum_{i \in A}\left(q_{i}-1\right)^{2} \\
\forall i \in A & q_{i}^{\min } \leq q_{i} \leq q_{i}^{\max } .
\end{array}
$$

```
Algorithm 1 CG algorithm for SRADP
    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 \(q^{*}\).
        For each aircraft pair \((i, j)\), compute the instant \(\tau_{i j}^{h} \in[0, T]\) as
                    \(\tau_{i j}^{h}=-\frac{\sum_{k \in K}\left(x_{i k}^{0}-x_{j k}^{0}\right)\left(q_{i}^{*} v_{i} u_{i k}-q_{j}^{*} v_{j} u_{j k}\right)}{\sum_{k \in K}\left(q_{i}^{*} v_{i} u_{i k}-q_{j}^{*} v_{j} u_{j k}\right)^{2}}\).
        if \(\sum_{k \in K}\left(\left(x_{i k}^{0}-x_{j k}^{0}\right)+\tau_{i j}^{h}\left(q_{i}^{*} v_{i} u_{i k}-q_{j}^{*} v_{j} u_{j k}\right)\right)^{2} \geq d^{2} \quad \forall i<j \in A\) then
            The algorithm terminates and \(q^{*}\) 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:
                \(\sum_{k \in K}\left(\left(x_{i k}^{0}-x_{j k}^{0}\right)+\tau_{i j}^{h}\left(q_{i} v_{i} u_{i k}-q_{j} v_{j} u_{j k}\right)\right)^{2} \geq d^{2}\).
            \(h:=h+1\)
        end if
    end while
```

The problem $R_{h}$, solved at each iteration of Algorithm 1 , is nonconvex since constraints 23) are of the form $f\left(q_{i}, q_{j}\right) \geq d^{2}$ with $f\left(q_{i}, q_{j}\right)$ 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. $\tau_{i j}^{h}$, 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 $h^{\prime} \in\{1, \ldots, h-1\}$,

$$
m:=\max \left\{\tau_{i j}^{h}, \tau_{i j}^{h^{\prime}}\right\}, \quad p:=\min \left\{\tau_{i j}^{h}, \tau_{i j}^{h^{\prime}}\right\}
$$

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

$$
\begin{gathered}
\sum_{k \in K}\left[\left(x_{i k}^{0}-x_{j k}^{0}\right)^{2}+m^{2} \psi_{i j k}^{2}+2 m\left(x_{i k}^{0}-x_{j k}^{0}\right) \psi_{i j k}\right] \\
\sum_{k \in K}\left[\left(x_{i k}^{0}-x_{j k}^{0}\right)^{2}+p^{2} \psi_{i j k}^{2}+2 p\left(x_{i k}^{0}-x_{j k}^{0}\right) \psi_{i j k}\right]
\end{gathered}
$$

First, we remove the nonnegative term $\sum_{k \in K}\left(x_{i k}^{0}-x_{j k}^{0}\right)^{2}$ from both cuts and just compare

$$
\sum_{k \in K}\left[m^{2} \psi_{i j k}^{2}+2 m\left(x_{i k}^{0}-x_{j k}^{0}\right) \psi_{i j k}\right], \quad \text { and } \quad \sum_{k \in K}\left[p^{2} \psi_{i j k}^{2}+2 p\left(x_{i k}^{0}-x_{j k}^{0}\right) \psi_{i j k}\right] .
$$

Moreover, notice that

$$
\sum_{k \in K} m^{2} \psi_{i j k}^{2} \geq \sum_{k \in K} p^{2} \psi_{i j k}^{2}
$$

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

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

$$
\sum_{k \in K}\left[m^{2} \psi_{i j k}^{2}+2 m\left(x_{i k}^{0}-x_{j k}^{0}\right) \psi_{i j k}\right] \geq \sum_{k \in K}\left[p^{2} \psi_{i j k}^{2}+2 p\left(x_{i k}^{0}-x_{j k}^{0}\right) \psi_{i j k}\right]
$$

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

$$
\sum_{k \in K}\left[m^{2} \psi_{i j k}^{2}+2 m\left(x_{i k}^{0}-x_{j k}^{0}\right) \psi_{i j k}\right] \leq \sum_{k \in K}\left[p^{2} \psi_{i j k}^{2}+2 p\left(x_{i k}^{0}-x_{j k}^{0}\right) \psi_{i j k}\right]
$$

If, for each feasible $q_{i}$ and $q_{j}$,

$$
\begin{equation*}
\sum_{k \in K}\left(x_{i k}^{0}-x_{j k}^{0}\right) \psi_{i j k} \geq 0 \tag{24}
\end{equation*}
$$

then

$$
\sum_{k \in K} 2 m\left(x_{i k}^{0}-x_{j k}^{0}\right) \psi_{i j k} \geq \sum_{k \in K} 2 p\left(x_{i k}^{0}-x_{j k}^{0}\right) \psi_{i j k}
$$

also holds for each $q_{i}, q_{j} \in\left[q^{\min }, q^{\max }\right]$, since $m \geq p$ by definition.
However, if Eq. 24 is true for all $q_{i}, q_{j} \in\left[q^{\frac{\text { min }}{}}, q^{\text {max }}\right]$ (i.e., the cut involving $p$ dominates the one involving $m$ ), then $\tau_{i j}^{h} \leq 0$ for all $h$. In fact, $\tau_{i j}^{h}$ 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 $q_{j}$,

$$
\begin{equation*}
\sum_{k \in K}\left(x_{i k}^{0}-x_{j k}^{0}\right) \psi_{i j k}<0 \tag{25}
\end{equation*}
$$

we can analyse the relationship between cuts by comparing the positive term $\left(m^{2}-p^{2}\right) \sum_{k \in K} \psi_{i j k}^{2}$, and the negative term $2(m-p) \sum_{k \in K}\left(x_{i k}^{0}-x_{j k}^{0}\right) \psi_{i j k}$. In general we cannot infer anything about the above mentioned relationship, unless we consider specific subclasses of the problem. Therefore, although $\tau_{i j}^{h}$ 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)- 14 c , which models the HACADP. In this case, the nonconvex problem $R_{1}$ solved at the first iteration is

$$
\begin{aligned}
\min _{\theta} & \sum_{i \in A} \theta_{i}^{2} \\
\forall i \in A & \theta_{i}^{\min } \leq \theta_{i} \leq \theta_{i}^{\max } .
\end{aligned}
$$

```
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 \(\theta^{*}\).
        For each aircraft pair \((i, j)\), compute the instant \(\tau_{i j}^{h} \in[0, T]\) as
\[
\begin{equation*}
\tau_{i j}^{h}=-\frac{\left(x_{i}^{0}-x_{j}^{0}\right) c_{i j}^{*}+\left(y_{i}^{0}-y_{j}^{0}\right) s_{i j}^{*}}{\left(c_{i j}^{*}\right)^{2}+\left(s_{i j}^{*}\right)^{2}} \tag{26}
\end{equation*}
\]
        with \(c_{i j}^{*}=\cos \left(\phi_{i}+\theta_{i}^{*}\right)-\cos \left(\phi_{j}+\theta_{j}^{*}\right)\) and \(s_{i j}^{*}=\sin \left(\phi_{i}+\theta_{i}^{*}\right)-\sin \left(\phi_{j}+\theta_{j}^{*}\right)\).
        if \(\left[\left(x_{i}^{0}-x_{j}^{0}\right)+\tau_{i j}^{h} c_{i j}^{*}\right]^{2}+\left[\left(y_{i}^{0}-y_{j}^{0}\right)+\tau_{i j}^{h} s_{i j}^{*}\right]^{2} \geq d^{2} \quad \forall i<j \in A\) then
            The algorithm terminates and \(\theta^{*}\) 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:
\[
\begin{align*}
& {\left[\left(x_{i}^{0}-x_{j}^{0}\right)+\tau_{i j}^{h}\left(\cos \left(\phi_{i}+\theta_{i}\right) v_{i}-\cos \left(\phi_{j}+\theta_{j}\right) v_{j}\right)\right]^{2} } \\
+ & {\left[\left(y_{i}^{0}-y_{j}^{0}\right)+\tau_{i j}^{h}\left(\sin \left(\phi_{i}+\theta_{i}\right) v_{i}-\sin \left(\phi_{j}+\theta_{j}\right) v_{j}\right)\right]^{2} \geq d^{2} } \tag{27}
\end{align*}
\]
\[
h:=h+1
\]
        end if
    end while
```

Again, the problem $R_{h}$, solved at each iteration of the algorithm, is nonconvex since the constraints (27) are of the form $f\left(\theta_{i}, \theta_{j}\right) \geq d^{2}$ with $f\left(\theta_{i}, \theta_{j}\right)$ convex. We find $\theta^{*}$ 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, $\tau_{i j}^{h}$ 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 Tab1), 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 $\gamma_{i}$ (angle that the vector of the direction $u_{i}$ forms with the axis $k_{3}$ ) and the heading angle $\phi_{i}$ (angle between the projection of $u_{i}$ onto the $k_{1} k_{2}$-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 $\phi_{i}$ randomly chosen in $\left[-\frac{\pi}{6}, \frac{\pi}{6}\right]$ 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 $d=5 \mathrm{NM}$. All the solvers are run with their default
settings. The tests are performed on a $2.53 \mathrm{GHz} \operatorname{Intel}(\mathrm{R}) \mathrm{Xeon}(\mathrm{R}) \mathrm{CPU}$ with 49.4 GB RAM.

### 6.1 ADP in 3 dimensions via speed regulation

For the sphere instances the initial speed is $v_{i}=400 \mathrm{NM} / \mathrm{h}$ for each $i \in A$ and the angles $\gamma_{i}$ and $\phi_{i}$ are randomly generated within $\left[0, \frac{\pi}{2}\right]$ and parameters $x_{i k}^{0}$ and $u_{i k}$ are given by

$$
u_{i 1}=\cos \left(\phi_{i}\right) \sin \left(\gamma_{i}\right), \quad u_{i 2}=\sin \left(\phi_{i}\right) \sin \left(\gamma_{i}\right), \quad u_{i 3}=\cos \left(\gamma_{i}\right), \quad x_{i k}^{0}=-r u_{i k}
$$

where the sphere radius $r$ is chosen in $\{100,200, \ldots, 700\}$. The bounds $q_{i}^{\text {min }}$ 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 11. 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 $v_{i}=400 \mathrm{NM} / \mathrm{h}$ for each $i \in A$ for all the instances. For the circle instances the angles $\phi_{i}$ are randomly generated and parameters $x_{i}^{0}$ and $y_{i}^{0}$ are given by

$$
x_{i}^{0}=-r \cos \left(\phi_{i}\right), \quad y_{i}^{0}=-r \sin \left(\phi_{i}\right) .
$$

For the random circle instances both the angles $\phi_{i}$ and the parameters $x_{i}^{0}$ and $y_{i}^{0}$ are randomly generated. The bounds $\theta_{i}^{\text {min }}$ and $\theta_{i}^{\text {max }}$ 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 22. 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 20 e . 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 23 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 wellknown 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 | $s l v$ | obj | cpu | slv | obj | cpu | slv | obj | cpu | $s l v$ |
| 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 |
| $p b \_n 4 \_1$ | 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 |
| $p b \_n 4 \_3$ | 4 | 6 | 0.001318 | 25.3 | C | 0.001318 | 85.2 | C | 0.001318 | 33.5 | C | 0.001317 | 36.9 | C |
| $p b \_n 5$-1 | 5 | 10 | 0.011629 | 25.3 | MS | 0.011629 | 38.4 | MS | 0.011629 | 27.4 | MS | 0.011591 | 938 | C |
| $p b \_n 5$-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 |
| $p b \_n 5$-4 | 5 | 10 | 0.014750 | 22.9 | MS | 0.014750 | 40.5 | MS | 0.014750 | 27.6 | MS | 0.014717 | 2.84 | MS |
| $p b \_n 5$ _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 |
| $p b \_n 5$-7 | 5 | 10 | 0.012273 | 31.6 | MS | 0.012273 | 42.3 | MS | 0.012274 | 37.2 | MS | 0.012248 | 1.54 | MS |
| $p b \_n 5$-8 | 5 | 10 | 0.017556 | 36.0 | MS | 0.017556 | 42.0 | MS | 0.017556 | 39.5 | MS | 0.017548 | 2.33 | MS |
| $p b \_n 5$-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 |
| $p b \_n 5 \_12$ | 5 | 10 | 0.012122 | 23.7 | MS | 0.012121 | 39.0 | MS | 0.012122 | 25.3 | MS | 0.012113 | 1.48 | MS |
| $p b \_n 5 \_13$ | 5 | 10 | 0.023265 | 32.1 | MS | 0.023265 | 43.9 | MS | 0.023265 | 33.4 | MS | 0.023269 | 1.55 | MS |
| $p b \_n 5 \_14$ | 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 |
| $p b \_n 5 \_16$ | 5 | 10 | 0.010378 | 31.9 | MS | 0.010378 | 93.8 | MS | 0.010378 | 35.8 | MS | 0.010344 | 2370 | C |
| $p b \_n 5 \_17$ | 5 | 10 | 0.011956 | 34.9 | MS | 0.011956 | 38.8 | MS | 0.011956 | 39.2 | MS | 0.011985 | 1.18 | MS |
| $p b \_n 5 \_18$ | 5 | 10 | 0.011167 | 24.6 | MS | 0.011167 | 32.1 | MS | 0.011167 | 27.5 | MS | 0.011130 | 473 | C |
| $p b \_n 5 \_19$ | 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 |
| $p b \_n 6 \_2$ | 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 |
| $p 67$-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 |  |  |  |  |  |  |  |
| $p b \_n 2$ | 2 | 1 | 0.001250 | 0.001250 | 0.001250 | 0.001250 | 0.001250 |
| $p b \_n 3 \_1$ | 3 | 3 | 0.002501 | 0.002501 | 0.002501 | 0.002500 | 0.002501 |
| $p b \_n 3 \_2$ | 3 | 3 | 0.006672 | 0.006672 | 0.006657 | 0.006650 | 0.006665 |
| $p b \_n 3 \_3$ | 3 | 3 | 0.000950 | 0.000950 | 0.000950 | 0.000950 | 0.000950 |
| $p b \_n 4 \_1$ | 4 | 6 | 0.007240 | 0.007240 | 0.007228 | 0.007217 | 0.007240 |
| $p b \_n 4 \_2$ | 4 | 6 | 0.017061 | 0.017094 | 0.017095 | 0.017098 | 0.017065 |
| $p b \_n 4 \_3$ | 4 | 6 | 0.001318 | 0.001318 | 0.001318 | 0.001317 | 0.001318 |
| $p b \_n 5 \_1$ | 5 | 10 | 0.011629 | 0.011629 | 0.011629 | 0.011591 | 0.011629 |
| $p b \_n 5 \_2$ | 5 | 10 | 0.018467 | 0.018467 | 0.018467 | 0.018482 | 0.018468 |
| $p b \_n 5 \_3$ | 5 | 10 | 0.017122 | 0.017121 | 0.017122 | 0.017103 | 0.017100 |
| $p b \_n 5 \_4$ | 5 | 10 | 0.014750 | 0.014750 | 0.014750 | 0.014717 | 0.014750 |
| $p b \_n 5 \_5$ | 5 | 10 | 0.012163 | 0.012163 | 0.012163 | 0.012119 | 0.012149 |
| $p b \_n 5 \_6$ | 5 | 10 | 0.011235 | 0.011235 | 0.011235 | 0.011225 | 0.011225 |
| $p b \_n 5 \_7$ | 5 | 10 | 0.012273 | 0.012273 | 0.012274 | 0.012248 | 0.012262 |
| $p b \_n 5 \_8$ | 5 | 10 | 0.017556 | 0.017556 | 0.017556 | 0.017548 | 0.017556 |
| $p b \_n 5 \_9$ | 5 | 10 | 0.019140 | 0.019140 | 0.019140 | 0.019088 | 0.019119 |
| $p b \_n 5 \_10$ | 5 | 10 | 0.026022 | 0.026022 | 0.026022 | 0.026122 | 0.025960 |
| $p b \_n 5 \_11$ | 5 | 10 | 0.011202 | 0.011202 | 0.011202 | 0.011179 | 0.011190 |
| $p b \_n 5 \_12$ | 5 | 10 | 0.012122 | 0.012121 | 0.012122 | 0.012113 | 0.012111 |
| $p b \_n 5 \_13$ | 5 | 10 | 0.023265 | 0.023265 | 0.023265 | 0.023269 | 0.023265 |
| $p b \_n 5 \_14$ | 5 | 10 | 0.013789 | 0.013789 | 0.013789 | 0.013779 | 0.013790 |
| $p b \_n 5 \_15$ | 5 | 10 | 0.014578 | 0.014578 | 0.014578 | 0.014543 | 0.014551 |
| $p b \_n 5 \_16$ | 5 | 10 | 0.010378 | 0.010378 | 0.010378 | 0.010344 | 0.010367 |
| $p b \_n 5 \_17$ | 5 | 10 | 0.011956 | 0.011956 | 0.011956 | 0.011985 | 0.011940 |
| $p b \_n 5 \_18$ | 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 |
| $p b \_n 5 \_20$ | 5 | 10 | 0.019774 | 0.019774 | 0.019774 | 0.019834 | 0.019739 |
| $p b \_n 5 \_21$ | 5 | 10 | 0.009050 | 0.009050 | 0.009051 | 0.009034 | 0.009051 |
| $p b \_n 5 \_22$ | 5 | 10 | 0.030611 | 0.030610 | 0.030611 | 0.030833 | 0.030577 |
| $p b \_n 5 \_23$ | 5 | 10 | 0.001543 | 0.001543 | 0.001543 | 0.001541 | 0.001543 |
| $p b \_n 6.1$ | 6 | 15 | 0.001667 | 0.001667 | 0.001667 | 0.001664 | 0.001649 |
| $p b \_n 6 \_2$ | 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 |
| $r p b 6 \_2$ | 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 |
| $r p b 7 \_1$ | 7 | 2 | 0.002199 | 0.000213 | 0.000213 | 0.000213 | 0.000210 |
| $r p b 7 \_2$ | 7 | 7 | 0.001161 | 0.001161 | 0.001161 | 0.001159 | 0.001162 |
| $r p b 7 \_3$ | 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 |
| $r p b 8$ _3 | 8 | 10 | 0.001006 | 0.001007 | 0.001006 | 0.001529 | 0.001019 |

## References

1. J. Villiers: Automatisation du controle 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. $8^{\text {th }}$ 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 flightlevel assignment and speed control for conflict resolution, Proceedings of the 48 h 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 $\mathbf{3 ( 1 )}, 3-11$ (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 MixedInteger 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)

[^0]:    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.

[^1]:    ${ }^{1} 1 \mathrm{NM}=1852 \mathrm{~m}$

