Two decomposition algorithms for solving a minimum weight maximum clique model for the air conflict resolution problem

Thibault Lehouillier, Jérémy Omer, François Soumis, Guy Desaulniers

Two Decomposition Algorithms for Solving a Minimum Weight Maximum Clique Model for the Air Conflict Resolution Problem*

Thibault Lehouillier[1,2], Jérémy Omer[3], François Soumis[1,2], and Guy Desaulniers[1,2]

[1]*Group for Research in Decision Analysis, HEC Montréal, 3000, chemin de la Côte-Sainte-Catherine, Montréal (Quebec) Canada, H3T 2A7*
[2]*École Polytechnique Montréal, 2900 Boulevard Édouard-Montpetit, Montréal (Quebec) Canada, H3T 1J4*
[3]*Institut de Recherche Mathématique de Rennes & INSA de Rennes, Rennes, France*

August 22, 2016

Abstract

In this article, we tackle the conflict resolution problem using a new variant of the minimum-weight maximum-clique model. The problem involves identifying maneuvers that maintain the required separation distance between all pairs of a set of aircraft while minimizing fuel costs. We design a graph in which the vertices correspond to a finite set of maneuvers and the edges connect conflict-free maneuvers. A maximum clique of minimal weight yields a conflict-free situation that involves all the aircraft and minimizes the costs induced. The innovation of the model is its cost structure: the costs of the vertices cannot be determined *a priori*, since they depend on the vertices in the clique. We formulate the problem as a mixed integer linear program. Since the modeling of the aircraft dynamics and the computation of trajectories is separated from the solution process, the model is flexible. As a consequence, our mathematical framework is valid for any hypotheses. In particular, the aircraft can perform dynamic velocity, heading, and flight-level changes. To solve instances involving a large number of aircraft spread over several flight levels, we introduce two decomposition algorithms. The first is a sequential mixed integer linear optimization procedure that iteratively refines the discretization of the maneuvers to yield a trade-off between computational time and cost. The second is a large neighborhood search heuristic that uses the first procedure as a subroutine. The best solutions for the available set of maneuvers are obtained in less than 10 seconds for instances with up to 250 aircraft randomly allocated to 20 flight levels.

Keywords: Air Traffic Control, Conflict Resolution , Mixed Integer Linear Optimization , Graph Theory , Decomposition Methods

*This is the preprint manuscript of an article published in European Journal of Operational Research. For the final version see `http://dx.doi.org/10.1016/j.ejor.2016.07.008`

1 Introduction

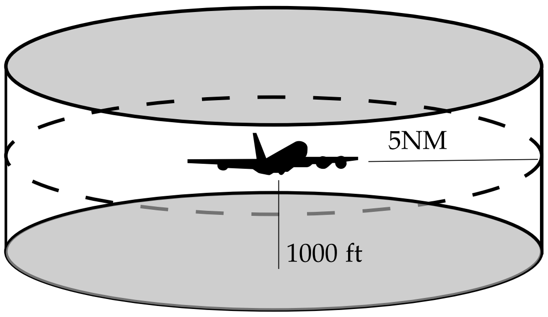
1.1 Context: Challenges of air traffic control

In recent years air traffic management (ATM) has attracted increasing attention, and research has focused on advanced decision algorithms. Such automated tools will be key components of future ATM systems such as the Single European Sky ATM Research (SESAR) SESAR Joint Undertaking (2012) project in Europe and the Next Gen Joint Planning and Development Office (2008) program in the United States. Optimization algorithms for air traffic control (ATC) are particularly relevant in the current context of growing traffic, where airspace capacity and safety become concerns. The latest long-term forecast from EUROCONTROL predicts that traffic demand will increase by 20% to 80% between 2012 and 2035 EUROCONTROL (2013). A simulation-based study performed by Lehouillier *et al.* Lehouillier et al. (2014) shows that for a 50% increase in traffic, the controllers in charge of busy sectors would have to resolve an average of 27 conflicts per hour. Decision tools are essential in such an environment.

1.2 Literature review

A fundamental challenge of ATC is the air conflict resolution (CR) problem. A conflict occurs when two aircraft fail to respect predefined horizontal and vertical separation distances of respectively 5 NM and 1000 ft, as illustrated in Figure 1. To resolve conflicts, the controllers impose speed, heading, or altitude-change maneuvers. Given the current position, speed, acceleration, and predicted trajectory of a set of aircraft, the CR problem consists in identifying the conflict-free maneuvers that minimize a given cost function.

Figure 1: Safety cylinder around an aircraft.



The CR problem has been widely studied. We provide a synthesis of the studies that had the greatest influence on our work; a more complete literature review may be found in Martín-Campo's thesis Martín-Campo (2010). Because aircraft trajectories are time-continuous, the most natural approach is to model the problem using optimal control Zhou et al. (1996). Analytical solutions can be found for only the simplest cases, but the models can be solved numerically using nonlinear programming techniques. For instance, Raghunathan *et al.* Raghunathan et al. (2004) use a time discretization of the problem to derive solutions for instances with more than two aircraft. One difficulty is that the nonlinear program (NLP) is nonconvex, so the global optimum cannot be found in a reasonable time and the solution is sensitive to the starting point.

Several heuristics have been developed to find feasible solutions quickly. Durand *et al.* Durand et al. (1996) and Meng and Qi Meng and Qi (2012) develop ant colony algorithms, where maneuvers are chosen from a finite discrete set of heading changes performed at constant speed. Alonso-Ayuso *et al.* Alonso-Ayuso et al. (2014) adapt a variable neighborhood search algorithm and consider only heading changes. Other methods use maneuvers extracted from a prescribed set Vivona et al. (2006), particle swarm optimization (see Gao *et al.* Gao et al. (2012) for heading changes), or neural networks (see Durand *et al.* Durand et al. (2000) and Christodoulou and Kontegeorgous Christodoulou and Kodaxakis (2006) for speed changes). These methods are fast, but convergence is not guaranteed.

Mixed integer linear and nonlinear programming provide powerful theoretical frameworks for CR. With the realistic restriction that the aircraft perform at most one maneuver at the initial time, Pallottino *et al.* Pallottino et al. (2002) exploit the geometry of the separation constraints to develop two mixed integer linear programs (MILPs) that allow either a speed change with a constant heading or a heading change with a constant speed. Vela *et al.* Vela et al. (2011) develop an MILP that allows both speed and heading changes, and Christodoulou and Costoulakis Christodoulou and Costoulakis (atia) describe a nonlinear model for three-dimensional CR. The MILP of Alonso-Ayuso *et al.* Alonso-Ayuso et al. (2011) allows both velocity and altitude changes. In Alonso-Ayuso et al. (2012), Alonso-Ayuso *et al.* extend the model of Pallottino et al. (2002) by replacing the instantaneous speed changes with continuous changes. Schouwenaars Schouwenaars (2006) and Omer and Farges Omer and Farges (2013) use a time-based discretization of the optimal control formulation. Vela *et al.* Vela et al. (2009) and Omer Omer (2015) develop MILPs with a space discretization that focus on the main points of interest of the CR.

In the ATM field, graph theory has primarily been used for air traffic flow management (ATFM) Bertsimas and Patterson (1998, 2000). In ATC, conflicts between aircraft are generally modeled by a graph in which the vertices represent the different aircraft and the edges link pairs of conflicting aircraft. Vela Vela (2011) and Sherali *et al.* Sherali et al. (2002) use conflict graphs in their models. Resmerita *et al.* Resmerita et al. (2003) study *a priori* CR by developing a multi-agent system where each aircraft must choose a path in a resource graph in which the vertices represent zones of the airspace and where the chosen paths must be conflict-free. Barnier and Brisset Barnier and Brisset (2004) assign different flight levels to aircraft with intersecting routes by looking for maximum cliques in a graph defining an assignment of all the aircraft to a set of given flight levels.

1.3 Contribution statement

We present a formulation of the CR problem as a variant of the minimum-weight maximum-cardinality clique (MWMCC) problem. A preliminary study is presented in Lehouillier et al. (2015b,a). We design a graph in which the vertices represent possible aircraft maneuvers and the edges link conflict-free maneuvers of different aircraft. The innovation of this model is its cost structure. The costs of the vertices are not known *a priori* since they depend on which maneuvers are in the clique. The model is flexible because it separates the solution process from the modeling of the aircraft dynamics and their maneuvers. As a consequence, our mathematical framework remains valid for any hypotheses on the aircraft dynamics and maneuvers, the computation of the separation distances, and the cost evaluation. This feature highlights its robustness, which is important in ATC because we must resolve many conflicts.

We have made several significant improvements to the model in Lehouillier et al. (2015b,a). First, we have corrected the cost computation. Second, our key contribution is that we have developed two decomposition algorithms to address the explosion of the number of vertices that occurs in large instances. The first algorithm is a sequential mixed integer linear optimization (SMILO) procedure that iteratively refines the discretization of the set of maneuvers without changing the number of vertices in the graph. This yields a trade-off between computational time and the cost of the optimal solution. This procedure is then used as a subroutine in a spatial decomposition that takes advantage of the geometric structure of the instances. The spatial decomposition is a large neighborhood search metaheuristic that exploits the weak interdependency between subsets of aircraft. Finally, we have tested our model on an extended benchmark that includes the structured instances with up to 20 aircraft described in Lehouillier et al. (2015b,a), and random instances with up to 60 aircraft on a single flight level and 250 aircraft over several flight levels. The results show that automated CR can be performed in a few seconds for large and dense areas of the airspace.

2 Problem Formulation

In this section, we discuss the modeling of the aircraft dynamics and maneuvers, the computation of the separation distances, and the cost evaluation. The choices made in this section represent a possible modeling of the problem. However, they are independent of the solution method, so considering other possibilities would not impact the validity of our overall method.

2.1 Modeling of aircraft dynamics

As is standard in the literature, we use a three-dimensional point-mass model for the aircraft dynamics:

$$\frac{dp_x}{dt} = V \cos\gamma \cos\chi \tag{1}$$

$$\frac{dp_y}{dt} = V \cos\gamma \sin\chi \tag{2}$$

$$\frac{dp_z}{dt} = V \sin\gamma \tag{3}$$

$$\frac{d\gamma}{dt} = \frac{g_0}{V}(n\cos\phi - \cos\gamma) \tag{4}$$

$$\frac{d\chi}{dt} = \frac{g_0}{V}\frac{n\sin\phi}{\cos\gamma} \tag{5}$$

$$\frac{dV}{dt} = \frac{F_T - F_D}{m} - g_0 \sin\gamma \tag{6}$$

The position of the aircraft is given by the coordinates (p_x, p_y, p_z) of its center of gravity in a local coordinate system, (p_x,p_y) being its coordinates in a horizontal plane and p_z its altitude. The aircraft flies at speed V and the angles χ, ϕ, and γ correspond respectively to its heading, roll, and pitch. The variables F_T and F_D are the norms of the thrust and drag forces respectively, m is the aircraft mass, n is the load factor, and g_0 is the gravitational acceleration.

We assume that aircraft follow their trajectories with a stepwise constant acceleration. Maneuvers are executed with a constant acceleration and yaw rate, and the speed vector remains constant between two consecutive maneuvers. This assumption is realistic because it respects the time-continuity of speed, and it corresponds to a setting where maneuvers are performed smoothly. Other speed changes could be considered, and they would impact only the computation of the separation distances and the maneuver costs. In the remainder of this article, $\mathcal{F} = [\![1; N]\!]$ denotes the set of aircraft.

2.2 Aircraft maneuvers

2.2.1 Types of maneuvers

We consider maneuvers of the following types:

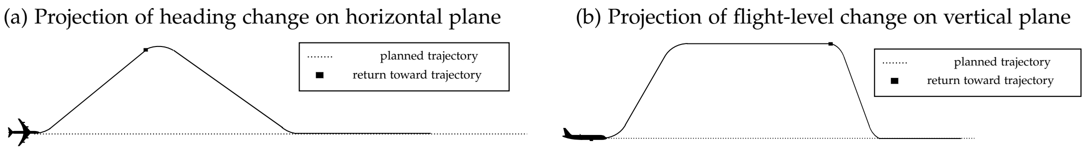
- Variable *NIL* refers to the *null* maneuver, i.e., no maneuver is performed.
- Variable H_θ is a heading change by an angle $\theta \in [-\pi; \pi]$[1].
- Variable S_δ is a relative speed change of δ%. We use relative speed changes because they are used in large-scale projects such as ERASMUS Brochard (2009).

[1]Positive angles correspond to counterclockwise rotations.

- Variable $V_{\delta h}$ is a change of δh flight levels.

Figure 2 describes the geometry of the heading change and the flight-level change. Heading changes are performed in a turning-point fashion as depicted in Figure 2a. Flight-level changes are followed by a return to the initial flight level, as in Figure 2b. We define $\mathcal{M} = \cup_{f=1}^{n} \mathcal{M}_f$ to be the set of all possible maneuvers, $\mathcal{M}_f$ being the set of maneuvers for aircraft $f \in \mathcal{F}$.

Figure 2: Geometry of heading change and flight-level change.


(a) Projection of heading change on horizontal plane
planned trajectory
return toward trajectory
(b) Projection of flight-level change on vertical plane
planned trajectory
return toward trajectory

2.2.2 Dynamics of the maneuvers

Since the analysis carried out by Omer and Farges Omer and Farges (2013) concludes that considering instantaneous maneuvers can lead to significant errors in the separation distances, we use a model with constant acceleration Paielli (2003). In Paielli (2003), Paielli states that the typical acceleration during a speed adjustment for commercial aircraft is of the order of 0.4 kn/s or 0.02 g. This value is chosen to respect the comfort of the passengers. Heading changes are approximated by a steady turn of constant rate and radius, given by the following equations:

$$\omega = \frac{d\chi}{dt} = \frac{g_0 \tan\phi}{V} \tag{7}$$

$$r = \frac{V^2}{g_0 \tan\phi} \tag{8}$$

We also consider the altitude maneuvers to be dynamic. The changes of flight level are performed at a vertical speed that is a function of thrust, drag, and true airspeed. Details on the computation of the vertical speed can be found in the BADA user manual BAD (2011).

2.2.3 Trajectory recovery

We consider that each aircraft follows a 4D contractual trajectory, which represents a compromise between the user's preferences and the capacity constraints of the network. The trajectories must satisfy time and space requirements over a sequence of 4D points; noncompliance leads to penalty fees. It is therefore important to ensure that after a maneuver is performed the aircraft recovers its initial trajectory as soon as possible. Ensuring strict velocity control can be costly and almost impossible in practice. Time recovery is therefore not required, but it is encouraged via a penalty on the time shift between the trajectory without conflict and the trajectory after a maneuver. We use the method of Omer Omer and Farges (2013) to compute the penalty costs: the penalty is estimated as the total cost induced by a time recovery of the 4D trajectory, at a speed depending on the sign of the shift.

2.2.4 Maneuver costs

We now discuss the computation of the maneuver cost. We wish to highlight that although the computations can be complex they do not interfere with the solution method described in Section 4. Moreover, more complex cost models could be considered without changes to the solution method.

For a jet commercial aircraft f with constant altitude, the fuel consumption by time and distance unit is

$$C_{t,f}\left(t, V_f(t)\right) = c_{1,f}\left(1 + \frac{V_f(t)}{c_{2,f}}\right) F_{T,f}(t) \tag{9}$$

$$C_{d,f}\left(t, V_f(t)\right) = \frac{C_{t,f}\left(t, V_f(t)\right)}{V_f(t)} \tag{10}$$

where the variables $c_{1,f}$ and $c_{2,f}$ are constants that depend on the aircraft type f.

The approach depends on the type of maneuver.

Speed change Consider a change of speed $V'_f = V^n_f(1+\delta)$ for a time t, where variable V^n_f is the nominal speed of aircraft f. Let C_{speed} be the cost of the maneuver. It is the sum of:

1. the cost of the additional fuel burnt during the maneuver, C^f_s;
2. the penalty for not respecting the 4D contract, C^{4D}_s.

Thus,

$$C_{\text{speed}} = C^f_s + C^{4D}_s \tag{11}$$

To compute C^f_s, it is useful to distinguish the cost incurred during the transition from V^n_f to V'_f, C^t_s, and that incurred on the portion of the trajectory flown with the new speed, C^n_s:

$$C^f_s = C^t_s + C^n_s \tag{12}$$

Let $t_{\delta,V}$ be the time required to go from V^n_f to V'_f. We compute C^t_s as follows:

$$C^t_s = \int_0^{t_{\delta,V}} C_{t,f}\left(t, V_f(t)\right) dt - t_{\delta,V} C_{t,f}\left(t, V^n_f\right) \tag{13}$$

The cost C^n_s is

$$C^n_s = (t - t_{\delta,V})\left(C_{t,f}(t, V'_f) - C_{t,f}(t, V^n_f)\right) \tag{14}$$

The penalty C^{4D}_s is deduced from the delay d^{4D}_s caused by the maneuver, which is computed as follows:

$$d^{4D}_s = \int_0^{t_{\delta,V}} V_f(t)dt + (t - t_{\delta,V})V'_f - tV^n_f \tag{15}$$

A recovery speed that depends on the sign of d^{4D}_s is used to return to the 4D trajectory, inducing the cost C^{4D}_s.

Heading change Let C_{heading} be the cost of a heading change by an angle θ for a period t. It is the sum of:

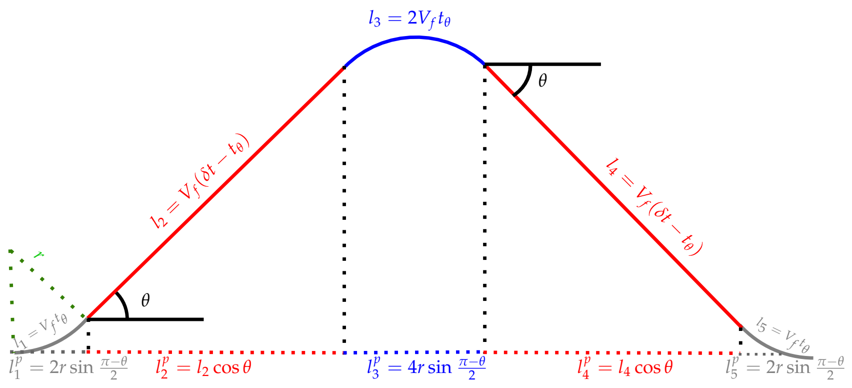
1. the cost of the additional distance induced by the maneuver, C^d_h;
2. the penalty for not respecting the 4D contract, C^{4D}_h.

Thus,

$$C_{\text{heading}} = C^d_h + C^{4D}_h \tag{16}$$

To recover the spatial trajectory, the aircraft performs a turn with an angle -2θ as detailed in Figure 3.

Figure 3: Geometry of the trajectory recovery following a heading change.


$l_3 = 2V_f t_\theta$
θ
$l_2 = V_f(\delta t - t_\theta)$
$l_4 = V_f(\delta t - t_\theta)$
r
θ
$l_1 = V_f t_\theta$
$l_5 = V_f t_\theta$
$l_1^p = 2r\sin\frac{\pi-\theta}{2}$
$l_2^p = l_2\cos\theta$
$l_3^p = 4r\sin\frac{\pi-\theta}{2}$
$l_4^p = l_4\cos\theta$
$l_5^p = 2r\sin\frac{\pi-\theta}{2}$

The aircraft flies an extra distance d given by

$$d = \sum_{i=1}^{5} l_i - l_i^p \tag{17}$$

where the expressions for l_i and l_i^p are given in Figure 3. The cost of the extra fuel burnt over d is then computed as follows:

$$C_d^h = C_{d,f}(t, V_f)d \tag{18}$$

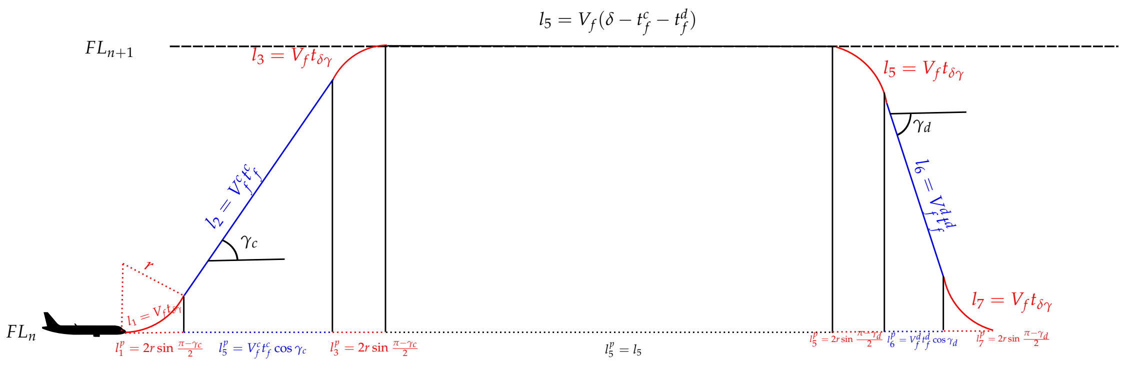
The penalty C_h^{4D} is then deduced from the delay d_h^{4D} due to the extra distance d:

$$d_h^{4D} = \frac{d}{V_f^n} \tag{19}$$

A recovery speed that depends on the sign of d_h^{4D} is used to return to the 4D trajectory, inducing the cost C_h^{4D}.

Flight-level change Consider a change of δh flight levels during a period t. Figure 4 illustrates the geometry of the maneuver for an example where an aircraft climbs one level.

Figure 4: Geometry of the trajectory recovery following a flight-level change.


$l_5 = V_f(\delta - t_f^c - t_f^d)$
FL_{n+1}
$l_3 = V_f t_{\delta\gamma}$
$l_5 = V_f t_{\delta\gamma}$
γ_d
$l_2 = V_f^c t_f^c$
$l_6 = V_f^d t_f^d$
γ_c
r
$l_1 = V_f t_{\delta\gamma}$
$l_7 = V_f t_{\delta\gamma}$
FL_n
$l_1^p = 2r\sin\frac{\pi-\gamma_c}{2}$
$l_5^p = V_f^c t_f^c \cos\gamma_c$
$l_3^p = 2r\sin\frac{\pi-\gamma_c}{2}$
$l_5^p = l_5$
$l_5^p = 2r\sin\frac{\pi-\gamma_d}{2}$
$l_6^p = V_f^d t_f^d \cos\gamma_d$
$l_7^p = 2r\sin\frac{\pi-\gamma_d}{2}$

Let C_{FL} be the cost of this maneuver. It is the sum of:

1. the extra cost during the ascent and descent, denoted C_{FL}^a and C_{FL}^d, respectively;
2. the difference between the cost on the initial flight level and that on the new flight level, denoted $C_{\text{FL}}^{\delta h}$;

3. the cost of the additional distance d, denoted C^d_{FL};

4. the 4D contract penalty C^{4D}_{FL}.

Thus,

$$C_{\text{FL}} = C^a_{\text{FL}} + C^d_{\text{FL}} + C^{\delta h}_{\text{FL}} + C^d_{\text{FL}} + C^p_{\text{FL}} \tag{20}$$

C^a_{FL} and C^d_{FL} are derived by

$$C^a_{\text{FL}} = t_a \left(C_{a,t}(f, V^a_f) - C_{t,f}(t, V^n_f) \right) \tag{21}$$

$$C^d_{\text{FL}} = t_d \left(C_{d,t}(f, V^d_f) - C_{t,f}(t, V^n_f) \right) \tag{22}$$

$$\tag{23}$$

where t_a, t_d are the durations of ascent and descent, $C_{a,t}$ and $C_{d,t}$ are the fuel costs during the ascent and descent, and V^a_f and V^d_f represent the ascent and descent speeds, respectively. The fuel consumption and the ascent and descent speeds are taken from the BADA tables for the corresponding aircraft types.

The difference in the fuel consumption between the two levels is as follows:

$$C^{\delta h}_{\text{FL}} = (\delta t - t_a - t_d)(C_{FL_{n+\delta h}} - C_{FL_n}) \tag{24}$$

where $C_{FL_{n+\delta h}}$ and C_{FL_n} denote the fuel consumption per time unit at the new and initial flight level, respectively. The fuel burnt on the additional distance d is as follows:

$$C^d_{\text{FL}} = C_{d,f}(t, V_f)d \tag{25}$$

where the distance d is

$$d = \sum_{i=1}^{7} l_i - l^p_i \tag{26}$$

and the expressions for l_i and l^p_i are given in Figure 4.

The computation of the 4D contract penalty, C^{4D}_{FL}, is similar to that for the heading maneuvers.

2.3 Aircraft separation

In the context of aircraft separation we use the following notation:

- $\mathcal{T}$: time horizon for the conflict resolution.
- $\boldsymbol{p}_i(t) \in \mathbb{R}^3$: position vector of aircraft i at time t. The variables $p_{i,x}(t)$, $p_{i,y}(t)$, and $p_{i,z}(t)$ denote respectively the abscissa, ordinate, and altitude components of the position vector.
- $\boldsymbol{s}_i(t) \in \mathbb{R}^3$: speed vector of aircraft i at time t. The variables $s_{i,x}(t)$, $s_{i,y}(t)$, and $s_{i,z}(t)$ denote respectively the abscissa, ordinate, and altitude components of the speed vector.
- $\boldsymbol{a}_i(t) \in \mathbb{R}^3$: acceleration vector of aircraft i at time t. The variables $a_{i,x}(t)$, $a_{i,y}(t)$, and $a_{i,z}(t)$ denote respectively the abscissa, ordinate, and altitude components of the acceleration vector.

Let i and j be two aircraft applying maneuvers m_i and m_j, respectively. Aircraft i and j are said to be separated at time t if and only if at least one of the following constraints holds:

$$d^h_{ij}(t)^2 = (p_{i,x}(t) - p_{j,x}(t))^2 + (p_{i,y}(t) - p_{j,y}(t))^2 \geq D^2_{\text{h,min}} \tag{27}$$

$$d^v_{ij}(t)^2 = (p_{i,z}(t) - p_{j,z}(t))^2 \geq D^2_{\text{v,min}} \tag{28}$$

At any time $t \in \mathcal{T}$, we may have both, one, or neither aircraft maneuvering. The set $\mathcal{T}$ can thus be divided into intervals where both i and j have a constant acceleration. For each interval, we find the time at which the aircraft are the closest to check whether the separation constraints hold. Let $\mathcal{T}_k$ be an interval, and let $t_0 \in \mathcal{T}$ be the starting time of maneuver m_i. If we assume that m_i is applied with a constant acceleration, we obtain the position and the speed vector of i at time $t_0 + t$ with t such that $t - t_0 \leq |\mathcal{T}_k|$:

$$\boldsymbol{p}_i(t_0+t) = \boldsymbol{p}_i(t_0) + (t-t_0)\boldsymbol{s}_i(t_0) + \frac{(t-t_0)^2}{2}\boldsymbol{a}_i(t_0) \tag{29}$$

$$\boldsymbol{s}_i(t_0+t) = \boldsymbol{s}_i(t_0) + (t-t_0)\boldsymbol{a}_i(t_0) \tag{30}$$

Let $\boldsymbol{p}^h_{ij}$ (respectively $\boldsymbol{s}^h_{ij}$, $\boldsymbol{a}^h_{ij}$) denote the horizontal position (respectively the speed and the acceleration) of aircraft j relative to aircraft i. We define

$$\begin{aligned} d^h_{ij}(t+\tau) &= ||\boldsymbol{p}^h_{ij}(t+\tau)|| \\ &= ||\boldsymbol{p}^h_{ij}(t) + \tau\boldsymbol{s}^h_{ij}(t) + \frac{\tau^2}{2}\boldsymbol{a}^h_{ij}(t)|| \end{aligned}$$

where $\tau \geq 0$.
Let $\tau_{ij} \in \operatorname*{argmin}_{\tau \geq 0} d^h_{ij}(t+\tau)^2$ and $t^h_{ij} \in \operatorname*{argmin}_{t \in \mathcal{T}} d^h_{ij}(t)^2$.
We have

$$t^h_{ij} = \begin{cases} 0 & \text{if } \tau_{ij} = 0 \\ |\mathcal{T}_k| & \text{if } \tau_{ij} \geq |\mathcal{T}_k| \\ \tau_{ij} & \text{otherwise} \end{cases}$$

where $|\mathcal{T}_k|$ is the length of interval $\mathcal{T}_k$. Aircraft i and j are horizontally separated during interval $\mathcal{T}$ if and only if

$$d^h_{ij}(t^h_{ij})^2 \geq D^2_{\text{h,min}} \tag{31}$$

Similarly, aircraft i and j are vertically separated during interval $\mathcal{T}$ if and only if

$$d^v_{ij}(t^v_{ij})^2 \geq D^2_{\text{v,min}} \tag{32}$$

Let $\mathcal{I}^h_{i,j}$ and $\mathcal{I}^v_{i,j}$ be the intervals during which i and j are not separated horizontally and vertically, respectively. Then i and j are separated if and only if

$$\mathcal{I}^h_{i,j} \cap \mathcal{I}^v_{i,j} = \emptyset \tag{33}$$

3 Modeling of the CR problem as an MWMCC problem

We now describe how the CR problem can be modeled as an MWMCC problem. This model is based on a preliminary study presented in Lehouillier et al. (2015b,a).

3.1 Graph theory definitions

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a simple undirected graph with vertex set $\mathcal{V}$ and edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$.

A *clique* in $\mathcal{G}$ is a vertex set $\mathcal{C}$ with the property that each pair of vertices in $\mathcal{C}$ is linked by an edge:

$$\mathcal{C} \subseteq \mathcal{V} \text{ is a clique} \Leftrightarrow \forall (u,v) \in \mathcal{C} \times \mathcal{C}, u \neq v, (u,v) \in \mathcal{E} \tag{34}$$

A *maximum* clique in $\mathcal{G}$ is a clique that is not a subset of any other clique in $\mathcal{G}$. The cardinality of a maximum clique of $\mathcal{G}$ is called the *clique number* and denoted $w(\mathcal{G})$. Let $c : \mathcal{V} \to \mathbb{R}$ be a vertex-weight function associated with $\mathcal{G}$. A *maximum* clique of *minimum* weight in $\mathcal{G}$ is a maximum clique $\mathcal{C}$ that minimizes $\sum_{v \in \mathcal{C}} c(v)$.

A *stable set* $\mathcal{S} \subseteq \mathcal{V}$ is a subset of vertices such that no two are adjacent in $\mathcal{G}$. A *bipartite* graph is a graph in which the vertices can be partitioned into two distinct stable sets, $\mathcal{V}_1$ and $\mathcal{V}_2$. Each edge of the graph then connects one vertex of $\mathcal{V}_1$ to a vertex of $\mathcal{V}_2$. This concept can be extended to $k-$*partite* graphs, where the vertex set is partitioned into k distinct stable sets.

The *density* of a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is defined as the ratio of the number of edges $|\mathcal{E}|$ to the number of edges in a complete graph with $|\mathcal{V}|$ vertices:

$$d_{\mathcal{G}} = \frac{|\mathcal{E}|}{\frac{|\mathcal{V}|(|\mathcal{V}|-1)}{2}} \tag{35}$$

3.2 Graph construction

We use a *conflict graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to model the CR problem.

3.2.1 Defining the vertices

The set of vertices is $\mathcal{V} = [\![1; |\mathcal{M}|]\!]$, and $\mathcal{V}_f$ is the set of vertices corresponding to aircraft f. In emergency scenarios where feasibility is a concern, one can introduce n vertices corresponding to expensive emergency maneuvers. Such maneuvers can for example correspond to those implemented by the Terminal Collision Avoidance system FAA (2011) or those described by Schouwenaars Schouwenaars (2006). However, since feasibility has not been an issue in our tests, we did not add these vertices.

3.2.2 Defining the edges

Let $(i, j) \in \mathcal{V} \times \mathcal{V}$ be a pair of vertices representing maneuvers $(m_i, m_j) \in \mathcal{M} \times \mathcal{M}$ of aircraft $(i, j) \in \mathcal{F} \times \mathcal{F}$. For $i \neq j$, we write $m_i \square m_j$ if no conflict occurs when aircraft i performs maneuver m_i while j performs m_j. The set of edges $\mathcal{E}$ corresponds to the pairs of maneuvers performed by two aircraft without introducing conflicts:

$$\mathcal{E} = \left\{(i, j) \in \mathcal{V} \times \mathcal{V}, i \neq j : m_i \square m_j\right\} \tag{36}$$

3.2.3 Relative density

We can define a measure of density based on the structure of the conflict graph. There is no edge between two different maneuvers of a given aircraft, which yields Observation 3.1.

Observation 3.1 *For all $f \in \mathcal{F}, \mathcal{V}_f$ is a stable set, i.e., there is no edge linking two distinct vertices of $\mathcal{V}_f$. Hence, the graph $\mathcal{G}$ is $|\mathcal{F}|$-partite.*

We define the *relative density* of $\mathcal{G}$ in Equation (37), which is an adaptation of the density of a graph to the conflict graph using Observation 3.1. This quantity is more meaningful, since it compares the number of edges of a conflict graph to the maximum number of edges a conflict graph can have.

$$d^*_{\mathcal{G}} = \frac{|\mathcal{E}|}{\frac{|\mathcal{V}|(|\mathcal{V}|-1)}{2} - \sum_{f \in \mathcal{F}} \frac{|\mathcal{M}_f|(|\mathcal{M}_f|-1)}{2}} \tag{37}$$

3.3 Conflict-free solution: Formulation and example

As mentioned earlier, given the current position, speed, acceleration, and planned trajectories of a set of aircraft, solving the CR problem involves finding a conflict-free set of maneuvers that minimizes the total cost. Observation 3.2 links the cliques in $\mathcal{G}$ to the CR problem:

Observation 3.2 *Let $\mathcal{C}$ be a clique in graph $\mathcal{G}$. Then $\mathcal{C}$ represents a set of conflict-free maneuvers for a subset of $\mathcal{F}$ of cardinality $|\mathcal{C}|$.*

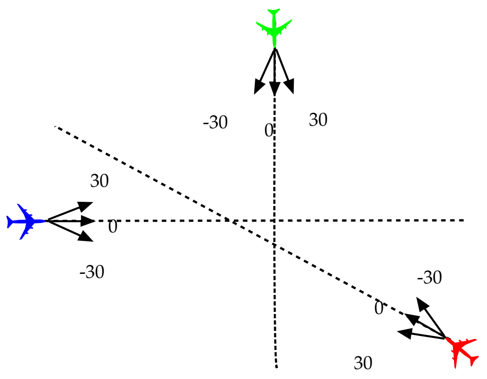
Observation 3.2 indicates that finding a set of conflict-free maneuvers for $\mathcal{F}$ is equivalent to finding a clique of $\mathcal{G}$ of cardinality $|\mathcal{F}|$. We can derive the following proposition:

Proposition 3.3 *If a conflict-free solution exists, then $\omega(\mathcal{G}) = |\mathcal{F}|$. Otherwise, $\omega(\mathcal{G})$ is the maximum number of flights involved in a conflict-free situation.*

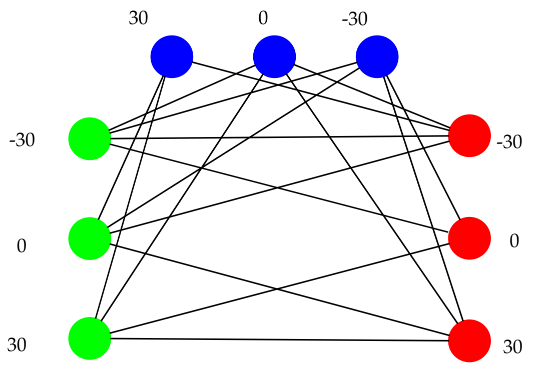
Figure 5 presents an example with three aircraft and the corresponding solution. If each aircraft follows its planned trajectory as indicated in Figure 5a, there will be conflicts. We assume here that, in addition to the null maneuver, only two heading changes ($\pm 30°$) are allowed. We build the CR graph shown in Figure 5b. The graph is multipartite, with each stable set representing the set of possible maneuvers for one aircraft. Solving the CR problem is then equivalent to searching for a minimum-weight clique of three vertices, i.e., a triangle. Figures 5c and 5d show the solution and the triangle of minimum weight respectively.

Figure 5: Example: Instance and solution.

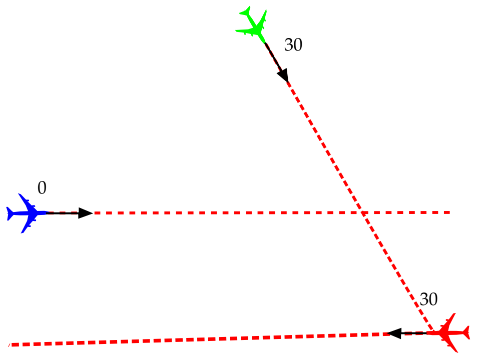
(a) Example with three aircraft



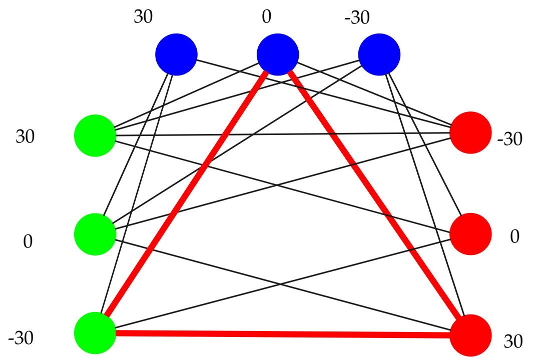
(b) Conflict graph $\mathcal{G}$ for example



(c) Solution



(d) Corresponding clique in conflict graph $\mathcal{G}$



We define the problem CR$_{\mathcal{M}}$ as the restriction of the CR problem to the set of maneuvers $\mathcal{M}$. Using Observations 3.2 and 3.3, we can restate the CR$_{\mathcal{M}}$ problem as follows: searching for a conflict-free solution of minimum cost is equivalent to solving the $\text{CR}_{\mathcal{M}}$ problem consisting of finding a clique of maximum cardinality and minimal cost in the graph $\mathcal{G}$.

As stated in Section 2.2, the cost of a maneuver depends on the time during which it is executed. One way to model this would be to discretize the execution time and to create the vertices accordingly. It would be straightforward to compute the cost of the vertices, using the method described in Section 2.2. The drawback of this method is the explosion of the number of vertices, which will drastically increase the computational time. To address this issue, we decided to keep the graph small by considering one vertex per maneuver. With this choice, the cost of the vertices cannot be determined *a priori*, since they depend on the maneuvers executed by the other aircraft. In other words, the cost of each vertex depends on the vertices in the clique. This problem is a new variant of the maximum-clique minimum-weight problem, where the weights are on the vertices, but they depend on the vertices in the clique.

3.4 Cost computation

For ease of presentation, and without loss of generality, in this section we make no distinction between a vertex and the corresponding maneuver. As explained above, the cost of a maneuver depends on its execution, which depends on the maneuvers performed by the other aircraft. As a consequence, we need to define the cost of the edges before the cost of the vertices.

3.4.1 Cost of the edges

An edge $e = (i, j)$ will be considered as a pair of maneuvers. We compute the cost of edge $e = (i, j)$ as a pair composed of the costs of maneuvers i and j, denoted $C_i^{(i,j)}$ and $C_j^{(i,j)}$. These costs correspond to an execution time t_i^j which is the minimum time for which maneuvers i and j must be performed to avoid a conflict when at least one of the aircraft returns to its original trajectory.

3.4.2 Cost of the vertices

To determine the cost of maneuver i, denoted c_i, we need to compute the time t_i for which it is performed. If i is not in the optimal solution, then $t_i = 0$. Otherwise,

$$t_i = \max_{j \in \mathcal{V} \cap \mathcal{C}} t_i^j \tag{38}$$

Equation (38) states that maneuver i has to be applied for a sufficient time to be conflict-free with every other chosen maneuver. Indeed, if aircraft i and j are conflict-free when they execute their maneuvers for a duration t, then they will remain conflict-free if they perform their maneuvers for $T > t$.
This leads to

$$c_i = \begin{cases} \max\limits_{j \in \mathcal{V} \cap \mathcal{C}} C_i^{(i,j)} & \text{if } i \in \mathcal{C} \\ 0 & \text{otherwise} \end{cases}$$

4 Methodology

4.1 MILP formulation

4.1.1 Motivation

Finding a maximum clique in an arbitrary graph is a well-known optimization problem that is $\mathcal{NP}$-hard; see Karp Karp (1972). The problem has been thoroughly studied and exact and heuristic methods have been developed. For an overview of the theoretical results and existing methods, see Bomze *et al.* Bomze et al. (1999) and Wu and Hao Wu and Hao (2015).

In the existing methods, the weights of the vertices are known beforehand, but in our model the costs of the vertices depend on which vertices are in the clique. Existing algorithms cannot be used, and we formulate our problem as an MILP that can be solved by any MILP solver.

4.1.2 Formulation

The decision variables of the model relate to the vertices of the graph. They correspond to the choice of the vertices in the clique and the cost of each vertex:

- $x_i = \begin{cases} 1 & \text{if vertex } i \text{ is in the maximum clique} \\ 0 & \text{otherwise} \end{cases}$
- $c_i \in \mathbb{R}_+$ is the cost of vertex i.

The clique search can then be modeled via the following MILP, denoted MIP:

$$\begin{aligned} \text{minimize } & \sum_{i \in \mathcal{V}} c_i && (39) \\ \text{subject to } & x_i + x_j \leq 1, \forall (i,j) \notin \mathcal{E} && (40) \\ & \sum_{i \in \mathcal{V}} x_i = N && (41) \\ & c_i \geq C_i^{(i,j)}(x_i + x_j - 1), \forall (i,j) \in \mathcal{E} && (42) \\ & x_i \in \{0;1\}, \forall i \in \mathcal{V} && (43) \\ & c_i \in \mathbb{R}_+, \forall i \in \mathcal{V} && (44) \end{aligned}$$

The objective function (39) minimizes the total maneuver cost. Constraints (40) are clique constraints stating that two nonadjacent vertices must not be part of the clique; this means that two conflicting maneuvers must not be part of the solution. Constraint (41) is based on Proposition 3.3, which defines the cardinality of the maximum clique. Constraints (42) compute the cost of the vertices: if a vertex is in the maximum clique, then its cost must be greater than the cost on every edge connecting it to the other vertices in the clique; this is the only constraint on the vertex cost. Constraints (43)–(44) are binarity and positivity constraints, respectively.

4.1.3 Strengthening the linear relaxation

Strengthening the linear relaxation of an MILP can reduce the computational time by providing better lower bounds. We added the following constraints:

$$\sum_{j \in \mathcal{V}_f} x_j = 1, \forall f \in \mathcal{F} \qquad (45)$$

Constraints (45) ensure that each aircraft is assigned a maneuver. These constraints were not included in the original formulation because constraints (40), (41), and (43) make them redundant. Each set of nodes $\mathcal{M}_f$ is a stable set, meaning that only one maneuver can be assigned to each aircraft in a clique. Since (41) ensures that the number of vertices in the clique is equal to the number of aircraft, (45) is always satisfied in a solution of (40)–(44). However, these constraints improve the linear relaxation because they prevent solutions where the fractional maneuvers are all assigned to the same aircraft.

4.2 Decomposition methods

The motivation for decomposition is two-fold. First, since the MWMCC problem is $\mathcal{NP}$-hard, the computational time will increase with the size of the instances. The size of the sets of maneuvers can also have an impact. Second, in practice the instances have important geometric characteristics: aircraft on different flight levels are weakly interdependent, meaning that they will almost never interfere with each other. However, these geometric considerations do not appear explicitly in our model.

We design two decomposition methods. In Section 5 we will analyze their effectiveness.

4.2.1 SMILO procedure

In this subsection, we present a SMILO procedure for the CR problem. This procedure iteratively solves several MILPs on graphs with the same number of vertices in which the discretization values are updated in a fashion similar to a trust region method. The goal is to obtain a trade-off between the computational time and the cost, and to study the impact of the chosen discretization. Algorithm 1 describes the SMILO procedure.

We use the following parameters:

- $\mathcal{F}$: set of aircraft;
- $v^f_{\min}, v^f_{\max}$: minimum and maximum speed deviation allowed for aircraft f;
- $\chi^f_{\min}$, $\chi^f_{\max}$: minimum and maximum heading deviation allowed for aircraft f;
- $\delta^f_v, \delta^f_\chi$: speed and heading discretization of the maneuvers of aircraft f;
- n^f_s, n^f_χ: number of speed and heading nodes for aircraft f, computed using the values of $v^f_{\min}, v^f_{\max}, \chi^f_{\min}, \chi^f_{\max}, \delta^f_v$, and δ^f_χ.

The procedure starts by storing the number of vertices representing speed and heading maneuvers for each aircraft. It sequentially solves the MIP until no improvement is achieved when updating the set of vertices $\mathcal{M}$ between two consecutive steps. The update of $\mathcal{M}$ depends on whether or not the current instance of the graph is feasible. If it is feasible, the update depends on the maneuver assigned to f in the current solution:

- If f perfoms no maneuver, $\mathcal{M}_f$ is erased, except for the NIL node. New sets of speed and heading maneuvers are added to $\mathcal{F}$ to obtain intervals centered around 0.
- If f performs a heading change of magnitude m, all the speed nodes are deleted from $\mathcal{M}_f$ and the heading interval is replaced with an interval with 0 and m as extrema. The discretization step is chosen to retain n^f_χ heading nodes.
- If f performs a speed change of magnitude m, all the heading nodes are deleted from $\mathcal{M}_f$ and the speed interval is replaced with an interval with 0 and m as extrema, depending on the sign of m. The discretization step is chosen to retain n^f_v speed nodes.

If the current instance is not feasible, all the parameters describing the maneuvers $\mathcal{F}$ are doubled, to allow for larger maneuvers while maintaining a constant number of vertices in the graph.

Algorithm 1 SMILO procedure for CR problem

```
 1: procedure SMILO($\mathcal{F}, v^1_{\min}, \ldots, v^N_{\min}, v^1_{\max}, \ldots, v^N_{\max}, \ldots, \chi^1_{\min}, \chi^N_{\min}, \chi^1_{\max}, \ldots, \chi^N_{\max}, \delta^1_v, \ldots, \delta^N_v, \delta^1_\chi, \ldots, \delta^N_\chi$)
 2:     for $f \in \mathcal{F}$ do
 3:         $n^f_v \leftarrow \frac{v^f_{\max} - v^f_{\min}}{\delta^f_v}$
 4:         $n^f_\chi \leftarrow \frac{\chi^f_{\max} - \chi^f_{\min}}{\delta^f_\chi}$
 5:     $z_c \leftarrow +\infty$
 6:     while $|z - z_c| > 0.01$ do
 7:         $z \leftarrow z_c$
 8:         $z \leftarrow SOLVE\_MIP(\mathcal{F}, v^1_{\min}, \ldots, v^N_{\min}, v^1_{\max}, \ldots, v^N_{\max}, \ldots, \chi^1_{\min}, \ldots, \chi^N_{\min}, \chi^1_{\max}, \ldots, \chi^N_{\max}, \delta^1_v, \ldots, \delta^N_v, \delta^1_\chi, \ldots, \delta^N_\chi)$
 9:         if $z < +\infty$ then
10:             for $f \in \mathcal{F}$ do
11:                 Let $m$ be the maneuver of aircraft $f$ in the last solution found
12:                 if $m$ is the null maneuver then
13:                     Erase all the heading and speed nodes of aircraft $f$
14:                     $\chi^f_{\max} \leftarrow \lfloor \frac{n^f_\chi}{2} \rfloor$
15:                     $\chi^f_{\min} \leftarrow -\chi^f_{\max}$
16:                     $\delta^f_\chi \leftarrow 1$
17:                     Build heading nodes with the values of $\chi^f_{\max}, \chi^f_{\min}$, and $\delta^f_\chi$
18:                     $v^f_{\max} \leftarrow \lfloor \frac{n^f_v}{2} \rfloor$
19:                     $v^f_{\min} \leftarrow -v^f_{\max}$
20:                     $\delta^f_v \leftarrow 1$
21:                     Build speed nodes with the values of $v^f_{\max}, v^f_{\min}$, and $\delta^f_v$
22:                 else
23:                     if $m$ is a heading maneuver then
24:                         Erase all speed nodes of aircraft $f$
25:                         $\chi^f_{\max} \leftarrow \max\{m, 0\}$
26:                         $\chi^f_{\min} \leftarrow \min\{m, 0\}$
27:                         $\delta^f_\chi \leftarrow \lfloor \frac{|m|}{n^f_\chi} \rfloor$
28:                         Build heading nodes with the values of $\chi^f_{\max}, \chi^f_{\min}$, and $\delta^f_\chi$
29:                     else
30:                         Erase all heading nodes of aircraft $f$
31:                         $v^f_{\max} \leftarrow \max\{m, 0\}$
32:                         $v^f_{\min} \leftarrow \min\{m, 0\}$
33:                         $\delta^f_v \leftarrow \lfloor \frac{|m|}{n^f_v} \rfloor$
34:                         Build speed nodes with the values of $v^f_{\max}, v^f_{\min}$, and $\delta^f_v$
35:         else
36:             for $f \in \mathcal{F}$ do
37:                 $v^f_{\max} \leftarrow 2v^f_{\max}$ , $\chi^f_{\max} \leftarrow 2\chi^f_{\max}$ , $v^f_{\min} \leftarrow 2v^f_{\min}$ , $\chi^f_{\min} \leftarrow 2\chi^f_{\min}$ , $\delta^f_v \leftarrow 2\delta^f_v$ , $\delta^f_\chi \leftarrow 2\delta^f_\chi$
38:         $z_c \leftarrow SOLVE\_MIP(\mathcal{F}, v^1_{\min}, \ldots, v^N_{\min}, v^1_{\max}, \ldots, v^N_{\max}, \ldots, \chi^1_{\min}, \ldots, \chi^N_{\min}, \chi^1_{\max}, \ldots, \chi^N_{\max}, \delta^1_v, \ldots, \delta^N_v, \delta^1_\chi, \ldots, \delta^N_\chi)$
```

4.2.2 Second decomposition method

The second decomposition method is inspired by the POPMUSIC algorithm of Taillard and Voss Taillard and Voss (2002). This metaheuristic is designed for combinatorial optimization problems that can be partially optimized. It generates neighborhoods that are a better fit for large problems. These neighborhoods need not be enumerated, since they can be implicitly explored with an optimization procedure. Algorithm 2 details the method.

Algorithm 2 Large Neighborhood Search (LNS) Algorithm

```
1: procedure LNS(r)
2:     Input: Solution $S$ composed of pieces $s_1, \dots, s_p$
3:     $O \leftarrow \emptyset$
4:     while $O \neq \{s_1, \dots, s_p\}$ do
5:         Select $s_i \notin O$
6:         Create a subproblem $R_i$ composed of the $r$ pieces $\{s_{i_1}, \dots, s_{i_r}\}$ most related to $s_i$
7:         Optimize $R_i$
8:         if $R_i$ has been improved then
9:             Update $S$
10:            $O \leftarrow \emptyset$
11:        else
12:            $O \leftarrow O \cup \{s_i\}$
```

To apply Algorithm 2, the user must define four key elements:

1. a definition of the pieces of a solution;
2. the selection procedure;
3. a definition of the relatedness of solution pieces;
4. the subproblem optimizer.

The algorithm works on a solution divided into p pieces. The algorithm first selects a piece p_0 to be optimized. It then creates a subproblem containing the r pieces of the solution that are the most closely related to p_0. If solving the subproblem yields no improvement in p_0, then this piece is added to the set of chosen pieces (labelled O). Otherwise, O is reset to the empty set. The algorithm repeats the process until the set O contains all the pieces of the solution.

In a nutshell, the algorithm iteratively tries to improve the current solution by performing neighborhood searches to improve every piece of the solution. The neighborhood of a piece of the solution is defined according to a user-specified relatedness criterion.

We apply Algorithm 2 to large instances with the aircraft randomly allocated to different flight levels. The four elements of our application are:

1. Each solution piece corresponds to one flight level.
2. We select pieces by starting from the lowest flight level.
3. The relatedness is the vertical distance between the flight levels.
4. The subproblem optimizer is Algorithm 1.

Algorithm 3 gives the details of the overall procedure. The set of aircraft is sorted by flight level. In the first loop (line 3), Algorithm 1 is used to optimize the problem for the corresponding aircraft, allowing only horizontal maneuvers. In the second part of the algorithm (line 10), for each flight level, Algorithm 1 is used to optimize the problem for the corresponding aircraft. Vertical maneuvers are now allowed, but the constraints consider the maneuvers of the set of aircraft on the adjacent levels. In other words, we authorize all types of maneuvers on the current flight level, but we fix the maneuvers of the aircraft on the adjacent levels to those that appear in the last solution found. Algorithm 3 stops when no improvement can be achieved.

Algorithm 3 Spatial decomposition method

```
1: procedure SPATIAL_ DECOMPOSITION($\mathcal{F}$)
2:     Input: $\mathcal{F}$: set of aircraft randomly generated on $p$ flight levels (FLs)
3:     for $i = 1, \dots, p$ do
4:         Resolve conflicts for FL $i$ without altitude maneuvers
5:         $s_i \leftarrow$ solution for FL $i$
6:     Call LNS(2)
7:     Input: Solution $S$ composed of pieces $s_1, \dots, s_p$
8:     $R_1$: resolve conflicts for FL 1 allowing altitude maneuvers, given the maneuvers of $s_2$
9:     $R_p$: resolve conflicts for FL $p$ allowing altitude maneuvers, given the maneuvers of $s_{p-1}$
10:    for $i = 2, \dots, p-1$ do
11:        $R_i$: resolve conflicts for FL $i$ allowing altitude maneuvers, given the maneuvers of $s_{i-1}$ and $s_{i+1}$
```

For this study, we apply Algorithm 3 to instances with the aircraft randomly allocated to several flight levels. However, Algorithm 3 could be applied to other types of instances. Adapting the relatedness between subsets of aircraft would divide the aircraft set into different clusters.

5 Results

In this section, we validate our model using a benchmark of complex two-dimensional and three-dimensional instances. The data used to compute the maneuvers and their costs were extracted from the BADA table for the Airbus A-320. The tests were performed on a computer equipped with a 3.4-GHz Intel Core i7-3770 processor and 8 GB of RAM. We implemented the algorithms in C++ and used CPLEX 12.5.1.0 CPL (2014) with its default options.

Our tables present the following information:

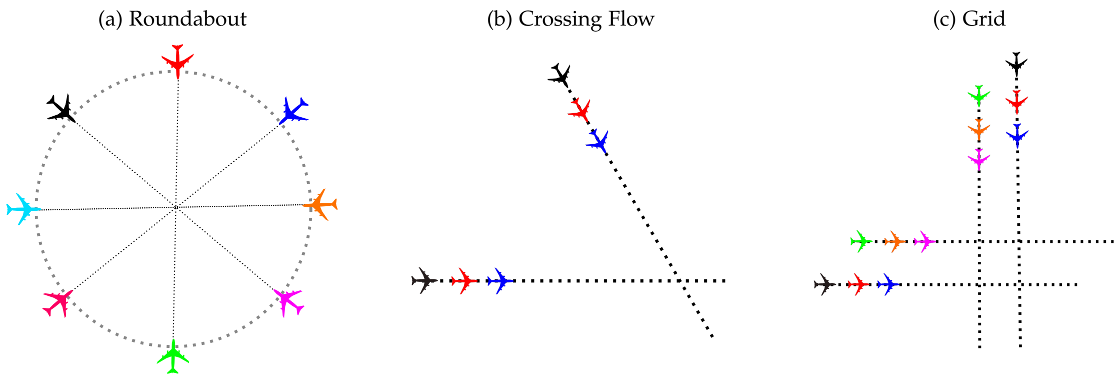
- *Case*: case configuration;
- $|\mathcal{F}|$: number of aircraft;
- $|\mathcal{V}|$: number of vertices;
- $|\mathcal{E}|$: number of edges;
- d^*: relative graph density;
- n: number of variables;
- m: number of constraints;
- z_{ip}: optimal value (in kilograms of fuel);
- n_{nodes}: number of branch-and-bound nodes;
- t_{lp}: time (s) to solve continuous relaxation of MILP;
- t_{ip}: time (s) to obtain z_{ip}.

5.1 Benchmark description

5.1.1 Structured instances

This benchmark has three types of instances. The roundabout instances $\mathcal{R}_n$ have n aircraft distributed on the circumference of a 100 NM radius; these aircraft fly toward the center at the same speed and altitude. The crossing flow instances $\mathcal{F}_{n,\theta,d}$ have two trails of n aircraft separated by d nautical miles that intersect each other at an angle θ. The grids $\mathcal{G}_{n,d}$ are composed of two flow instances $\mathcal{F}_{n,\frac{\pi}{2},d}$, with one instance translated 15 NM north-east of the other. Figure 6 gives an example of each instance.

Figure 6: Examples



5.1.2 Single-level random benchmark

This benchmark consists of random instances, where the aircraft are uniformly distributed in a square sector with a side length of 50 NM. To avoid infeasible instances, we perform preprocessing before solving the problem: for each pair of aircraft that will lose separation within the first 30 s of observation, we randomly delete one of the two aircraft. We generate 15% more aircraft than the number desired to anticipate the effect of the preprocessing. If too many aircraft remain after the preprocessing, we randomly remove extra aircraft until the desired number is reached.

5.1.3 Multi-level benchmark

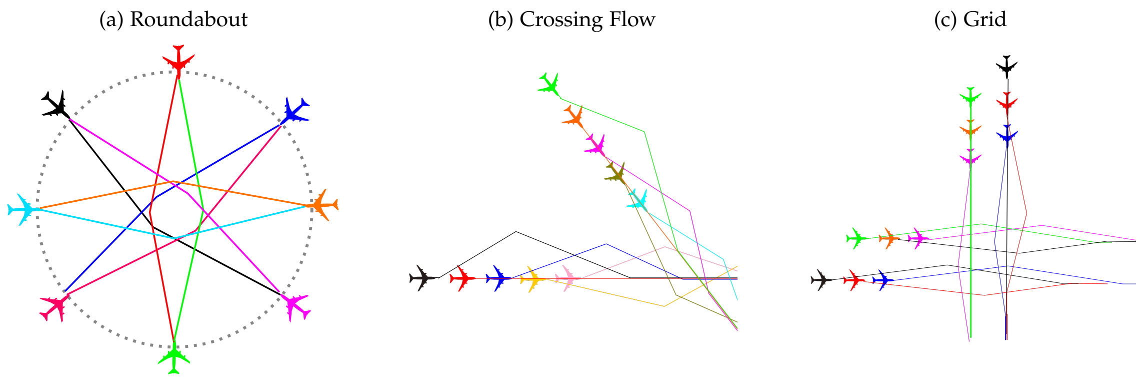
This benchmark contains instances that are more realistic. We generate more aircraft than for the single-level benchmark (from 50 to 200 aircraft in increments of 25). The aircraft generation is performed as in the single-level case. The aircraft are then randomly assigned to different flight levels, following a uniform distribution. We use $\mathcal{M}_{n,m}$ to label the instance where n aircraft are assigned to m flight levels.

5.2 Computational results

5.2.1 Structured instances

Figure 7 shows the solutions for the instances of Figure 6. Figure 7a shows the optimal solution for $\mathcal{R}_8$: the aircraft perform a right turn of 5° and avoid each other in a roundabout fashion before returning to their initial trajectories. Instance $\mathcal{F}_{3,\frac{\pi}{4},10}$ is solved symmetrically: each group of aircraft performs the same set of heading changes. The first aircraft of each group turns right by 5°, while the second and third aircraft turn left by 5° and 10° respectively. Instance $\mathcal{G}_{3,10}$ is also solved symmetrically: the horizontal and vertical groups perform the same set of maneuvers.

Figure 7: Solutions of the examples



The first set of simulations considers only horizontal maneuvers, with relative speed changes of ±2% , ±4%, and ±6% and heading changes of $\pm 5°, \pm 10°, and \pm 15°$. Table 1 gives the dimensions of the instances and the results for the original model. Section 5.3 gives the SMILO results for these instances. Algorithm 3 was not applied to this benchmark set because the inherent symmetry makes the design of a relatedness procedure not necessarily relevant.

The original model yields the optimal solution in real-time; complex problems with up to 20 aircraft can be solved in less than 15 s. This result is satisfying since the graph is dense.

Table 1: Dimensions of the instances and results for the virtual benchmark using only horizontal maneuvers

		Graph $\mathcal{G}$				MILP		Solution			
Instance type	Case	$\lvert\mathcal{F}\rvert$	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert$	d	m	n	z_{ip}	*nodes*	t_{lp}	t_{ip}
	$\mathcal{R}_4$	4	52	612	0.6	104	1333	2.66	27	0.02	0.07
	$\mathcal{R}_8$	8	104	2744	0.58	208	5705	5.34	75	0.02	0.51
Roundabout	$\mathcal{R}_{12}$	12	156	6300	0.56	312	12925	19.99	84	0.02	2.95
	$\mathcal{R}_{16}$	16	208	11396	0.56	416	23225	42.73	39	0.02	6.99
	$\mathcal{R}_{20}$	20	260	17756	0.55	520	36053	86.59	71	0.02	11.63
	$\mathcal{F}_{1,60,10}$	2	26	102	0.6	52	259	1.32	0	0.02	0.05
	$\mathcal{F}_{2,60,10}$	4	52	736	0.73	104	1581	2.66	0	0.02	0.06
	$\mathcal{F}_{3,60,10}$	6	78	1980	0.78	156	4123	4	0	0.02	0.18
	$\mathcal{F}_{4,60,10}$	8	104	3846	0.81	208	7909	5.34	57	0.02	0.57
Flow	$\mathcal{F}_{5,60,10}$	10	130	6349	0.83	260	12969	6.68	0	0.02	0.9
	$\mathcal{F}_{6,60,10}$	12	156	9483	0.85	312	19291	9.96	70	0.02	1.96
	$\mathcal{F}_{7,60,10}$	14	182	13252	0.86	364	26883	13.3	0	0.02	1.44
	$\mathcal{F}_{8,60,10}$	16	208	17659	0.87	416	35751	18.66	0	0.02	1.7
	$\mathcal{F}_{9,60,10}$	18	234	19057	0.88	468	42579	30.18	10	0.02	1.79
	$\mathcal{F}_{10,60,10}$	20	260	22563	0.87	520	47264	41.61	0	0.02	1.85
	$\mathcal{G}_{2,1,10}$	4	52	787	0.78	104	1683	1.32	35	0.02	0.18
	$\mathcal{G}_{2,2,10}$	8	104	3780	0.8	208	7777	3.33	0	0.02	0.28
Grid	$\mathcal{G}_{2,3,10}$	12	156	9072	0.81	312	18469	6.01	29	0.02	1.98
	$\mathcal{G}_{2,4,10}$	16	208	16854	0.83	416	34141	11.23	55	0.02	5.87
	$\mathcal{G}_{2,5,10}$	20	260	27207	0.85	520	54955	16.06	138	0.02	13.59

In the second simulation set, we introduce altitude maneuvers: the aircraft are allowed to move to an adjacent flight level. Table 2 gives the results. The solution times tend to slightly increase; this can be explained by the introduction of a new set of high-degree vertices. Every change of flight level is conflict-free with all the horizontal maneuvers. Nevertheless, the solution can still be computed quickly. These results are promising since our instances involve denser traffic than occurs in real-life situations.

Table 2: Dimensions of the instances and results for the virtual benchmark with flight-level changes

		Graph $\mathcal{G}$				MILP		Solution			
Instance type	Case	$\lvert\mathcal{F}\rvert$	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert$	d	m	n	z_{ip}	*nodes*	t_{lp}	t_{ip}
	$\mathcal{R}_4$	4	60	936	0.69	120	1997	2.66	0	0.02	0.08
	$\mathcal{R}_8$	8	120	4256	0.68	240	8761	5.34	67	0.02	0.7
Roundabout	$\mathcal{R}_{12}$	12	180	9864	0.66	360	20101	19.99	108	0.02	4.27
	$\mathcal{R}_{16}$	16	240	17876	0.66	480	36249	42.73	96	0.02	16.63
	$\mathcal{R}_{20}$	20	300	28016	0.66	600	56653	86.59	277	0.02	37.1
	$\mathcal{F}_{1,60,10}$	2	30	156	0.69	60	375	1.32	0	0.02	0.02
	$\mathcal{F}_{2,60,10}$	4	60	1068	0.79	120	2261	2.66	54	0.02	0.22
	$\mathcal{F}_{3,60,10}$	6	90	2814	0.83	180	5815	4	0	0.02	0.23
	$\mathcal{F}_{4,60,10}$	8	120	5406	0.86	240	11061	5.34	75	0.02	0.98
Flow	$\mathcal{F}_{5,60,10}$	10	150	8859	0.87	300	18029	6.68	0	0.02	1.19
	$\mathcal{F}_{6,60,10}$	12	180	13167	0.89	360	26707	9.96	74	0.02	2.4
	$\mathcal{F}_{7,60,10}$	14	210	18334	0.9	420	37103	13.3	123	0.02	3.6
	$\mathcal{F}_{8,60,10}$	16	240	24363	0.9	480	49223	18.66	103	0.02	7.2
	$\mathcal{F}_{9,60,10}$	18	270	31261	0.91	540	63081	30.18	79	0.02	9.13
	$\mathcal{F}_{10,60,10}$	20	300	39036	0.91	600	78693	41.61	92	0.02	11.51
	$\mathcal{G}_{2,1,10}$	4	60	1115	0.83	120	2355	1.32	53	0.02	0.1
	$\mathcal{G}_{2,2,10}$	8	120	5332	0.85	240	10913	3.33	0	0.02	0.42
Grid	$\mathcal{G}_{2,3,10}$	12	180	12744	0.86	360	25861	6.01	0	0.02	2.01
	$\mathcal{G}_{2,4,10}$	16	240	23542	0.87	480	47581	11.23	185	0.02	10.05
	$\mathcal{G}_{2,5,10}$	20	300	37807	0.87	600	76235	16.06	180	0.02	19.91

5.2.2 Single-level random benchmark

Figure 8a shows a randomly chosen instance $\mathcal{U}_{15}$ and Figure 8b shows the corresponding solution. The initial speed vectors are represented by dotted vectors, and the requested maneuvers are represented by solid vectors. The two aircraft circled in red changed their flight levels.

The results are reported in Table 3, where the figures are averages over 100 simulations. Table 4 gives the results for the single-level random benchmark with the addition of flight-level changes: the aircraft can climb to the next level or descend to the level below.

For some instances the computational time is now higher. To investigate this issue, we ran simulations for different numbers of possible maneuvers. The results showed that the computational time is sensitive to the number of vertices in a given instance. This observation motivated the SMILO procedure presented in Section 4.2.

5.3 Detailed results for the SMILO procedure on the benchmark without altitude changes

This simulation set was designed to explore two issues. The first is to identify the impact of the number of maneuvers on the objective function and the computational time. The second is to evaluate the performance of the SMILO procedure and to investigate the trade-off between the optimal value and the computational time.

We used four parameter sets:

Figure 8: Random instance $\mathcal{U}_{15}$ and its solution

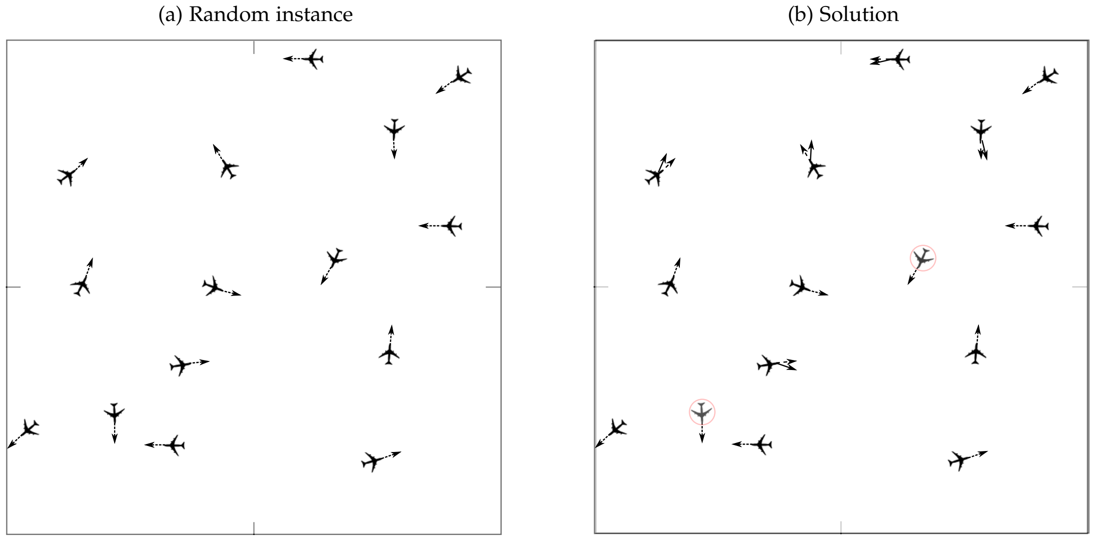


Table 3: Dimensions of the instances and results for the single-level random benchmark with horizontal maneuvers

		Graph $\mathcal{G}$				MILP		Solution			
Instance type	Case	$\|\mathcal{F}\|$	$\|\mathcal{V}\|$	$\|\mathcal{E}\|$	d	m	n	z_{ip}	*nodes*	t_{lp}	t_{ip}
Random	$\mathcal{U}_5$	5	55	726	1	110	1567	0.05	0	0.02	0
	$\mathcal{U}_{10}$	10	110	4830	0.99	220	9890	0.23	0	0.02	0
	$\mathcal{U}_{15}$	15	165	9865	0.99	330	20075	0.86	0	0.02	7.56
	$\mathcal{U}_{20}$	20	220	14365	0.99	440	29190	1.54	8	0.02	11.1
	$\mathcal{U}_{25}$	25	275	19182	0.98	550	38939	2.15	19	0.02	12.1
	$\mathcal{U}_{30}$	30	330	28750	0.99	660	58190	2.21	45	0.02	14.32
	$\mathcal{U}_{35}$	35	385	31406	0.99	770	63617	2.41	75	0.02	16.25
	$\mathcal{U}_{40}$	40	440	52766	0.99	880	106452	3.21	119	0.02	19.63
	$\mathcal{U}_{45}$	45	495	62051	0.99	990	125137	3.26	179	0.02	20.02
	$\mathcal{U}_{50}$	50	550	57215	1	1100	115580	3.87	225	0.02	20.06
	$\mathcal{U}_{55}$	55	605	69090	0.99	1210	139445	4.83	346	0.02	22.21
	$\mathcal{U}_{60}$	60	660	75338	0.98	1320	152056	6.32	561	0.02	29.35

- A large discretization, with relative speed changes of $\pm 6\%$ and heading changes of $\pm 15°$, yielding an objective function value z^l_{ip} in time t^l_{ip}.
- A medium discretization, with relative speed changes of $\pm 2\%$, $\pm 4\%$, and $\pm 6\%$ and heading changes of $\pm 5°, \pm 10°, and \pm 15°$, yielding an objective function value z^m_{ip} in time t^m_{ip}.
- A narrow discretization, with 12 relative speed changes between -6% and 6% with a step of 1% and heading changes between $-15°$ and $15°$ with a step of $1°$. These parameters yield an objective function value z^n_{ip} in time t^n_{ip};
- The SMILO procedure with four speed changes and four heading changes, yielding an objective function value z_{SMILO} in time t_{SMILO}.

Table 5 gives the results of these simulations. The results show that the choice of the discretization size has a critical impact on the computational time. Going from a medium

Table 4: Dimensions of the instances and results for the single-level random benchmark with horizontal maneuvers and flight-level changes

		Graph $\mathcal{G}$				MILP		Solution			
Instance type	Case	$\|\mathcal{F}\|$	$\|\mathcal{V}\|$	$\|\mathcal{E}\|$	d	m	n	z_{ip}	*nodes*	t_{lp}	t_{ip}
Random	$\mathcal{U}_5$	5	75	1209	1	150	2573	0.01	0	0.02	0.01
	$\mathcal{U}_{10}$	10	150	4839	1	300	9988	0.01	0	0.02	7.07
	$\mathcal{U}_{15}$	15	225	8939	0.98	450	18343	0.32	0	0.02	15.25
	$\mathcal{U}_{20}$	20	300	16732	0.99	600	34084	0.97	0	0.02	18.55
	$\mathcal{U}_{25}$	25	375	22345	0.98	750	45465	1.29	38	0.02	20.55
	$\mathcal{U}_{30}$	30	450	32248	0.99	900	65426	1.37	31	0.02	21.46
	$\mathcal{U}_{35}$	35	525	42027	0.99	1050	85139	1.76	0	0.02	25.22
	$\mathcal{U}_{40}$	40	600	50386	0.99	1200	102012	2.04	0	0.02	29.15
	$\mathcal{U}_{45}$	45	675	60764	0.99	1350	122923	3.18	41	0.02	34.02
	$\mathcal{U}_{50}$	50	750	69035	0.99	1500	139620	3.12	0	0.02	37.47
	$\mathcal{U}_{55}$	55	825	84344	0.99	1650	170393	4.55	89	0.02	40.16
	$\mathcal{U}_{60}$	60	900	75126	0.99	1800	152112	6.24	26	0.02	55.25

Table 5: Dimensions of instances and results

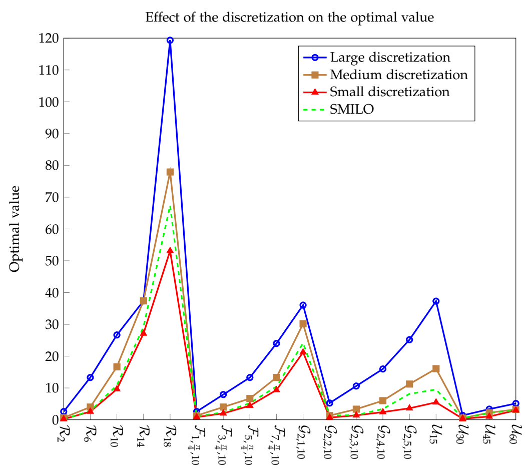
		Large discretization		Medium discretization		Small discretization		Iterative procedure		
Instance type	Case	z^l_{ip}	t^l_{ip}	z^m_{ip}	t^m_{ip}	z^n_{ip}	t^n_{ip}	z_{SMILO}	t_{SMILO}	CPLEX Calls
Roundabout	$\mathcal{R}_4$	7.93	0.04	2.66	0.07	0.96	1.45	1.06	0.06	4
	$\mathcal{R}_8$	18.64	0.25	5.34	0.51	5.34	69.2	6.42	0.26	4
	$\mathcal{R}_{12}$	32.03	0.72	19.99	2.95	15.71	362.06	18.03	0.75	3
	$\mathcal{R}_{16}$	42.73	2.37	42.73	6.99	38.67	2414.48	42.73	2.41	3
	$\mathcal{R}_{20}$	132.65	10.59	86.59	11.63	76.56	1162.29	86.59	10.39	4
Flow	$\mathcal{F}_{1,60,10}$	2.58	0.01	1.32	0.05	0.47	0.45	0.67	0.07	3
	$\mathcal{F}_{2,60,10}$	5.26	0.03	2.66	0.06	1.14	1.27	2.09	0.09	3
	$\mathcal{F}_{3,60,10}$	7.93	0.08	4	0.18	2	9.33	2.32	1.11	4
	$\mathcal{F}_{4,60,10}$	10.61	0.17	5.34	0.57	3.09	48.51	4.53	2.19	4
	$\mathcal{F}_{5,60,10}$	13.29	0.43	6.68	0.9	4.43	157.62	5.22	2.54	4
	$\mathcal{F}_{6,60,10}$	18.64	0.82	9.96	1.96	6.35	681.09	7.39	5.82	3
	$\mathcal{F}_{7,60,10}$	24	0.9	13.3	1.44	9.35	2431.05	10.45	5.95	4
	$\mathcal{F}_{8,60,10}$	29.35	1.17	18.66	1.7	14.67	3600.25	15.01	6.82	4
	$\mathcal{F}_{9,60,10}$	36.05	2.77	30.18	1.79	21.23	3675.12	24.11	7.43	5
	$\mathcal{F}_{10,60,10}$	47.22	3.33	41.61	1.85	37.17	3700.34	39.51	8.82	5
Grid	$\mathcal{G}_{2,1,10}$	5.26	0.04	1.32	0.18	0.69	1.95	1.32	0.55	3
	$\mathcal{G}_{2,2,10}$	10.61	0.18	3.33	0.28	1.39	14.93	1.44	0.66	4
	$\mathcal{G}_{2,3,10}$	15.97	0.64	6.01	1.98	2.41	62.33	3.41	6.96	6
	$\mathcal{G}_{2,4,10}$	25.17	2.77	11.23	5.87	3.61	317.8	8.14	13.21	3
	$\mathcal{G}_{2,5,10}$	37.3	4.5	16.06	13.59	5.48	1845.36	9.51	17.72	5
Random	$\mathcal{U}_{15}$	1.37	8.12	0.54	9.1	0.34	118.45	0.46	21.34	3
	$\mathcal{U}_{30}$	3.34	10.03	2.21	12.32	1.02	500.12	2.11	24.26	4
	$\mathcal{U}_{45}$	5.12	14.32	3.26	19.02	2.97	1002.87	3.07	32.6	6
	$\mathcal{U}_{60}$	8.45	17.01	6.32	23.35	5.98	1237.12	6.11	37.18	3

discretization to a narrow discretization divides the cost of the solution by 2 on average, but the solution time is 18 times higher on average. The SMILO results show that it provides a good trade-off between efficient solutions and computational time. The SMILO solutions are on average 41% more expensive than those for the medium discretization, but it takes 50% less time to find them. This is especially useful for the random instances, where the original optimization model seemed less efficient.

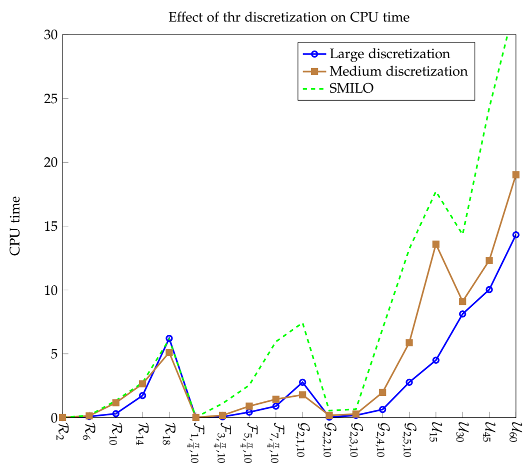
Figure 9 summarizes Table 5. Figure 9a shows the influence of the discretization on the quality of the best solution found and compares the SMILO solution to those obtained with the different discretizations. The SMILO solution is an intermediate between the solutions for the large and medium discretizations. Figure 9b compares the SMILO solution time with that for the large and medium discretizations; the solution times for the small discretization are omitted because the scale is different. The SMILO procedure needs less time than the medium discretization to find the optimal value for the available set of maneuvers.

Figure 9: Influence of the discretization on the optimal value and the CPU time.

(a) Influence on objective value



(b) Influence on CPU time



5.3.1 Evaluating the second decomposition method on the multi-level random benchmark

Table 6 gives the results for the second decomposition method on the multi-level random benchmark. We compare this method with the classical model. Here z_{ip} (respectively t_{ip}) corresponds to the optimal value (respectively CPU time) of the second decomposition method using the first method as a subroutine, and z^f_{best} (respectively t^f_{best}) is the value (respectively the CPU time) of the best solution found within one hour of computation. For the classical model only 6 of 24 instances are solved to optimality within one hour. This is because the model does not exploit the geometry: it considers the full instance instead of dividing it into flight levels, drastically increasing the complexity of the computations. In contrast, the decomposition method benefits from the geometry of the instances and the weak interdependency between the flight levels: every instance is solved to optimality within 9 s.

6 Conclusions

We began by designing an optimization model for air conflict resolution. We first designed a graph in which the vertices correspond to maneuvers and the edges link conflict-free maneuvers of distinct aircraft. A solution corresponds to a maximum clique of minimum cost in this graph. The costs of the vertices depend on the vertices belonging to the maximum clique. This makes our model a new variant of the search for a maximum clique of minimum weight. The main advantage of our model is its flexibility, arising because the solution process is separated from the modeling of the problem. The framework remains valid under a wide range of assumptions, and we will in the future be able to compare this model to other models.

Since the clique search problem is $\mathcal{NP}$-hard, we expected that the solution time should be sensitive to the number of maneuvers per aircraft. Moreover, in practice the set of aircraft has weak geometric dependencies that could be exploited. However, these dependencies do not appear explicitly in the model, and we designed two decomposition methods to take advantage of these features. The first is a sequential MILP that iteratively solves the problem while changing the discretization. This method provides a trade-off between efficient solutions and computational time. The second decomposition uses this method as a subroutine in a metaheuristic exploiting the geometry of the instances by solving pieces locally before finding a

Table 6: Dimensions of instances and results for the second decomposition method on the multi-level random benchmark

Case	Size		Solution					
	$\lvert\mathcal{F}\rvert$	$\lvert\mathcal{L}\rvert$	z_{ip}	*nodes*	t_{ip}	CPLEX Calls	z^{f}_{best}	t^{f}_{best}
$\mathcal{M}_{100,10}$	100	10	6.27	0	3.01	20	5.16	2692.44
$\mathcal{M}_{100,12}$	100	12	4.96	0	1.49	24	4.96	2310.54
$\mathcal{M}_{100,14}$	100	14	4.93	0	0.54	28	4.93	2066.45
$\mathcal{M}_{100,16}$	100	16	3.92	0	0.95	32	3.92	1975.94
$\mathcal{M}_{100,18}$	100	18	3.33	0	0.61	36	3.33	1964.45
$\mathcal{M}_{100,20}$	100	20	2.98	0	0.39	40	2.98	1712.24
$\mathcal{M}_{150,10}$	150	10	18.15	0	2.85	20	23.05	3600
$\mathcal{M}_{150,12}$	150	12	12.45	0	2.47	24	19.05	3600
$\mathcal{M}_{150,14}$	150	14	9.67	0	1.16	28	17.03	3600
$\mathcal{M}_{150,16}$	150	16	9.03	0	1.71	32	16.15	3600
$\mathcal{M}_{150,18}$	150	18	7.05	0	0.9	36	12.48	3600
$\mathcal{M}_{150,20}$	150	20	2.68	32	0.62	40	6.12	3600
$\mathcal{M}_{200,10}$	200	10	13.4	35	5.63	20	45.2	3600
$\mathcal{M}_{200,12}$	200	12	12.71	0	4.03	24	31.02	3600
$\mathcal{M}_{200,14}$	200	14	11.97	25	4.42	28	29.45	3600
$\mathcal{M}_{200,16}$	200	16	12.04	0	3.5	32	22.08	3600
$\mathcal{M}_{200,18}$	200	18	8.15	0	1.85	36	18.45	3600
$\mathcal{M}_{200,20}$	200	20	5.36	0	3.3	40	12.45	3600
$\mathcal{M}_{250,10}$	250	10	30.24	234	8.12	20	101.35	3600
$\mathcal{M}_{250,12}$	250	12	24.15	42	8.1	24	80.15	3600
$\mathcal{M}_{250,14}$	250	14	21.45	0	5.12	28	78.11	3600
$\mathcal{M}_{250,16}$	250	16	18.04	0	4	32	64.15	3600
$\mathcal{M}_{250,18}$	250	18	16.41	0	4.26	36	24.48	3600
$\mathcal{M}_{250,20}$	250	20	11.05	0	3.81	40	21.35	3600

global solution.

We tested our model on complex structured instances. The computational times were low (less than 15 s for instances with up to 20 aircraft). For larger instances, the times slightly increase but remain almost real-time. The simulations indicate that the model is sensitive to the number of maneuvers. The first decomposition method provides an efficient way to solve the problem according to the user's preferences, which may be time- or cost-oriented. The second procedure solved instances with up to 250 aircraft spread over up to 20 flight levels in less than 5 s, whereas the original model could not find the optimal solution within an hour.

Future research will introduce stochastic elements: we could consider errors in the trajectory predictions or introduce wind effects to make the study more realistic. We plan to use real-life instances, particularly instances involving altitude changes, to validate our model. The second decomposition method is particularly appropriate for such instances.

Acknowledgement

This work was carried out under the project OPR-601 funded by the CRIAQ (Consortium on Research and Innovation in Aerospace in Quebec).

References

(2011). Introduction to TCAS II - version 7.1. Technical report, Federal Aviation Administration.

(2011). User manual for the Base of Aircraft Data (BADA). Technical Report 11/03/08-08, Eurocontrol.

(2014). CPLEX v12.5. User's manual for CPLEX. Technical Report 11/03/08-08, IBM ILOG.

Alonso-Ayuso, A., Escudero, L. F., and Martín-Campo, F. J. (2011). Collision avoidance in air traffic management: A mixed-integer linear optimization approach. *IEEE Trans. Intell. Transport. Syst.*, 12(1):47–57.

Alonso-Ayuso, A., Escudero, L. F., and Martín-Campo, F. J. (2012). A mixed 0–1 nonlinear optimization model and algorithmic approach for the collision avoidance in ATM: Velocity changes through a time horizon. *Computers & Operations Research*, 39(12):3136–3146.

Alonso-Ayuso, A., Escudero, L. F., Martín-Campo, F. J., and Mladenović, N. (2014). A VNS metaheuristic for solving the aircraft conflict detection and resolution problem by performing turn changes. *Journal of Global Optimization*.

Barnier, N. and Brisset, P. (2004). Graph coloring for air traffic flow management. *Annals of Operations Research*, 130(1–4):163–178.

Bertsimas, D. and Patterson, S. S. (1998). The air traffic flow management problem with enroute capacities. *Operations Research*, 46(3):406–422.

Bertsimas, D. and Patterson, S. S. (2000). The traffic flow management rerouting problem in air traffic control: A dynamic network flow approach. *Transportation Science*, 34(3):239–255.

Bomze, I. M., Budinich, M., Pardalos, P. M., and Pelillo, M. (1999). The maximum clique problem. In Du, D.-Z. and Pardalos, P. M., editors, *Handbook of Combinatorial Optimization*, pages 1–74. Springer.

Brochard, M. (2009). ERASMUS final report v1.1. technical report d4.6. Technical report, EUROCONTROL, Eurocontrol Experimental Centre, Bretigny, France.

Christodoulou, M. and Costoulakis, C. (May 12-15, 2004, Dubrovnik, Croatia). Nonlinear mixed integer programming for aircraft collision avoidance in free flight. In *Proceedings of the 12th IEEE Mediterranean Electrotechnical Conference*, volume 1, pages 327–330. IEEE.

Christodoulou, M. A. and Kodaxakis, S. G. (2006). Automatic commercial aircraft-collision avoidance in free flight: The three-dimensional problem. *IEEE Transactions on Intelligent Transportation Systems*, 7(2):242–249.

Durand, N., Alliot, J.-M., and Médioni, F. (2000). Neural nets trained by genetic algorithms for collision avoidance. *Applied Intelligence*, 13(3):205–213.

Durand, N., Alliot, J.-M., and Noailles, J. (1996). Automatic aircraft conflict resolution using genetic algorithms. *Proceedings of the Symposium on Applied Computing, Philadelphia.*

EUROCONTROL (2013). Eurocontrol long-term forecast: IFR flight movements 2013–2035. Technical report, Eurocontrol - STATFOR.

Gao, Y., Zhang, X., and Guan, X. (2012). Cooperative multi-aircraft conflict resolution based on co-evolution. In *International Symposium on Instrumentation & Measurement, Sensor Network and Automation (IMSNA)*, volume 1, pages 310–313. IEEE.

Joint Planning and Development Office (2008). Next Gen air transportation system integrated work plan. Technical report.

Karp, R. M. (1972). Reducibility among combinatorial problems. In Miller, R. E., Thatcher, J. W., and Bohlinger, J. D., editors, *Complexity of Computer Computations*, pages 85–103. Springer US.

Lehouillier, T., Omer, J., Soumis, F., and Allignol, C. (2014). Interactions between operations and planning in air traffic control. In *Proceedings of the 2nd International Conference on Research in Air Transportation, Istanbul*.

Lehouillier, T., Omer, J., Soumis, F., and Desaulniers, G. (2015a). A flexible framework for solving the air conflict detection and resolution problem using maximum cliques in a graph. *11th USA/Europe Air Traffic Management R&D Seminar, June 23-27, Lisbon, Portugal*.

Lehouillier, T., Omer, J., Soumis, F., and Desaulniers, G. (2015b). A new variant of the minimum-weight maximum-cardinality clique problem to solve conflicts between aircraft. In Le Thi, H. A., Pham Dinh, T., and Nguyen, N. T., editors, *Modelling, Computation and Optimization in Information Systems and Management Sciences*, volume 359 of *Advances in Intelligent Systems and Computing*, pages 3–14. Springer International Publishing.

Martín-Campo, F. J. (2010). *The collision avoidance problem: Methods and algorithms*. PhD thesis, Universidad Rey Juan Carlos, Madrid.

Meng, G. and Qi, F. (2012). Flight conflict resolution for civil aviation based on ant colony optimization. In *Fifth International Symposium on Computational Intelligence and Design (ISCID)*, volume 1, pages 239–241. IEEE.

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

Omer, J. and Farges, J.-L. (2013). Hybridization of nonlinear and mixed-integer linear programming for aircraft separation with trajectory recovery. *IEEE Transactions on Intelligent Transportation Systems*, 14(3):1218–1230.

Paielli, R. A. (2003). Modeling maneuver dynamics in air traffic conflict resolution. *Journal of Guidance, Control, and Dynamics*, 26(3):407–415.

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

Raghunathan, A. U., Gopal, V., Subramanian, D., Biegler, L. T., and Samad, T. (2004). Dynamic optimization strategies for three-dimensional conflict resolution of multiple aircraft. *Journal of Guidance, Control, and Dynamics*, 27(4):586–594.

Resmerita, S., Heymann, M., and Meyer, G. (December 2003). A framework for conflict resolution in air traffic management. In *42nd IEEE Conference on Decision and Control*, volume 2, pages 2035–2040. IEEE.

Schouwenaars, T. (2006). *Safe trajectory planning of autonomous vehicles*. PhD thesis, Massachusetts Institute of Technology.

SESAR Joint Undertaking (2012). European ATM Master Plan, 2nd edition. Technical report.

Sherali, H. D., Cole Smith, J., and Trani, A. A. (2002). An airspace planning model for selecting flight-plans under workload, safety, and equity considerations. *Transportation Science*, 36(4):378–397.

Taillard, É. D. and Voss, S. (2002). Popmusic — partial optimization metaheuristic under special intensification conditions. In *Operations Research/Computer Science Interfaces Series*, pages 613–629. Springer Science + Business Media.

Vela, A., Solak, S., Clarke, J.-P., Singhose, W. E., Barnes, E. R., and Johnson, E. L. (2011). Near real-time fuel-optimal en route conflict resolution. *IEEE Transactions on Intelligent Transportation Systems*, 12(1):47–57.

Vela, A., Solak, S., Singhose, W. E., and Clarke, J.-P. (16–18 December 2009). A mixed integer program for flight-level assignment and speed control for conflict resolution. In *Proceedings of the 48th IEEE Conference on Decision and Control*, pages 5219–5226, Shanghai, China.

Vela, A. E. (2011). *Understanding conflict-resolution taskload: Implementing advisory conflict-detection and resolution algorithms in an airspace*. PhD thesis, Georgia Institute of Technology.

Vivona, R., Karr, D., and Roscoe, D. (2006). Pattern-based genetic algorithm for airborne conflict resolution. In *AIAA Guidance, Navigation and Control Conference and Exhibit, Keystone, Colorado*.

Wu, Q. and Hao, J.-K. (2015). A review on algorithms for maximum clique problems. *European Journal of Operational Research*, 242(3):693–709.

Zhou, K., Doyle, J. C., Glover, K., et al. (1996). *Robust and optimal control*, volume 40. Prentice Hall New Jersey.