



**HAL**  
open science

# **A New Variant of the Minimum-Weight Maximum-Cardinality Clique Problem to Solve Conflicts between Aircraft**

Thibault Lehouillier, Jérémy Omer, François Soumis, Guy Desaulniers

► **To cite this version:**

Thibault Lehouillier, Jérémy Omer, François Soumis, Guy Desaulniers. A New Variant of the Minimum-Weight Maximum-Cardinality Clique Problem to Solve Conflicts between Aircraft. *Modelling, Computation and Optimization in Information Systems and Management Sciences*, pp.3-14, 2015, *Advances in Intelligent Systems and Computing*, <10.1007/978-3-319-18161-5\_1>. <hal-02099567>

**HAL Id: hal-02099567**

**<https://hal.science/hal-02099567v1>**

Submitted on 4 Feb 2022

**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.



HAL Authorization

# A New Variant of the Minimum-Weight Maximum-Cardinality Clique Problem to Solve Conflicts between Aircraft

Thibault Lehouillier, Jérémy Omer, François Soumis, Guy Desaulniers

Group on Research in Decision Analysis  
3000, Côte-Sainte-Catherine Road  
Montreal, QC H3T 2A7, Canada  
{thibault.lehouillier, jeremy.omer, francois.soumis, guy.desaulniers}@polymtl.ca

**Abstract.** In this article, we formulate a new variant of the problem of finding a maximum clique of minimum weight in a graph applied to the detection and resolution of conflicts between aircraft. The innovation of the model relies on the cost structure: the cost of the vertices cannot be determined a priori, since they depend on the vertices in the clique. We apply this formulation to the resolution of conflicts between aircraft by building a graph whose vertices correspond to a set of maneuvers and whose edges link conflict-free maneuvers. A maximum clique of minimal weight yields a conflict-free situation involving all aircraft and minimizing the costs induced. We solve the problem as a mixed integer linear program. Simulations on a benchmark of complex instances highlight computational times smaller than 20 seconds for situations involving up to 20 aircraft.

**Keywords:** Air Traffic Control, Conflict Resolution, Maximum Clique, Mixed Integer Linear Programming

## 1 Introduction

Developing advanced decision algorithms for the air traffic control (ATC) is of great importance for the overall safety and capacity of the airspace. Resolution algorithms for the air conflict detection and resolution problem are relevant especially in a context of growing traffic, where capacity and safety become an issue. Indeed, a simulation-based study performed by Lehouillier et al. [1] shows that the controllers in charge of the traffic in 2035, which will have increased by 50%, would have to solve on average 27 conflicts per hour in a busy sector.

Maintaining separation between aircraft is usually referred to as the air conflict detection and resolution (CDR) problem. A conflict is a predicted loss of separation, i.e., when two aircraft are too close to each other regarding predefined horizontal and vertical separation distances of 5NM and 1000ft respectively. To solve a conflict, the controllers issue maneuvers that can consist of speed, heading or altitude changes. Given the current position, speed, acceleration and the

predicted trajectory of a set of aircraft, the CDR problem corresponds to identifying the maneuvers required to avoid all conflicts while minimizing the costs induced.

The CDR problem is one of the most widely studied problems in air traffic management. For a comprehensive coverage of the existing literature, the reader may refer to the review in Martín-Campo thesis [2]. Exact methods include optimal control, which can be associated with nonlinear programming. However, these methods suffer from the sensitivity to the starting point of the resolution and the high computational time. Mixed integer linear and nonlinear programming (MILPs and MINLPs) techniques are often considered. Omer and Farges [3] present a time-discretization of optimal control. Omer [4] also develops a space discretization using the points of interest for the conflict resolution. Pallottino et al. [5] develop MILPS solving the problem with speed changes and constant headings or with heading changes and constant speeds. Alonso-Ayuso et al. [6] develop a MILP that considers speed and altitude changes. However, MINLPs suffer from high computational times and do not give any optimality guarantee in finite time. Besides, the hypotheses made in MILPs to have linear constraints may not work in all situations. Several heuristics were developed to find a solution rapidly. Examples of techniques developed include ant colony algorithms like in Durand and Alliot [7], variable neighborhood searches (see Alonso-Ayuso et al. [8]). Other fast methods include particle swarm optimization, prescribed sets or neural networks. Heuristics find a solution rapidly, but the hypotheses can be restrictive and the convergence is not guaranteed. Graph theory is seldom used in ATC. Generally, conflicts between aircraft are modeled by a graph whose vertices represent the different aircraft and whose edges link pairs of conflicting aircraft, like in Vela [9]. Barnier and Brisset [10] assign flight levels to aircraft with intersecting routes by looking for maximum cliques in a graph where a proper coloring of the vertices defines an assignment of all aircraft to a set of flight levels.

The model presented in this article uses the concept of a clique in a graph, which is a subset of the vertices where each pair of elements is linked by an edge. Finding a maximum clique in an arbitrary graph is a well-known optimization problem that is  $\mathcal{NP}$ -hard. The problem has been thoroughly studied and several methods, both exact and heuristic, have been developed. For a comprehensive coverage on the subject, one can refer to Bomze et al. [11] and Hao et al. [12].

We formulate the air conflict detection and resolution problem as a new variant of the problem of finding a maximum clique of minimum weight in a graph. To this end, we build a graph whose vertices represent a set of possible maneuvers and where a clique yields a conflict-free solution involving all the aircraft. On the one hand, our model is innovative due to the cost structure for the vertices. With this model, we can maintain a reasonable size for the graph built, hence reducing the computational time. On the other hand, our model significance relies on its flexibility: a modification of the problem constraints or objective function do not jeopardize the validity of the mathematical framework developed. Being flexible is critical in ATC: in addition to being able to cover

more ground, it will allow meaningful comparisons with existing models in the literature.

## 2 Problem Formulation

### 2.1 Modeling aircraft dynamics

To model the flight dynamics, we use the three-dimensional point-mass model presented in the BADA user manual [13]. Aircraft follow their planned 4D trajectory, which is a sequence of 4D points requiring time and space accuracy, leaving the remainder of the trajectory almost unconstrained. The non-compliance with this contract costs penalty fees to companies. As a consequence, an aircraft needs to recover its initial 4D trajectory after performing a maneuver. We assume that the planned speed for an aircraft corresponds to its nominal speed, i.e., the speed minimizing the fuel burn rate per distance unit traveled using the model described in [13].

Maneuvers are performed dynamically as described in [14], where the author states that the typical acceleration during a speed adjustment is in the order of 0.4kn/s. Heading changes are approximated by a steady turn of constant rate and radius. The changes of flight level are performed with a vertical speed, whose computation is detailed in [13], as a function of the thrust, drag, and true airspeed.

### 2.2 On cliques and stables

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

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

$$\mathcal{C} \subseteq \mathcal{V} \text{ is a clique} \Leftrightarrow \forall (u, v) \in \mathcal{C} \times \mathcal{C}, (u, v) \in \mathcal{E} \quad (1)$$

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 *clique number* and is denoted by  $w(\mathcal{G})$ . Let  $c : \mathcal{V} \rightarrow \mathbb{R}$  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 no two of which are adjacent. A *bipartite* graph is a graph whose vertices can be partitioned into two distinct stable sets  $\mathcal{V}_1$  and  $\mathcal{V}_2$ . Each edge of the graph connects one vertex of one stable to a vertex in the other stable. This concept is extended to *k-partite* graphs, where the vertex set is partitioned into  $k$  distinct stable sets.

### 2.3 Graph construction

In this subsection, we introduce the graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  used to model the CDR problem.

**Defining the vertices** Let  $\mathcal{F} = \llbracket 1; n \rrbracket$  denote the set of the considered aircraft. We define  $\mathcal{M} = \cup_{f=1}^n \mathcal{M}_f$  as the set of the possible maneuvers,  $\mathcal{M}_f$  being the set of maneuvers for aircraft  $f \in \mathcal{F}$ . We consider both horizontal and vertical maneuvers of the following types:

- *NIL* refers to the *null* maneuver, i.e., when no maneuver is performed;
- $H_\theta$  corresponds to a heading change by an angle  $\theta \in [-\pi; \pi]^1$ ;
- $S_\delta$  corresponds to a relative speed change of  $\delta\%$ ;
- $V_{\delta h}$  denotes a change of  $\delta h$  flight levels.

A maneuver  $m \in \mathcal{M}$  is described as a triplet  $(\delta\chi_m, \delta V_m, \delta FL_m)$  corresponding to the heading, speed and flight level changes induced by  $m$ . The set of vertices is defined as  $\mathcal{V} = \llbracket 1; |\mathcal{M}| \rrbracket^2$ . We note  $\mathcal{V}_f$  the set of vertices corresponding to aircraft  $f$ .

In emergency scenarios where the feasibility of the problem can be an issue, it is possible to introduce  $n$  vertices corresponding to costly emergency maneuvers to ensure the feasibility of the problem. However, since the feasibility was not an issue for the tested instances, those vertices were not considered in this article. The weight of the vertices correspond to the fuel consumption induced by the corresponding maneuvers. We give further detail in Subsection 2.3.

**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  $(f_i, f_j) \in \mathcal{F} \times \mathcal{F}$ . For  $i \neq j$ , we write  $m_i \square m_j$  when no conflict occurs if aircraft  $f_i$  follows maneuver  $m_i$  while aircraft  $f_j$  performs maneuver  $m_j$ . The set of edges  $\mathcal{E}$  corresponds to the pairs of maneuvers performed by two different aircraft without creating conflicts:

$$\mathcal{E} = \{(i, j) \in \mathcal{V} \times \mathcal{V}, i \neq j : m_i \square m_j\} \quad (2)$$

It is important to note that there is no edge between two different maneuvers of a given aircraft, which yields Proposition 1.

**Proposition 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.*

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  $(f_i, f_j) \in \mathcal{F} \times \mathcal{F}$ . The methodology used to compute if the edge  $(i, j)$  is added to  $\mathcal{G}$  is described with the following notations:

- $\mathcal{T}$ : time horizon for the conflict resolution;
- $\mathbf{p}_{f_i}(t) \in \mathbb{R}^3$ : position vector of aircraft  $f_i$  at time  $t$ .  $p_{f_i,x}(t)$ ,  $p_{f_i,y}(t)$  and  $p_{f_i,z}(t)$  denote respectively the abscissa, ordinate and altitude components of the position vector;
- $\mathbf{s}_{f_i}(t) \in \mathbb{R}^3$ : speed vector of aircraft  $f_i$  at time  $t$ .  $s_{f_i,x}(t)$ ,  $s_{f_i,y}(t)$  and  $s_{f_i,z}(t)$  denote respectively the abscissa, ordinate and altitude components of the speed vector;

<sup>1</sup> positive angles correspond to counter-clockwise rotations

<sup>2</sup>  $|\mathcal{M}|$  is the cardinality of set  $\mathcal{M}$

- $\mathbf{a}_{f_i}(t) \in \mathbb{R}^3$ : acceleration vector of aircraft  $f_i$  at time  $t$ .  $a_{f_i,x}(t)$ ,  $a_{f_i,y}(t)$  and  $a_{f_i,z}(t)$  denote respectively the abscissa, ordinate and altitude components of the acceleration vector;
- $\mathbf{p}_{f_j}(t)$ ,  $\mathbf{s}_{f_j}(t)$  and  $\mathbf{a}_{f_j}(t)$  are also defined following the same notations.

The definition of the maneuvers  $m_i$  and  $m_j$  applied to  $f_i$  and  $f_j$  is used to project the aircraft trajectory over time. Aircraft  $f_i$  and  $f_j$  are said to be separated at time  $t$  if and only if at least one of constraints (3) and (4) holds:

$$d_{f_i f_j}^h(t)^2 = (p_{f_i,x}(t) - p_{f_j,x}(t))^2 + (p_{f_i,y}(t) - p_{f_j,y}(t))^2 \geq D_{h,\min}^2 \quad (3)$$

$$d_{f_i f_j}^v(t)^2 = (p_{f_i,z}(t) - p_{f_j,z}(t))^2 \geq D_{v,\min}^2 \quad (4)$$

3

At any time  $t \in \mathcal{T}$ , either none, one or both aircraft are maneuvering.  $\mathcal{T}$  can thus be divided into intervals where both  $f_i$  and  $f_j$  have a constant acceleration. For each interval, we compute the time at which the aircraft are the closest to verify if the separation constraints hold. Let  $\mathcal{T}_k$  be one of these intervals. Consider  $f_i$  and  $t_0 \in \mathcal{T}$  be the starting time of maneuver  $m_i$ . If we assume that maneuver  $m_i$  is applied with a constant acceleration, we obtain the position and the speed vector of  $f_i$  at time  $t_0 + t$  with  $t$  such that  $t - t_0 \leq |\mathcal{T}_k|$ :

$$\mathbf{p}_{f_i}(t_0 + t) = \mathbf{p}_{f_i}(t_0) + (t - t_0)\mathbf{s}_{f_i}(t_0) + \frac{(t - t_0)^2}{2}\mathbf{a}_{f_i}(t_0) \quad (5)$$

$$\mathbf{s}_{f_i}(t_0 + t) = \mathbf{s}_{f_i}(t_0) + (t - t_0)\mathbf{a}_{f_i}(t_0) \quad (6)$$

Let  $\mathbf{p}_{f_i f_j}^h$  (respectively  $\mathbf{s}_{f_i f_j}^h$ ,  $\mathbf{a}_{f_i f_j}^h$ ) denote respectively the horizontal position, the speed and the acceleration of aircraft  $f_j$  relatively to aircraft  $f_i$ . We define

$$\begin{aligned} d_{f_i f_j}^h(t + \tau) &= \|\mathbf{p}_{f_i f_j}^h(t + \tau)\| \\ &= \|\mathbf{p}_{f_i f_j}^h(t) + \tau\mathbf{s}_{f_i f_j}^h(t) + \frac{\tau^2}{2}\mathbf{a}_{f_i f_j}^h(t)\| \end{aligned}$$

where  $\tau \geq 0$ .

Let  $\tau_{f_i f_j} \in \operatorname{argmin}_{\tau \geq 0} d_{f_i f_j}^h(t + \tau)^2$ , and  $t_{f_i f_j}^h \in \operatorname{argmin}_{t \in \mathcal{T}} d_{f_i f_j}^h(t)^2$ .

$$\text{We have: } t_{f_i f_j}^h = \begin{cases} 0 & \text{if } \tau_{f_i f_j} = 0 \\ |\mathcal{T}| & \text{if } \tau_{f_i f_j} \geq |\mathcal{T}_k| \\ \tau_{f_i f_j} & \text{otherwise} \end{cases}$$

Aircraft  $f_i$  and  $f_j$  are horizontally separated during interval  $\mathcal{T}$  if and only if (7) holds:

$$d_{f_i f_j}^h(t_{f_i f_j}^h)^2 \geq D_{h,\min}^2 \quad (7)$$

By a similar reasoning, aircraft  $f_i$  and  $f_j$  are vertically separated during interval  $\mathcal{T}$  if and only if (8) holds:

$$d_{f_i f_j}^v(t_{f_i f_j}^v)^2 \geq D_{v,\min}^2 \quad (8)$$

<sup>3</sup> In this paper we choose  $D_{h,\min} = 5\text{NM}$  and  $D_{v,\min} = 1000\text{ft}$ .

If either (7) or (8) holds when aircraft  $f_i$  and  $f_j$  apply maneuvers  $m_i$  and  $m_j$ , then an edge is created between  $i$  and  $j$ . As explained in 2.1, it is important that every aircraft initiates a safe return towards its initial trajectory once the conflict is avoided. For each edge, we compute the minimum time necessary before one or both aircraft can recover their initial trajectories. The cost of the recovery of a trajectory is detailed in Subsection 2.3.

**Application to the CDR problem** As mentioned in Section 1, given the current position, speed, acceleration and the planned trajectories of a set of aircraft, solving the CDR problem consists in finding a conflict-free set of maneuvers that minimizes the costs. Proposition 2 links the cliques in  $\mathcal{G}$  to the CDR problem:

**Proposition 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  $|\mathcal{C}|$ .*

Proposition 2 shows 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 derive the following theorem:

**Theorem 1.** *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.*

We define the problem  $\text{CDR}_{\mathcal{M}}$  as the restriction of the CDR problem to the set of maneuvers  $\mathcal{M}$ . Using both Proposition 2 and Theorem 1, we can state anew the  $\text{CDR}_{\mathcal{M}}$  problem as follows: solving the  $\text{CDR}_{\mathcal{M}}$  problem consists in finding a clique of maximum cardinality and minimal cost in graph  $\mathcal{G}$ . In fact, we consider a new variant of a clique problem where the weight associated with a vertex is not known a priori and rather depends on the edges induced by the clique. Indeed, the cost associated with a maneuver depends on the duration that this maneuver will be performed before returning towards the planned trajectory. Because this duration depends on the maneuvers selected for the other aircraft, it cannot be determined a priori and must be computed as the maximum duration needed to avoid a loss of separation with all other aircraft given their chosen maneuvers. To handle such vertex costs, we first define edge costs.

**Computing the cost of the edges** The cost measure chosen for this article corresponds to the extra fuel consumption induced by the maneuvers, i.e., the additional fuel required to return to the 4D trajectory after the maneuver is performed. We use the model given in [13]. For a jet commercial aircraft  $f$ , the fuel consumption by time and distance unit is given by (9) and (10):

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

$$C_{d,f}(t) = \frac{C_{t,f}(t)}{V_f(t)} \quad (10)$$

where  $c_{1,f}$  and  $c_{2,f}$  are numerical constants depending on the type of aircraft  $f$ .

We compute the cost of an edge  $e = (i, j)$  linking two vertices representing two maneuvers of aircraft  $f_i$  and  $f_j$ , denoted  $m_i$  and  $m_j$ , as a pair constituted of the extra fuel costs for both  $f_i$  and  $f_j$ , denoted  $C_i^{(i,j)}$  and  $C_j^{(i,j)}$ . The additional consumed fuel corresponds to the performed maneuver along with the fuel required to recover the initial 4D trajectory. After a change of speed of  $\delta\%$  during a period  $\delta t$ , the aircraft recovers its 4D trajectory by making the opposite change of speed during  $\delta t$ . After a change of direction  $\delta\chi$  during a period  $\delta t$ , the aircraft performs a turn with an angle  $\theta_r$  in order to recover its physical trajectory along with a change of speed to retrieve the 4D trajectory. The cost induced is the extra fuel burnt when the aircraft flies at the recovery speed and the fuel burnt on the extra distance induced by the maneuver. For a flight level change, we compute the extra cost as the difference of consumption between the different flight levels, along with the cost of changing twice of flight level. The distance flown is also longer, and this extra distance is also accounted for.

**Computing the cost of the vertices** Several techniques can be followed in order to determine the vertices cost. The basic one would be to discretize the duration of the maneuver, and to create the vertices accordingly. In this situation, computing the costs would be straight-forward. However, the drawback of this method is that the graph built is huge, which could result in a difficult resolution. We choose to follow another structure of cost because it is more compact in terms of graph size.

Let us consider a vertex  $i$  which corresponds to a maneuver  $m_i$  for an aircraft  $f_i$ . The cost of each edge linking  $i$  to one of its neighbors  $j$ , associated to a maneuver  $m_j$  for aircraft  $f_j$ , corresponds to  $f_i$  applying  $m_i$  during a time  $t_i^j$ , which depends on  $m_j$ . Time  $t_i^j$  is the minimum time during which  $f_i$  must apply  $m_i$  in order to avoid any conflict if one or both aircraft return to their initial trajectory. Following maneuver  $m_i$  for a duration  $t_i^j$  induces a cost  $C_i^{(i,j)}$ . If  $i$  is part of the maximum clique  $\mathcal{C}$  to be determined, we need to establish the time  $t_i$  during which maneuver  $m_i$  is actually applied in order to determine its cost  $c_i$ .  $t_i$  is obtained by:

$$t_i = \max_{j \in \mathcal{V} \cap \mathcal{C}} t_i^j \quad (11)$$

As a consequence, we have that  $c_i$  is the cost of aircraft  $f_i$  applying  $m_i$  during  $t_i$ . If  $i$  is not part of the maximum clique  $\mathcal{C}$ , then no constraint is imposed on the cost  $c_i$ . As detailed in Section 3, the optimization model will automatically force the value of  $c_i$  to 0. To conclude, we have that for any  $i \in \mathcal{V}$ :

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

## 2.4 Illustrative example

For the sake of clarity, an illustrative example with three aircraft is given in Figure 1. If each aircraft follows its planned trajectory, conflicts will happen

between the blue aircraft and the two others. For this example, we assume that, in addition to the null maneuver, only two heading changes ( $\pm 30^\circ$ ) are allowed. We build the CDR graph shown in Figure 1(b). The graph is 3-partite, as the vertex set is partitionned into 3 stable sets of 3 vertices each. Solving the CDR is then equivalent to searching for a minimum-weight clique of 3 vertices, i.e., a triangle.

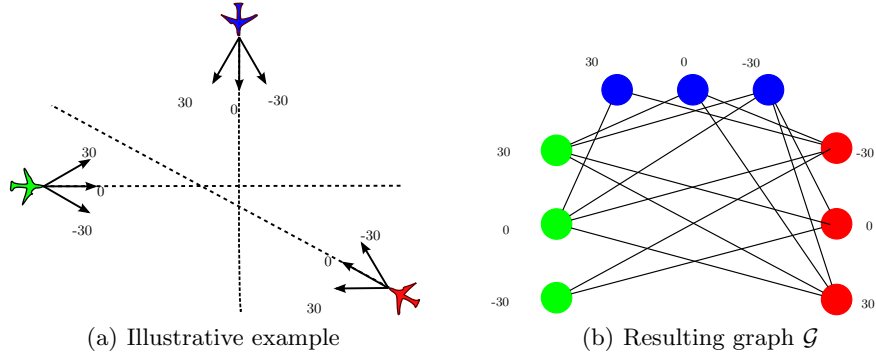


Fig. 1: Illustrative example with three aircraft

### 3 Methodology

Determining the cost of a vertex  $i$  is very specific, since it is correlated to whether or not  $i$  belongs to a maximum clique  $\mathcal{C}$ . As a consequence, the algorithms usually used in existing librairies dedicated to graph theory cannot be used for our model. We formulate the problem as a mixed-integer linear program using the following variables:

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

We describe the clique search by the following linear integer program:

$$\text{minimize } \sum_{i \in \mathcal{V}} c_i \quad (12)$$

$$\text{subject to } x_i + x_j \leq 1, \forall (i, j) \notin \mathcal{E} \quad (13)$$

$$\sum_{i \in \mathcal{V}} x_i = |\mathcal{F}| \quad (14)$$

$$c_i \geq C_i^{(i,j)}(x_i + x_j - 1), \forall (i, j) \in \mathcal{E} \quad (15)$$

$$x_i \in \{0, 1\}, \forall i \in \mathcal{V} \quad (16)$$

$$c_i \in \mathbb{R}_+, \forall i \in \mathcal{V} \quad (17)$$

The objective function (12) minimizes the cost of the maneuvers. (13) are clique constraints, and constraint (14) exploits Theorem 1 defining the cardinality of the maximum clique. Constraints (15) are used to compute the cost of the vertices: if a vertex is in the maximum clique, then its cost must be greater than its cost on all edges connecting it to other vertices in the clique.

## 4 Results

All tests were performed on a computer equipped with an Intel Core i7-3770 processor, 3.4 GHz, 8-GB RAM. The algorithms were implemented in C++ and using CPLEX 12.5.1.0<sup>4</sup>.

The headings of the tables presented in these section are given as follows:

- *case*: case configuration;
- $|\mathcal{F}|$ : number of aircraft;
- $|\mathcal{V}|$ : number of vertices;
- $|\mathcal{E}|$ : number of edges;
- $d = \frac{2|\mathcal{E}|}{|\mathcal{V}|(|\mathcal{V}|-1)}$ : graph density;
- $n$ : number of variables;
- $m$ : number of constraints;
- $z_{ip}$ : optimal value for the problem;
- *nodes*: number of branch-and-bound nodes;
- $t_{lp}$ : time (in seconds) to continuous relaxation of the MILP;
- $t_{ip}$ : time (in seconds) to obtain the  $z_{ip}$  value;

### 4.1 Benchmark description

The benchmark used for this study gathers three types of instances. The first set is roundabout instances  $\mathcal{R}_n$ , where  $n$  aircraft are distributed on the circumference of a 100NM radius and fly towards the center at the same speed and altitude. The second set is crossing flow instances  $\mathcal{F}_{n,\theta,d}$ , where two trails of  $n$  aircraft separated by  $d$  nautical miles intersect each other with an angle  $\theta$ . The last type of instance is a grid  $\mathcal{G}_{n,d}$  constituted of two crossing flow instances  $\mathcal{F}_{n,\frac{\pi}{2},d}$  with a  $90^\circ$  angle, one instance being translated 15NM North-East from the other. An example of these instances is given on Figure 2.

### 4.2 Computational results

The first set of simulations considers only horizontal maneuvers, with relative speed changes of  $\pm 3\%$  and  $\pm 6\%$  and heading changes of  $\pm 5^\circ$ ,  $\pm 10^\circ$ ,  $\pm 15^\circ$ . The graph remains small when one considers this set of maneuvers, and their small magnitude makes them less costly. Nevertheless, if these values were to be inefficient to solve all the conflicts, we could introduce maneuvers of larger magnitude.

<sup>4</sup> see the IBM-ILOG CPLEX v12.5. User's manual for CPLEX

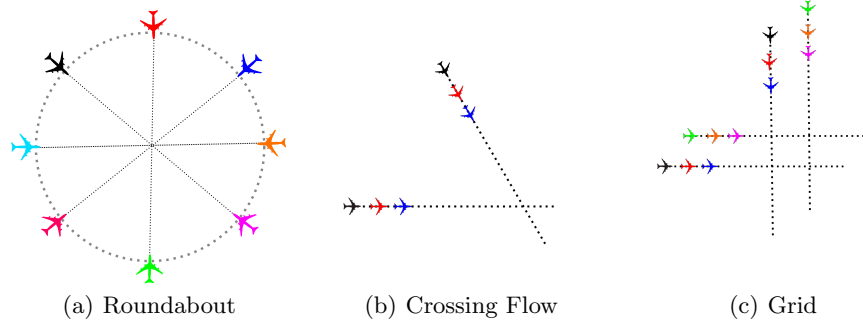


Fig. 2: Examples

Table 1 gathers information about the graph  $\mathcal{G}$ , the MILP and the main computational results. The solution time for the continuous relaxation is very small, but the quality of the relaxation is mediocre. Indeed, the fractional solution of the linear relaxation chooses two maneuvers for each aircraft with a value of 0.5. Constraints (15) force the cost of each vertex to be 0, yielding an optimal value of 0 and a gap of 100%. Results also display short solution times: problems known to be complex with 20 aircraft are solved to optimality in less than 15 seconds. This result is very satisfying since the density of the graph is high.

Instance type	Case	Graph $\mathcal{G}$				MILP		Resolution			
		$ \mathcal{F} $	$ \mathcal{V} $	$ \mathcal{E} $	$d$	$m$	$n$	$z_{ip}$	nodes	$t_{lp}$	$t_{ip}$
Roundabout	$\mathcal{R}_2$	2	22	90	0.39	44	225	3.71	6	0	0.02
	$\mathcal{R}_4$	4	44	492	0.52	88	1073	14.98	73	0	0.02
	$\mathcal{R}_6$	6	66	1194	0.56	132	2521	22.7	0	0.01	0.15
	$\mathcal{R}_8$	8	88	2184	0.57	176	4545	31.05	47	0.01	0.53
	$\mathcal{R}_{10}$	10	110	3430	0.57	220	7081	112.7	208	0.05	1.56
	$\mathcal{R}_{12}$	12	132	4944	0.57	264	10153	189.27	581	0.09	3.41
	$\mathcal{R}_{14}$	14	154	6720	0.57	308	13749	224.75	183	0.1	6.98
	$\mathcal{R}_{16}$	16	176	8896	0.57	352	18145	261.44	162	0.15	9.5
	$\mathcal{R}_{18}$	18	198	11358	0.58	396	23113	636.7	257	0.21	12.1
$\mathcal{R}_{20}$	20	220	1402	0.58	440	28461	740.6	210	0.27	3.2	
Flows	$\mathcal{F}_{5,30,10}$	10	110	4522	0.75	220	9265	49.08	405	0.02	1.5
	$\mathcal{F}_{5,45,10}$	10	110	4518	0.75	220	9257	41.29	535	0.02	1.52
	$\mathcal{F}_{5,60,10}$	10	110	4478	0.75	220	9177	34.49	238	0.02	1.39
	$\mathcal{F}_{5,75,10}$	10	110	4492	0.75	220	9205	30.66	496	0.02	1.34
	$\mathcal{F}_{5,90,10}$	10	110	4528	0.76	220	9277	28.28	269	0.02	1.41
Grids	$\mathcal{G}_{2,3,10}$	12	132	6645	0.78	264	13555	57.65	564	0.01	3.64
	$\mathcal{G}_{2,5,10}$	20	220	19724	0.82	440	39889	121.92	2740	0.2	12.7

Table 1: Dimensions of the instances and computational results

In the second simulation set, we introduce altitude maneuvers: aircraft are allowed to move to an adjacent flight level. Table 2 reports the main results. The values of the optimal solutions for the roundabout instances remain the same, highlighting that it is optimal to make simple turns instead of changing flight levels. For the crossing flows and the grid instances, it is more efficient for some aircraft to change their flight level instead of turning or changing their speed. As a consequence, the solutions are less expensive. Solution times tend to slightly increase, but the solution can still be computed in a short time. These results are promising since the instances tested are denser than real-life instances.

Instance type	Case	Graph $\mathcal{G}$				MILP		Resolution			
		$ \mathcal{F} $	$ \mathcal{V} $	$ \mathcal{E} $	$d$	$m$	$n$	$z_{ip}$	$nodes$	$t_{ip}$	$t_{ip}$
Roundabout	$\mathcal{R}_2$	2	26	116	0.36	52	285	3.71	7	0	0.05
	$\mathcal{R}_4$	4	52	832	0.63	104	1769	14.98	153	0	0.07
	$\mathcal{R}_6$	6	78	2076	0.69	156	4309	22.7	440	0.01	0.19
	$\mathcal{R}_8$	8	104	3840	0.72	208	7889	31.05	245	0.01	0.83
	$\mathcal{R}_{10}$	10	130	6080	0.73	260	12421	112.7	375	0.05	1.4
	$\mathcal{R}_{12}$	12	156	9096	0.75	312	18505	189.27	648	0.09	3.11
	$\mathcal{R}_{14}$	14	182	12208	0.74	364	24781	224.75	256	0.1	6.46
	$\mathcal{R}_{16}$	16	208	16416	0.76	416	33249	261.44	210	0.15	9.08
	$\mathcal{R}_{18}$	18	234	20772	0.76	468	42013	636.7	289	0.21	12.1
$\mathcal{R}_{20}$	20	260	25760	0.77	520	52041	740.6	223	0.27	6.5	
Flows	$\mathcal{F}_{5,30,10}$	10	110	4522	0.75	220	9265	46.12	401	0.02	1.5
	$\mathcal{F}_{5,45,10}$	10	110	4518	0.75	220	9257	40.12	588	0.02	1.52
	$\mathcal{F}_{5,60,10}$	10	110	4478	0.75	220	9177	31.69	324	0.02	1.39
	$\mathcal{F}_{5,75,10}$	10	110	4492	0.75	220	9205	30.11	542	0.02	1.34
	$\mathcal{F}_{5,90,10}$	10	110	4528	0.76	220	9277	26.12	287	0.02	1.41
Grids	$\mathcal{G}_{2,3,10}$	12	132	6645	0.78	264	13555	45.18	612	0.01	3.64
	$\mathcal{G}_{2,5,10}$	20	220	19724	0.82	440	39889	108.12	2910	0.2	12.7

Table 2: Dimensions of the instances and computational results

## 5 Conclusions

A new variant of the problem of finding a maximum clique of minimum weight in a graph and its application to aircraft conflict resolution have been presented. The innovation of the model comes from the cost structure: the costs of the vertices cannot be determined a priori since they depend on the vertices in the clique. As a consequence, we model the problem as a MILP. The model performs well, since complex instances involving up to 20 aircraft are solved to optimality in near real-time. The design of the model is flexible, meaning that tuning some parameters of the model will allow meaningful comparisons with existing models.

These conclusions validate the model as a basis for further research. For instance, techniques reducing the size of the graph are of interest. Adding uncertainties is also a meaningful extension of the model. Additional benchmarks

including real-life instance and random scenarios will be necessary in order to challenge the model.

## References

1. T. Lehoullier, J. Omer, F. Soumis, and C. Allignol, “Interactions between operations and planning in air traffic control,” Proceedings of the 2nd International Conference of Research in Air Transportation, Istanbul, 2014.
2. M. Campo and F. Javier, “The collision avoidance problem: methods and algorithms,” Ph.D. dissertation, 2010.
3. J. Omer and J.-L. Farges, “Hybridization of nonlinear and mixed-integer linear programming for aircraft separation with trajectory recovery,” IEEE Transactions on Intelligent Transportation Systems, vol. 14, no. 3, pp. 1218–1230, 2013.
4. J. Omer, “A space-discretized mixed-integer linear model for air-conflict resolution with speed and heading maneuvers,” 2014.
5. L. Pallottino, E. M. Feron, and A. Bicchi, “Conflict resolution problems for air traffic management systems solved with mixed integer programming,” Intelligent Transportation Systems, IEEE Transactions on, vol. 3, no. 1, pp. 3–11, 2002.
6. A. Alonso-Ayuso, L. F. Escudero, F. J. Martín-Campo, and F. Javier, “Collision avoidance in air traffic management: a mixed-integer linear optimization approach,” IEEE Transactions on Intelligent Transportation Systems, vol. 12, no. 1, pp. 47–57, 2011.
7. N. Durand, J.-M. Alliot, and J. Noailles, “Automatic aircraft conflict resolution using genetic algorithms,” Proceedings of the Symposium Applied Computing, Philadelphia, 1996.
8. A. Alonso-Ayuso, L. F. Escudero, F. J. Martín-Campo, and N. Mladenović, “A vns metaheuristic for solving the aircraft conflict detection and resolution problem by performing turn changes,” Journal of Global Optimization, pp. 1–14, 2014.
9. A. E. Vela, “Understanding conflict-resolution taskload: implementing advisory conflict-detection and resolution algorithms in an airspace,” 2011.
10. N. Barnier and P. Brisset, “Graph coloring for air traffic flow management,” Annals of operations research, vol. 130, no. 1-4, pp. 163–178, 2004.
11. I. M. Bomze, M. Budinich, P. M. Pardalos, and M. Pelillo, “The maximum clique problem,” in Handbook of combinatorial optimization. Springer, 1999, pp. 1–74.
12. Q. Wu and J.-K. Hao, “A review on algorithms for maximum clique problems,” European Journal of Operational Research, 2014.
13. “User manual for the Base of Aircraft Data (BADA),” Eurocontrol, Tech. Rep. 11/03/08-08, 2011.
14. R. A. Paielli, “Modeling maneuver dynamics in air traffic conflict resolution,” Journal of guidance, control, and dynamics, vol. 26, no. 3, pp. 407–415, 2003.