Mixed-Integer Nonlinear Programming for Aircraft Conflict Avoidance by Sequentially Applying Velocity and Heading Angle Changes

Sonia Cafieri, Riadh Omheni

Mixed-Integer Nonlinear Programming for Aircraft Conflict Avoidance by Sequentially Applying Velocity and Heading Angle Changes

Sonia Cafieri and Riadh Omheni

ENAC, University of Toulouse, F-31055 Toulouse, France

Abstract

We consider the problem of aircraft conflict avoidance in Air Traffic Management systems. Given an initial configuration of a number of aircraft sharing the same airspace, the main goal of conflict avoidance is to guarantee that a minimum safety distance between each pair of aircraft is always respected during their flights. We consider aircraft separation achieved by heading angle deviations, and propose a mixed 0-1 nonlinear optimization model, that is then combined with another one which is based on aircraft speed regulation. A two-step solution approach is proposed, where the two models are sequentially solved using a state-of-the-art mixed-integer nonlinear programming solver. Numerical results validate the proposed approach and clearly show the benefit of combining the two considered separation maneuvers.

Keywords Air traffic management, Conflict avoidance, Mixed-integer nonlinear programming, Deterministic global optimization, Modeling, MINLP

1 Introduction

We address a real-life problem arising in the context of Air Traffic Management (ATM), and more specifically in Air Traffic Control, that is the problem of aircraft conflict avoidance during en-route flights. Two aircraft are said

to be in *conflict* if during their flights there is a loss of a standard separation between their trajectories, that has to be always respected instead to guarantee air traffic safety. The main aim of Air Traffic Management is that of ensuring that aircraft are always safely guided in the sky and on the ground, and aircraft conflict avoidance constitutes one the major challenges to be faced. This is specially due to safety issues in a context of growing air traffic on the world scale. Air traffic in Europe, for example, is expected to double within the next two decades [1]. This implies an increasing workload for air traffic controllers, with a consequent growing difficulty in ensuring air traffic safety. Thus, ATM needs more automation in tackling crucial processes like aircraft conflicts resolution, as pointed out in the context of major ATM projects Sesar[2] and NextGen[3].

Developing mathematical models and efficient and reliable algorithms for aircraft conflict avoidance has been the focus of a great deal of research attention in last years. Aircraft conflict avoidance could indeed be probably considered as an emerging application of Operations Research.

Problem statement

Let us consider a set of aircraft sharing the same airspace. Conflict avoidance is performed monitoring a selected portion of the airspace over a given time horizon, and issuing suitable separation maneuvers when a loss of pairwise separation between aircraft is detected. We consider the case of aircraft in the en-route airspace, where the standard separation norms are 5 NM horizontally and 1000 ft vertically (1NM (nautical mile)= 1852 m; 1ft (feet)= 0.3048 m). Furthermore, we consider aircraft that are all allocated to the same flight level, and thus our focus is on horizontal separation maneuvers. Flight level changes to resolve conflicts are in fact rarely employed because of fuel consumption and of passengers discomfort. The problem we address is at a *tactical level*, i.e., aircraft in their en-route cruise flights are monitored in a short time window (20-30 minutes) and conflicts are resolved a short time before the loss of separation potentially occurs (so, a fast solving, preferably in the order of seconds, is generally expected). We are given aircraft with their initial positions in the observed airspace, and with their velocity vectors, that is their speed and their heading angles defining their trajectories. The aim is that of providing, starting from such initial configuration, a new one that is conflict-free for all aircraft.

To solve an aircraft conflict, various maneuvers can be employed, mainly

including altitude changes (AC), heading angle changes (HAC) and velocity changes (VC). Mathematical models are then based on this kind of maneuvers. Optimization problems naturally arise in this context, as in general one aims at separate aircraft while deviating them as less as possible from their original trajectory.

Based on the comprehensive survey conducted by Kuchar and Yang [4], it appears that the majority of detection and resolution methods up to the year 2000 focus on AC and HAC. In particular, the great attention towards heading angle changes is related to the fact that these maneuvers are currently the most used by air traffic controllers. In the last 15 years, some *subliminal* speed control-based models have been proposed. The subliminal speed control, that was firstly introduced in the context of the European project ERASMUS [5], consists of modifying the aircraft speeds within a very small range around their original speeds, without informing air traffic controllers. This kind of control is highly promising to introduce some automation in ATM, thus reducing the workload of air traffic controllers.

In the following, we review some approaches based on mixed-integer programming, that constitutes the framework of the approaches developed in this paper.

Mixed-integer programming literature

Mixed-integer programming approaches have been used in aircraft conflict detection and resolution since the 2000's. An overview of MINLP modeling is presented in [6]. In 2002, Pallottino, Feron and Bicchi [7] proposed two different formulations of the multi-aircraft conflict avoidance problem as a mixed-integer linear program (MILP) by allowing all aircraft to perform either VC or HAC, but not both. These models are then solved by standard MILP software. Christodoulou and Costoulakis [8] proposed a combinatorial approach that combines aircraft velocities and heading angles. The obtained problem is a mixed-integer nonlinear program (MINLP), that is solved for small-scale instances. Vela et al. [9] proposed a mathematical model that is formulated as a MILP and is based on concepts of speed control and flight-level assignment. In a second work, Vela et al. [10] presented a MILP model to identify the required heading angle and speed changes for each aircraft to avoid conflicts. As an objective function, they chose to minimize the fuel costs incurred due to the considered changes. MILP models based on speed control were also proposed in [11, 12] where the total conflict duration

is minimized. Since 2011, Alonso-Ayuso, Escudero and Martín-Campo have proposed three mathematical models which all extend the work of Pallottino, Feron and Bicchi [7]. In [13], the authors presented a mixed 0-1 linear optimization model based on VC and AC. The second model, presented in [14], is a mixed 0-1 nonlinear optimization one where velocity change is used as a maneuver to resolve conflicts. In 2014, a non-convex MINLP model based on turn changes has been developed [15]. Once all conflicts are solved, the aircraft are forced to return to the original flight configuration by solving an unconstrained quadratic program. This feature will be used in our work presented in this paper. Note finally that there is no computational comparison of the performances of above three models. Cafieri and Durand [16] proposed MINLP formulations for deconfliction based on speed regulation, where each aircraft is allowed to fly with a modified speed in a time window. More recently, a MINLP model for maximizing the number of conflicts that can be solved when only velocity regulation is performed has been introduced by Cafieri [17]. In addition to mixed-integer programming, local continuous optimization to air traffic conflict resolution has been proposed. Peyronne et al. [18] presented a B-spline trajectory model involving only one continuous variable per aircraft, and a semi-infinite programming formulation for the separation constraints and showed some numerical results on problems involving up to six conflicting aircraft.

Contribution statement and paper structure

In this paper, we consider aircraft separation achieved by heading angle changes and by speed regulation. First, a mixed-integer nonlinear program (HAC) is proposed, where aircraft conflicts are solved by aircraft heading angle deviations. Then, this model is combined with another mixed 0-1 nonlinear program, that maximizes the number of aircraft conflicts that can be solved by subliminal speed regulation.

These two MINLPs are solved using a state-of-the-art global exact solver. A two-step methodology is then proposed, where the solution of the second MINLP is used as a pre-processing step for the first one. This is shown to be able to significantly speed-up the resolution process. Numerical results clearly show, indeed, the benefit of the proposed combination of the two considered aircraft separation maneuvers, thus validating the proposed approach. Such an approach appears to be, on the other hand, appropriate for the considered real-time application, where subliminal speed control

is expected acting as a first filter for the traffic. Then, the traffic is made fully conflict-free thanks to heading changes, that correspond to maneuvers currently mainly employed by air traffic controllers.

The paper is organized as follows. In Section 2, we provide a general description of the addressed problem and present two MINLP models for aircraft conflict avoidance. A mixed-integer nonlinear program based on HAC is first proposed. Then, a mixed-integer program aiming to maximize the number of solved conflicts using VC is presented. We give thereafter, in Section 3, a detailed description of the proposed algorithm, combining the solution of the above MINLPs. In Section 4, numerical results are presented and some numerical issues are discussed. Conclusions and perspectives for future work are drawn in Section 5.

2 Modeling aircraft conflict avoidance

Let A denote a set of aircraft sharing the same airspace and flying during their cruise phase, all at the same flight level. Each aircraft $i \in A$ is identified by the triplet (x_i, y_i, ϕ_i) giving its abscissa, ordinate and direction of motion in $]-\pi, \pi]$, with (x_i^0, y_i^0, ϕ_i) the initial triplet, at time $t = 0$. The minimum separation distance condition between two aircraft i and $j \in A$ can be expressed as follows:

$$(x_i(t) - x_j(t))^2 + (y_i(t) - y_j(t))^2 \geq d^2 \qquad \forall t, \tag{1}$$

where d is the minimum required horizontal separation distance and $x_i(t)$ and $y_i(t)$ are the coordinates of the aircraft i at time t. When condition (1) is not satisfied for some $t \geq 0$, aircraft i and j are said to be in conflict. In such a case, (at least) one maneuver to separate the aircraft has to be performed. In the following subsection, we focus on separation achieved by aircraft heading angle changes and propose a MINLP model for solving aircraft conflicts.

2.1 MINLP for HAC

Let us consider n aircraft with known initial spatial coordinates and trajectory directions (heading), flying through an air sector during a time window observed by air traffic controllers. Our goal is to determine for each of the n aircraft, $i \in A$, its minimum heading angle change that allows to satisfy the

separation condition (1), thus obtaining a conflict-free aircraft configuration. An example is displayed on Figures 1 and 2.

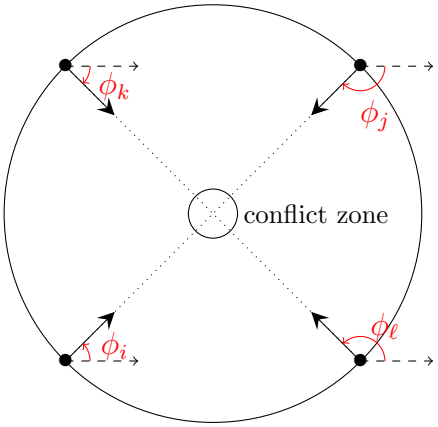

Figure 1: Initial configuration for 4 conflicting aircraft flying towards the center of a circle of a given radius.

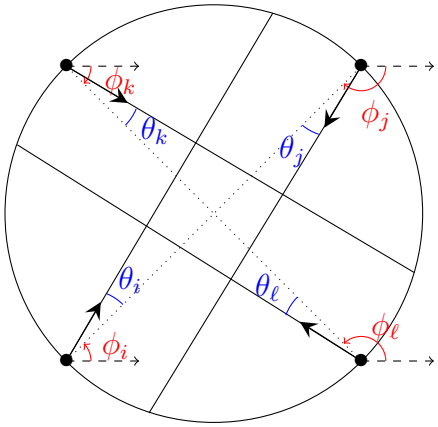

Figure 2: After conflict resolution, the new trajectory of each aircraft is given by a solid line.

The proposed approach, presented in the following, is based on the work of Cafieri and Durand [16]. In that work, aircraft headings were kept fixed, while velocity vectors were modified by changing aircraft speeds to achieve separation. Here, we consider the opposite situation: aircraft speeds are fixed, while we allow aircraft change their headings.

Assume that uniform motion laws apply for aircraft. The abscissa and ordinate of an aircraft i at time t are given by

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

where θ_i is the angle variation for aircraft i for solving the conflict situations and v_i is its initial speed. Note that v_i is fixed for all i, while θ_i, for all i represent the decision variables of the optimization problem that we are going to define. Substituting the expressions of $x_i(t)$ and $y_i(t)$ into (1), we obtain

$$t^2\|V_{ij}^r\|^2 + 2tX_{ij}^{r0} \cdot V_{ij}^r + \|X_{ij}^{r0}\|^2 - d^2 \geq 0, \tag{2}$$

where the relative initial distance X_{ij}^{r0} between aircraft i and j and the rela-

tive speed V_{ij}^r are vectors in $\mathbb{R}^2$ given respectively by

$$X_{ij}^{r0} := \begin{pmatrix} x_i^0 - x_j^0 \\ y_i^0 - y_j^0 \end{pmatrix} \tag{3}$$

and

$$V_{ij}^r := \begin{pmatrix} \cos(\phi_i + \theta_i)v_i - \cos(\phi_j + \theta_j)v_j \\ \sin(\phi_i + \theta_i)v_i - \sin(\phi_j + \theta_j)v_j \end{pmatrix} \tag{4}$$

and $X_{ij}^{r0} \cdot V_{ij}^r$ denotes the inner product of X_{ij}^{r0} and V_{ij}^r. Equation (2) is a quadratic equation in one unknown t. By differentiating its left-hand side member, the time for which the condition (2) is minimal, and so aircraft i and j have minimal relative position, is

$$t_{ij}^m = -\frac{X_{ij}^{r0} \cdot V_{ij}^r}{\|V_{ij}^r\|^2}.$$

By substituting this into (2), we obtain the following expression:

$$\|V_{ij}^r\|^2 \left(\|X_{ij}^{r_0}\|^2 - d^2\right) - (X_{ij}^{r0} \cdot V_{ij}^r)^2 \geq 0, \tag{5}$$

which no longer depends on t. The separation condition (5) is to be imposed only when $t_{ij}^m \geq 0$. To this end, we introduce a new, binary, variable y_{ij} for each pair of aircraft i and j such that

$$y_{ij} = \begin{cases} 1 & \text{if } t_{ij}^m \geq 0, \\ 0 & \text{otherwise.} \end{cases}$$

This yields the following separation condition for aircraft i and j

$$y_{ij} \left(\|V_{ij}^r\|^2 \left(\|X_{ij}^{r_0}\|^2 - d^2\right) - (X_{ij}^{r0} \cdot V_{ij}^r)^2\right) \geq 0.$$

For each pair of aircraft, such a condition constitutes the main constraint of the optimization problem for aircraft deconfliction.
As for the objective function, we choose to minimize for all aircraft their heading angle variations, that is

$$\min \sum_{i \in A} \theta_i^2.$$

The main motivation for this choice is to ensure that flight plans of the considered aircraft do not change significantly with respect to the original

ones. Notice that other objective functions can be considered, such as the minimization of the fuel cost or the total difference in flight time between the original expected time and that needed to perform the HAC maneuvers to avoid all potential conflicts.

The proposed optimization problem is summarized below. For the sake of simplification, we use the notation $B := \{(i,j) \in A \times A : i < j\}$. The decision variables are: θ_i, for all $i \in A$, t_{ij}^m for all $(i,j) \in B$, and y_{ij} for all $(i,j) \in B$.

$$\min \sum_{i \in A} \theta_i^2 \tag{6a}$$

s.t.

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

$$y_{ij} \left(\|V_{ij}^r\|^2 \left(\|X_{ij}^{r_0}\|^2 - d^2 \right) - (X_{ij}^{r0} \cdot V_{ij}^r)^2 \right) \geq 0, \quad \forall\, (i,j) \in B, \tag{6c}$$

$$t_{ij}^m = -\frac{X_{ij}^{r0} \cdot V_{ij}^r}{\|V_{ij}^r\|^2}, \quad \forall\, (i,j) \in B, \tag{6d}$$

$$t_{ij}^m (2y_{ij} - 1) \geq 0, \quad \forall\, (i,j) \in B, \tag{6e}$$

$$y_{ij} \in \{0,1\}, \quad \forall\, (i,j) \in B \tag{6f}$$

where X_{ij}^{r0} and V_{ij}^r are given by (3) and (4) respectively. Constraints (6b) are imposed to ensure that the heading angle variation for each aircraft satisfies some bounds, due to operational reasons. In our implementation, these bounds are set to $\theta_i^{min} = -\pi/6$ and $\theta_i^{max} = \pi/6$, for all $i \in A$. Constraints (6c) are the separation constraints for each pair of aircraft. Constraints (6d) define t_{ij}^m, (6e) are added to check the sign of t_{ij}^m (the separation condition (6c) is only imposed when $t_{ij}^m \geq 0$), and (6f) are integrality constraints on y_{ij}.

Note that trigonometric functions appear in the definition of V_{ij}^r. Further nonlinearities appear in constraints (6c), (6d) and (6e). Some of these nonlinearities can actually be reformulated using standard techniques [19]. Constraints (6e), which contain the product of a binary variable and a continuous variable, can be reformulated for all $(i,j) \in B$ to two linear inequalities, once bounds on variables t_{ij}^m are computed (starting from bounds deduced on V_{ij}^r). Then, a *big-M* reformulation can be applied on constraints (6c).

From some numerical experiments, however, it appears that these reformulations do not have a relevant impact on the computational efficiency of the resolution process. This is probably due to the fact that one of the main difficulties in the numerical solution of problem (6) is specially related to nonlinearities coming from trigonometric functions. The above reformulations will be therefore not considered in the following.

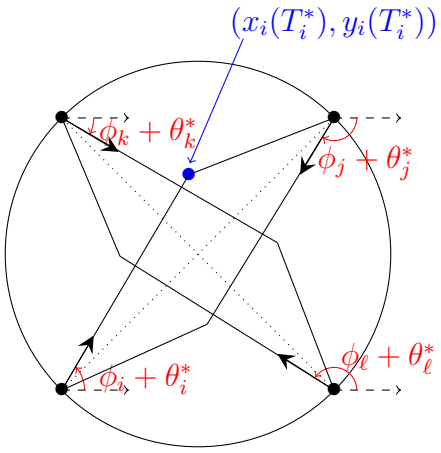

Figure 3: Forcing aircraft to return to their initial trajectories (dotted lines) after conflict resolution. Superscripts * indicate optimal values. At the point $(x_i(T_i^*), y_i(T_i^*))$, aircraft i can return to its initial trajectory being separated from all other aircraft.

After having solved problem (6), and so determined the optimal heading angle change θ_i^* for each aircraft $i \in A$ that ensures avoiding all potential conflicts, new changes in the heading angles have to be made in order to return aircraft to their initial trajectories (see Figure 3). The idea proposed by Alonso-Ayuso, Escudero and Martín-Campo [15, Section 4] can be easily applied in our context. It consists of determining the optimal time for which each aircraft can return to its trajectory in a final conflict-free configuration, by solving an unconstrained quadratic programming (QP) problem for each pair of aircraft $(i, j) \in B$. The objective function of this problem is the relative Euclidean distance between aircraft, that for each pair (i, j) is computed using the new aircraft spatial coordinates obtained by using heading angles from (6), and is minimized with respect to time. More precisely, knowing θ_i^* for each aircraft $i \in A$ from the solution of (6), the new coordinates of

aircraft i are given by

$$x_i(t) = x_i^0 + \cos(\phi_i + \theta_i^*)v_i t \quad \text{and} \quad y_i(t) = y_i^0 + \sin(\phi_i + \theta_i^*)v_i t. \tag{7}$$

Using these formulas, the quadratic program to be solved for each pair of aircraft $(i, j) \in B$ is

$$\min_{t_{ij}} \quad \left\| \begin{pmatrix} x_i(t_{ij}) - x_j(t_{ij}) \\ y_i(t_{ij}) - y_j(t_{ij}) \end{pmatrix} \right\|^2 \tag{8}$$

that allows us to compute for each $(i, j) \in B$ the minimum time such that i and j are separated using their new heading. Problem (8) is a convex continuous unconstrained QP problem, that can be easily solved by standard QP solvers. In practice, it is always solved, for each $(i, j) \in B$, after the solution of the HAC problem (6), *i.e.*, after computation of optimal θ_i^* for each aircraft $i \in A$. Once the optimal solution t_{ij}^* for problem (8) is found, we compute

$$T_i^* := \max_{\substack{j \neq i \\ j \in A}} \; t_{ij}^*$$

as the optimal time for which aircraft i can return to its initial trajectory in a conflict-free configuration. Using (7), this time corresponds to the point $(x_i(T_i^*), y_i(T_i^*))$ as shown in Figure 3. destination Knowing this point and the exit point from the air sector, it is easy to determine the new trajectory of each aircraft to come back to its initial trajectory. This is given by the straight line that connects both points. In [15], Alonso-Ayuso, Escudero and Martín-Campo gave an explicit formula of the corresponding new angle of motion.

4, standard terms of the the functions. good The solution, by global MINLP solvers, of the HAC (6) followed by the QP (8) may be quite time consuming (as discussed in Section 4). HAC (6) is a nonconvex problem which may have many local minima. Additionally, the presence of mixed 0-1 variables and of the nonlinearity of cosine and sine functions may impact the performance of a global solver. In order to speed up the solution process, the proposed mathematical programming model can be properly combined with another one. This is presented in the next subsection.

2.2 Maximizing the number of solved conflicts using VC

Recently, Cafieri [17] presented a model for the maximization of the number of aircraft conflicts that can be solved by only performing velocity regulation. We briefly recall hereafter this model. The headings of aircraft are kept fixed (so, no trigonometric functions appear) and the main decision variables are: $\bar{v}_i$ for all i, a continuous decision variable representing the aircraft speed, which is eventually modified with respect to the original one to solve conflicts, and z_{ij} for all $(i,j) \in B$, a binary decision variable defined as

$$z_{ij} = \begin{cases} 1 & \text{if aircraft } i \text{ and } j \text{ are separated (no conflict),} \\ 0 & \text{otherwise.} \end{cases}$$

Further variables are y_{ij} for all $(i,j) \in B$, used into another condition for aircraft separation, and a linearizing variable w_{ij} for all $(i,j) \in B$, both described below. The model reads as follows.

$$\max \sum_{(i,j)\in B} w_{ij} \tag{9a}$$

s.t.

$$\bar{v}_i^{min} \leq \bar{v}_i \leq \bar{v}_i^{max}, \quad \forall\, i \in A, \tag{9b}$$

$$\left((X_{ij}^{r_0} \cdot \bar{v}_{ij}^{r})^2 - \|\bar{v}_{ij}^{r}\|^2(\|X_{ij}^{r_0}\|^2 - d^2)\right)(2z_{ij} - 1) \leq 0, \quad \forall\, (i,j) \in B, \tag{9c}$$

$$(X_{ij}^{r_0} \cdot \bar{v}_{ij}^{r})(2y_{ij} - 1) \geq 0, \quad \forall\, (i,j) \in B, \tag{9d}$$

$$w_{ij} \geq y_{ij}, \quad \forall\, (i,j) \in B, \tag{9e}$$

$$w_{ij} \geq z_{ij}, \quad \forall\, (i,j) \in B, \tag{9f}$$

$$w_{ij} \leq y_{ij} + z_{ij}, \quad \forall\, (i,j) \in B, \tag{9g}$$

$$w_{ij} \in [0,1], \quad \forall\, (i,j) \in B, \tag{9h}$$

$$y_{ij}, z_{ij} \in \{0,1\}, \quad \forall\, (i,j) \in B. \tag{9i}$$

The bounds on $\bar{v}_i$, given by constraints (9b), are imposed to ensure that speed variations are small enough so that a subliminal control of velocities is performed, as suggested in the context of the ERASMUS project [20]. More precisely, the bounds $\bar{v}_i^{min}$ and $\bar{v}_i^{max}$ for each aircraft i are respectively given

by $0.94v_i$ and $1.03v_i$, to ensure that speeds vary between $-6\,\%$ and $+3\,\%$ of the original speed. Constraint (9c) expresses that i and j are separated when the corresponding z_{ij} is equal to 1. A further constraint, (9d), accounts for pairs of aircraft that are separated when, if separated at their initial positions, they have intersecting trajectories (here considered as straight lines) that are diverging ($X_{ij}^{r_0} \cdot \bar{v}_{ij}^{r} \geq 0$). To do so, additional binary variables $y_{ij}, \forall (i,j)$ are introduced, equal to 1 when the condition $X_{ij}^{r_0} \cdot \bar{v}_{ij}^{r} \geq 0$ is satisfied. Variables $w_{ij}, \forall (i,j)$ are then used to model the "or" condition relating the two separation constraints, (6c) (satisfied when $z_{ij} = 1$) and $X_{ij}^{r_0} \cdot \bar{v}_{ij}^{r} \geq 0$ (satisfied when $y_{ij} = 1$). The objective function is a linear function simply counting the pairs of separated aircraft.

Note that one of the main differences of this model with respect to the above presented HAC is that here the aircraft heading angles are kept fixed, while only the speeds are changed. Furthermore, here we do not aim at solving all conflicts optimizing a given criterion, but at maximizing the number of conflicts that the (subliminal) speed control can solve. In this sense, model (9) can be used as a first "filter" of air traffic in a given air sector, leaving eventually other conflicts to be solved by means of another separation maneuver, such as heading deviation. Such an approach can be easily seen in the context of an automation of aircraft speed variations expected in a next future, thus reducing the controllers' workload. These considerations lead us to propose the algorithm given in the next section.

3 Algorithm Max VC+HAC

Models (6) and (9) are MINLPs. In both cases, some nonlinearities come from the product of continuous and binary variables. In addition to that, the use of trigonometric functions in the HAC model adds strong nonlinearities that make the problem hard to solve. (6) resulting in On the other side, model (9) gives no guarantee to solve all conflicts, in particular head-to-head conflicts whose solutions need the use of maneuvers other than speed changes. Note also that the presence of constraints (9b) can be a source of infeasibility for some instances, meaning that it may happen that some conflicts cannot be solved by carrying out small adjustments in the speeds of aircraft. Remark that providing the HAC problem with input data such that a number of aircraft pairs are already separated (and hence separation constraints are straightforwardly satisfied) can speed up the solution process.

Based on these remarks and in order to take full advantage of the features of both models (6) and (9) and to provide a trade-off between efficiency and reliability, we propose a new algorithm for aircraft conflict avoidance, whose steps are summarized in Algorithm 1.

Algorithm 1 Aircraft deconfliction: Max VC + HAC

Step 1 Detect all head-to-head conflicts.

Step 2 Solve Max VC (9) without considering head-to-head conflicts.

Step 3 **If** all conflicts are solved **then**
Stop.
Else
Solve HAC (6) with aircraft having new velocities given by the solution of Max VC (9) and then go to Step 4.
End if

Step 4 Solve the QP (8) for each pair of aircraft $(i, j) \in B$.

Given an initial configuration of n aircraft sharing the same airspace during an observed time window, Algorithm 1 begins the resolution process by detecting all head-to-head conflicts. Then, the Max VC model (9), applied on the remaining pairs of aircraft, is solved at Step 2 to determine the possible changes to the speed of aircraft so as to separate as many aircraft pairs as possible. Our choice of not considering head-to-head conflicts when solving (9) is argued by the fact that this type of conflicts cannot be solved by only performing velocity regulation. Step 2 of Algorithm 1 will be henceforth referred as a *pre-processing* step. It acts as a first filter on the considered air traffic, before completing the deconfliction process by heading deviations. If all conflicts are solved after applying the pre-processing step, then the resolution process stops. Otherwise, model (6) is solved to determine a new aircraft configuration such that all conflicts are avoided. Note that for this model, the new speeds computed at Step 2 are used. The benefit of this choice is confirmed by the numerical results presented in Section 4. Once the optimal heading angle changes are found, we force the aircraft to return to their destination points by solving model (8) at Step 4.

Note that Algorithm 1 always outputs an optimal solution corresponding to an aircraft configuration where all conflicts are solved. When Algorithm 1

stops just after the solution of Max VC, because this is enough to solve all conflicts, the obtained solution is optimal with respect to criterion (9a), while when HAC is applied, optimal values are computed to deviate as less as possible from the original flight plans. However, note that from the operational viewpoint, even in the case of a solution obtained using Max VC only, where there is not a minimization of the deviation from original flight plans, such a deviation is never relevant thanks to the subliminal control framework.

4 Numerical experiments

Models (6), (8) and (9) are implemented using the AMPL modeling Language [21]. We made a comparative test of Algorithm 1 with/without Step 1 and Step 2 on two collections of test problems. The first collection is built by randomly placing on a circle of a given radius a number of aircraft that are initially all headed to the center. This leads to a highly symmetric configuration, where symmetry is also reinforced by considering the same speed for all aircraft. It is well known that this kind of test problem, while being unrealistic, is very hard to solve by exact global solvers. These problems, named `pb_n*` in Table 4, are known in the literature as *circle problems*. For the second collection, aircraft are arranged around a circle and have trajectories randomly chosen with a heading angle $\in [-\pi/6, \pi/6]$ with respect to the diameter of the circle. The end point of each trajectory belongs to the circle as well. Note that these problems are more realistic than circle problems without deviation. In Table 5, they are referred to as `rpb_n*`. For both collections, the aircraft move at the same speed, namely 400 NM/h. After executing Step 2 of Algorithm 1, new values may be assigned to speeds of all (or some) aircraft. Given constraints (9b), these new values range from 376 NM/h to 412 NM/h. Note that this kind of instances have been also considered by [8, 15, 16, 18]. The standard separation distance, d, to be respected between aircraft trajectories is equal to 5 NM. To get a perspective on the size of problems being solved, Table 1 summarizes the sizes of the MINLP problem (HAC model) varying the number of aircraft. Recall that the objective function and the equality and inequality constraints (except the bound constraints) are nonlinear.

In this paper, we are interested in the global solution of the problem at hand by deterministic MINLP solvers (this kind of solvers need an algebraic representation of the objective and constraint functions for the computa-

Table 1: Dimensions of Problem (4) depending on the number of aircraft.

n	# variables		# auxiliary variables		# constraints	
	continuous	integer	continuous	integer	equality	inequality
2	5	1	21	1	3	6
3	12	3	49	3	9	12
4	22	6	89	6	18	20
5	35	10	141	10	30	30
6	51	15	205	15	45	42
7	70	21	281	21	63	56
8	92	28	369	28	84	72

tion of convex envelopes and under-estimators [22]). MINLP solvers falling in the category of interest are the following: Antigone [23], Baron [24], Couenne [25], LindoApi [26] and Scip [27]. Our choice of the solver for the numerical experiments has been established on the basis of the capability of the solvers to handle trigonometric functions and of their availability as free solvers. Table 2 summarizes the characteristics of the above MINLP solvers. To the best of our knowledge, trigonometric functions are handled by two solvers only, Couenne and LindoApi, and only Couenne is a free open-source solver. This motivated our selection of Couenne as MINLP solver for our numerical experiments. For the resolution of the convex QP (8), we used the open source software package for large-scale non-linear optimization Ipopt [28], that implements an Interior-Point method. Both solvers were run with their default settings.

Table 2: An overview on deterministic global solvers for nonconvex MINLP.

Solver	Free	AMPL	Deterministic	Trigonometric functions
Antigone	–	–	×	–
Baron	–	–	×	–
Couenne	×	×	×	×
LindoApi	–	–	×	×
Scip	×	×	×	–

Tests were performed on a 2.66 GHz Intel Xeon (octo core) processor with 32GB of RAM and Linux Operating System. Tables 4 and 5 show the comparison of the results obtained using HAC with Max VC as a pre-processing step (Algorithm 1) and using HAC only on the original problem. The headings are as follows: n, number of aircraft; n_c, number of all potential

conflicts; n_{hthc}, number of head-to-head conflicts; *time*, computing time in seconds, and *obj*, optimal value of the objective function of model (6).

As previously mentioned, the goal of the second step of Algorithm 1 is not to solve all conflicts, but to maximize the number of conflicts that can be solved using the subliminal speed control maneuver. We are specially interested in a first step of the algorithm that could be fast, while solving as many conflicts as possible. In order to avoid a time-consuming resolution of Max VC (9), a CPU time limit of 30 seconds was imposed. As it can be seen from Tables 4 and 5, the use of Max VC (9) as a pre-processing step for HAC (6) significantly improves the solution in terms of CPU time and quality of solution. The most significant time improvement occurs for problem **rpb_n8_3**, for which the computing time is reduced from 11967.96 seconds to less than 10 seconds, exhibiting a speed-up by a factor of 1316. Note that for this problem, 9 out of 10 conflicts were solved using the pre-processing step.

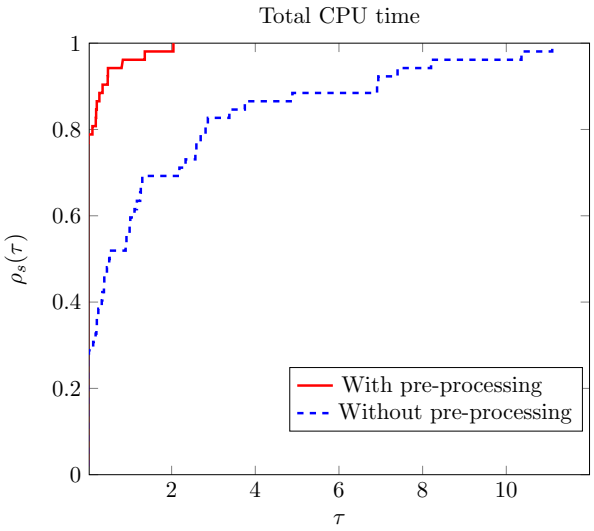

Figure 4: Performance profile comparing the total CPU time for Algorithm 1 with and without pre-processing on a set of 52 randomly generated problems.

To compare the numerical results presented in Tables 4 and 5 more clearly, we make use of the performance profiles of Dolan and Moré [29]. Figure 4 summarizes, using a performance profile, these results. For $\tau \geq 0$, $\rho_s(\tau)$ is the fraction of problems for which the performance of a given algorithm is within

Table 3: Solutions of problem **pb_n5_1** using HAC with and without pre-processing

Optimal HACs	With pre-processing	Without pre-processing
θ_1^*	-0.033039	0.043072
θ_2^*	0.073481	0.074654
θ_3^*	0.030125	0.039003
θ_4^*	-0.020949	0.040227
θ_5^*	0.020258	0.032408

a factor 2^τ of the best one. Figure 4 clearly highlights the benefit of using the proposed pre-processing step before solving the HAC model. In fact, the efficiency of this version of Algorithm 1, which is readable on the left vertical axis of the performance profile, is 80% whereas that of Algorithm 1 without applying a pre-processing step does not exceed 30%. This efficiency gain can be explained by the fact that the pre-processing step provides model (6) with input data where in most cases the majority of conflicts are solved and there only remain some difficult conflicts (e.g. head-to-head conflicts) for which heading angle deviations have to be performed. This greatly simplify the branch-and-bound procedure for the solution of (6). In term of robustness, both versions of Algorithm 1 have the same rate of robustness since they solve all considered instances. However, better solutions were obtained when using a pre-processing for all tested problems, as presented in Tables 4 and 5. One reason behind such improvement is that the use of a pre-processing step breaks up the symmetry introduced in the initial problem by changing the velocity of some (or all) aircraft, thus making the feasible region of problem (6) larger than that without pre-processing. In fact, one observes that the solution of problem (6) without pre-processing consists of a symmetric configuration with all (or some) aircraft deviated either to the left (positive turn) or to the right (negative change). On the contrary, the optimal heading angle changes obtained by solving the HAC model after applying a pre-processing step can be a mixture of positive and negative values. The example in Table 3 illustrates this situation.

Increasing the number of aircraft, the number of constraints and variables (continuous and binary) increases. As expected in a Branch and Bound method, like the one implemented in the exact global solver COUENNE, this leads to a significant increase in the computing time required to explore all

the branches of the search tree to ensure that the obtained solution is a global optimum. The same conclusion was reached in [16, 30]. For problems `pb_n*`, that are highly symmetric, instances up to $n = 5$ are efficiently solved, while from $n \geq 6$ they are highly computational demanding. Problems `rpb_n*` appear to be easier to solve than the circle problems, and we are able to efficiently solve instances up to $n = 8$.

Remark 1. *In some cases, the optimal solution returned by* Couenne *is slightly infeasible. This can be explained by the difficulty of the treatment of trigonometric functions (cosine and sine used in our HAC model) by MINLP solvers. Note that such behavior for* Couenne *has been already observed on some instances of the MINLPLIB collection [31]. To overcome this problem, we run* Ipopt *for the HAC model with integer variables fixed and the optimal solution returned by* Couenne *as a starting point. In all cases, it appears that a small modification of the variables is enough to obtain a "more feasible" solution. For example, by running Algorithm 1 on the problem* ***pb_n5_1*** *we obtain the optimal objective value* $f^* = 0.008227$*. When testing the feasibility of the solution, it appears that one conflict remains not solved. When applying the procedure described above,* Ipopt *returns the following slightly modified solution* $f^* = 0.00824\mathbf{8}$*, for which all conflicts are solved considering the associated heading angle changes. Note finally that the time needed to accomplish this task is negligible comparing to that needed for solving problem* (6)*. In all our tests, it does not exceed 0.02 seconds.*

5 Conclusions

We proposed a MINLP formulation (HAC) of the aircraft conflict avoidance problem, where potential conflicts are solved by aircraft heading angle deviations. This formulation has a quadratic objective and nonlinear nonconvex constraints. To speed-up the solution process, we also proposed the combination of the above MINLP with another one, MaxVC, where the number of conflicts that can be solved by regulating the aircraft speeds is maximized. The idea is to provide the first MINLP with a pre-processing step, that from the operational viewpoint can be interpreted as a preliminary filter of the considered air traffic. Numerical results are encouraging and show the interest of the proposed two-step approach. As expected, the use of exact global solvers allows us to solve medium-scale problems. Alternative solution approaches will be investigated in future work to raise the size of solved problems, as

well as further mathematical programming formulations and their possible reformulations.

Acknowledgment

The authors would like to thank the editor and the two anonymous referees, whose comments helped to improve the presentation of the paper. Financial support by French National Research Agency (ANR) through grant ANR 12-JS02-009-01 "ATOMIC" is gratefully acknowledged.

References

[1] EUROCONTROL, Eurocontrol long-term forecast: IFR Flight Movements 2010–2030, Tech. rep., Eurocontrol–Air Traffic Statistics and Forecast (2010).

[2] SESAR consortium: The european ATM master plan, Tech. rep., European Commission and EUROCONTROL (2009).

[3] J. Hansman, Impact of NextGen integration on improving efficiency and safety of operations, in: In Proceedings of TRB: the 91st Annual Meeting of the Transportation Research Board, Washington D.C, USA, 2012.

[4] J. Kuchar, L. Yang, A review of conflict detection and resolution modeling methods, IEEE Transactions on Intelligent Transportation Systems 1 (4) (2000) 179–189.

[5] M. Brochard, Erasmus - en route air traffic soft management ultimate system, Tech. rep., Eurocontrol Experimental Centre (2006).

[6] S. Cafieri, MINLP in Air Traffic Management: Aircraft conflict avoidance, in: T. Terlaky, M. Anjos, S. Ahmed (Eds.), Advances and Trends in Optimization with Engineering Applications, MOS-SIAM Series on Optimization, SIAM, Philadelphia, in press.

[7] L. Pallottino, E. M. Feron, A. Bicchi, Conflict resolution problems for air traffic management systems solved with mixed integer programming,

IEEE Transactions on Intelligent Transportation Systems 3 (1) (2002) 3–11.

[8] M. Christodoulou, C. Costoulakis, Nonlinear mixed integer programming for aircraft collision avoidance in free flight, in: Electrotechnical Conference, 2004. MELECON 2004. Proceedings of the 12th IEEE Mediterranean, Vol. 1, 2004, pp. 327–330.

[9] A. Vela, S. Solak, W. Singhose, J.-P. Clarke, A mixed integer program for flight-level assignment and speed control for conflict resolution, in: Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on, 2009, pp. 5219–5226.

[10] A. Vela, S. Solak, J. Clarke, W. Singhose, E. Barnes, E. Johnson, Near real-time fuel-optimal en route conflict resolution, IEEE Transactions on Intelligent Transportation Systems 11 (4) (2010) 826–837.

[11] D. Rey, C. Rapine, R. Fondacci, N.-E. E. Faouzi, Minimization of potential air conflicts through speed regulation, Transportation Research Record: Journal of the Transportation Research Board 2300 (2012) 59–67.

[12] D. Rey, C. Rapine, R. Fondacci, N.-E. E. Faouzi, Subliminal speed control in air traffic management: Optimization and simulation, Transportation Science 50 (1) (2016) 240–262.

[13] A. Alonso-Ayuso, L. F. Escudero, F. J. Martín-Campo, Collision avoidance in air traffic management: A mixed-integer linear optimization approach, IEEE Transactions on Intelligent Transportation Systems 12 (1) (2011) 47–57.

[14] A. Alonso-Ayuso, L. F. Escudero, F. J. Martín-Campo, 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) (2012) 3136–3146.

[15] A. Alonso-Ayuso, L. F. Escudero, F. J. Martn-Campo, Exact and approximate solving of the aircraft collision resolution problem via turn changes, Transportation Science 50 (1) (2016) 263–274.

[16] S. Cafieri, N. Durand, Aircraft deconfliction with speed regulation: new models from mixed-integer optimization, Journal of Global Optimization 58 (4) (2014) 613–629.

[17] S. Cafieri, Maximizing the number of solved aircraft conflicts through velocity regulation, in: MAGO 2014, 12th Global Optimization Workshop, Málaga, Spain, 2014, pp. 1–4.

[18] C. Peyronne, A. R. Conn, M. Mongeau, D. Delahaye, Solving air traffic conflict problems via local continuous optimization, European Journal of Operational Research 241 (2) (2015) 502–512.

[19] L. Liberti, S. Cafieri, F. Tarissan, Reformulations in mathematical programming: A computational approach, in: A. Abraham, A.-E. Hassanien, P. Siarry, A. Engelbrecht (Eds.), Foundations of Computational Intelligence Volume 3, Vol. 203 of Studies in Computational Intelligence, Springer Berlin Heidelberg, 2009, pp. 153–234.

[20] C. Dupré, D. Bonini, G. Granger, How erasmus can support an increase in capacity in 2020, in: Proceedings of the 7th International Conference on Computing, Communications and Control Technologies: CCCT, 2009.

[21] R. Fourer, D. M. Gay, B. W. Kernighan, AMPL: A Modeling Language for Mathematical Programming, 2nd Edition, Brooks/Cole, 2002.

[22] M. R. Bussieck, S. Vigerske, Minlp solver software, in: J. Cochran, L. Cox, P. Keskinocak, J. Kharoufeh, J. Smith (Eds.), Wiley Encyclopedia of Operations Research and Management Science, John Wiley & Sons, Inc., 2011.

[23] R. Misener, C. A. Floudas, Antigone: Algorithms for continuous / integer global optimization of nonlinear equations, Journal of Global Optimization 59 (2) (2014) 503–526.

[24] N. V. Sahinidis, Baron: A general purpose global optimization software package, Journal of Global Optimization 8 (2) (1996) 201–205.

[25] P. Belotti, J. Lee, L. Liberti, F. Margot, A. Wachter, Branching and bounds tightening techniques for non-convex MINLP, Optimization Methods Software 24 (4-5) (2009) 597–634.

[26] Y. Lin, L. Schrage, The global solver in the LINDO API, Optimization Methods and Software 24 (4-5) (2009) 657–668.

[27] T. Achterberg, SCIP: solving constraint integer programs, Mathematical Programming Computation 1 (1) (2009) 1–41.

[28] A. Wächter, L. T. Biegler, On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming, Mathematical Programming 106 (1) (2006) 25–57.

[29] E. D. Dolan, J. J. Mor, Benchmarking optimization software with performance profiles, Mathematical Programming 91 (2) (2002) 201–213.

[30] A. Alonso-Ayuso, L. F. Escudero, F. J. Martín-Campo, An exact multi-objective mixed integer nonlinear optimization approach for aircraft conflict resolution, TOP 24 (2) (2016) 381–408.

[31] M. R. Bussieck, A. S. Drud, A. Meeraus, MINLPLib - a collection of test models for mixed-integer nonlinear programming., INFORMS Journal on Computing 15 (1) (2003) 114–119.

Table 4: Conflicts resolution for circle problem without deviation using HAC with and without pre-processing

				Max VC Pre-processing		HAC (with pre-processing)			HAC (without pre-processing)		
Name	n	n_c	n_{hth}	n_{rc}	time (s)	n_{rc}	time (s)	obj	n_{rc}	time (s)	obj
pb_n2	2	1	1	1	0.00	0	0.14	0.001250	0	0.13	0.001250
pb_n3_1	3	3	0	3	1.99	0	0.82	0.002327	0	0.90	0.002501
pb_n3_2	3	3	0	1	1.22	0	1.29	0.001667	0	1.97	0.006665
pb_n3_3	3	3	1	1	0.05	0	0.43	0.000314	0	1.08	0.000950
pb_n4_1	4	6	0	3	6.18	0	5.07	0.003172	0	9.90	0.007240
pb_n4_2	4	6	0	2	5.60	0	7.26	0.007399	0	31.64	0.017065
pb_n4_3	4	6	2	2	0.09	0	0.21	0.000630	0	0.56	0.001318
pb_n5_1	5	10	1	5	16.14	0	24.49	0.008248	0	58.21	0.011629
pb_n5_2	5	10	1	5	23.85	0	30.69	0.017774	0	68.60	0.018468
pb_n5_3	5	10	1	6	19.24	0	25.79	0.006740	0	89.79	0.017100
pb_n5_4	5	10	2	5	25.81	0	32.71	0.012480	0	80.17	0.014750
pb_n5_5	5	10	0	5	21.65	0	26.73	0.005652	0	63.05	0.012149
pb_n5_6	5	10	0	6	31.21	0	16.45	0.004208	0	47.18	0.011225
pb_n5_7	5	10	1	5	23.62	0	33.46	0.006951	0	66.13	0.012262
pb_n5_8	5	10	2	5	19.92	0	66.16	0.019788	0	122.44	0.017556

Name	n	n_c	n_{hth}	Max VC Pre-processing		HAC (with pre-processing)			HAC (without pre-processing)		
				n_{rc}	time (s)	n_{rc}	time (s)	obj	n_{rc}	time (s)	obj
pb_n5_9	5	10	0	6	30.98	0	100.68	0.008972	0	142.52	0.019119
pb_n5_10	5	10	1	6	26.38	0	47.87	0.008536	0	160.61	0.025960
pb_n5_11	5	10	0	6	30.00	0	25.09	0.006958	0	69.90	0.011190
pb_n5_12	5	10	0	5	30.00	0	29.18	0.009471	0	68.06	0.012111
pb_n5_13	5	10	0	4	30.00	0	134.26	0.026393	0	331.32	0.023265
pb_n5_14	5	10	1	4	24.58	0	46.44	0.013300	0	51.21	0.013790
pb_n5_15	5	10	0	6	30.00	0	86.28	0.006918	0	220.09	0.014551
pb_n5_16	5	10	1	6	30.00	0	35.04	0.007874	0	56.80	0.010367
pb_n5_17	5	10	0	6	30.00	0	59.73	0.008833	0	77.51	0.011940
pb_n5_18	5	10	1	6	30.00	0	62.28	0.010092	0	41.34	0.011153
pb_n5_19	5	10	0	6	30.00	0	25.03	0.006623	0	39.53	0.009920
pb_n5_20	5	10	1	5	30.00	0	117.74	0.020694	0	167.77	0.019739
pb_n5_21	5	10	0	6	30.00	0	32.90	0.007354	0	62.62	0.009051
pb_n5_22	5	10	2	6	8.89	0	45.49	0.007651	0	352.03	0.030577
pb_n5_23	5	10	2	2	0.08	0	7.59	0.000278	0	55.95	0.001543
pb_n6_1	6	15	3	3	1.83	0	2819.51	0.000408	0	6773.31	0.001649
pb_n6_2	6	15	3	3	0.54	0	1915.25	0.000417	0	8722.30	0.001661

Table 5: Conflicts resolution for circle problem with deviation using HAC with and without pre-processing

Name	n	n_c	n_{hth}	Max VC Pre-processing		HAC (with pre-processing)			HAC (without pre-processing)		
				n_{rc}	time (s)	n_{rc}	time (s)	obj	n_{rc}	time (s)	obj
rpb_n2_1	2	1	1	1	0.00	0	0.04	0.000141	0	0.04	0.000141
rpb_n2_2	2	1	1	1	0.00	0	0.05	0.000795	0	0.05	0.000795
rpb_n3_1	3	2	0	0	0.04	0	0.00	0.000000	0	0.19	0.000078
rpb_n3_2	3	2	0	0	0.04	0	0.00	0.000000	0	0.83	0.000513
rpb_n3_3	3	1	0	0	0.29	0	0.00	0.000000	0	0.38	0.000113
rpb_n4_1	4	1	0	0	0.03	0	0.00	0.000000	0	2.07	0.000156
rpb_n4_2	4	3	1	1	0.04	0	0.66	0.000218	0	4.22	0.001175
rpb_n4_3	4	2	0	1	3.09	0	0.65	0.000104	0	2.41	0.000202
rpb_n5_1	5	6	0	0	0.04	0	0.00	0.000000	0	16.90	0.000408
rpb_n5_2	5	3	0	0	0.05	0	0.00	0.000000	0	32.39	0.000450
rpb_n5_3	5	8	0	0	5.22	0	0.00	0.000000	0	26.61	0.000613
rpb_n6_1	6	5	1	1	0.06	0	2.77	0.000052	0	340.59	0.000955
rpb_n6_2	6	9	2	2	0.05	0	23.04	0.000083	0	310.64	0.000855
rpb_n6_3	6	4	0	1	30.96	0	16.52	0.000067	0	333.30	0.000693
rpb_n7_1	7	2	0	1	30.53	0	9.58	0.000011	0	52.58	0.000210
rpb_n7_2	7	7	0	2	30.53	0	82.78	0.000173	0	13919.22	0.001162

Name	n	n_c	n_{hth}	Max VC Pre-processing		HAC (with pre-processing)			HAC (without pre-processing)		
				n_{rc}	time (s)	n_{rc}	time (s)	obj	n_{rc}	time (s)	obj
rpb_n7_3	7	9	0	4	30.50	0	122.88	0.000094	0	126.95	0.002637
rpb_n8_1	8	12	2	2	12.19	0	2561.98	0.000095	0	26795.60	0.001189
rpb_n8_2	8	5	0	0	0.09	0	0.00	0.000000	0	5370.44	0.000373
rpb_n8_3	8	10	1	1	0.08	0	9.01	0.000026	0	11967.96	0.001019