A Variable Neighborhood Search approach for the aircraft conflict resolution problem

Antonio Alonso-Ayuso
Escuela Técnica Superior de Ingeniería Informática
Universidad Rey Juan Carlos
Móstoles, Madrid (Spain)
antonio.alonso@urjc.es

F. Javier Martín-Campo
Departamento de Estadística e Investigación Operativa
Universidad Complutense de Madrid
Madrid (Spain)
javier.martin.campo@mat.ucm.es

Laureano F. Escudero
Escuela Técnica Superior de Ingeniería Informática
Universidad Rey Juan Carlos
Móstoles, Madrid (Spain)
laureano.escudero@urjc.es

Nenad Mladenović
Laboratoire Automatique, Mécanique, Informatique Humaines
Université de Valenciennes et du Hainaut-Cambrésis
Valenciennes, France
Nenad.Mladenovic@univ-valenciennes.fr

***Abstract*—A metaheuristic approach based on Variable Neighborhood Search to tackle the enroute aircraft conflict resolution problem is presented. The three maneuvers (velocity, heading angle and altitude level variations) are allowed to be performed. Based on geometric rules, we can determine if a conflict situation occurs between each pair of aircraft under consideration. As three different maneuvers could be performed, they are dealt with in a multiobjective framework, allowing Air Traffic Control officers to make the final decision based on economic and comfort terms.**

Air Traffic Management; aircraft conflict avoidance; Variable Neighborhood Search; multiobjective optimization

I. INTRODUCTION

The number of passengers in air transportation has grown during the last decades increasing the number of aircraft flying the sky. Due to the complexity of managing such traffic, automatic tools are required to deal with an important problem that is the aircraft Conflict Resolution Problem (CRP) and aims to provide safety to the airspace. The goal of this problem is to provide a new aircraft configuration in which any aircraft does not violate the safety distances that must keep on flight. Those distances are 5 nautical miles (nm) horizontally and 2000 feet (ft) vertically defining a cylinder with 2.5 nm. of radius and 2000 ft. of height as safety region around each aircraft. In order to avoid any conflict situation, defined as the event in which two or more aircraft experience a loss of the safety distances, every aircraft is allowed to perform three types of maneuvers, namely velocity (VC) and turn heading angle (TC) maneuvers (as horizontal) and altitude level (AC) maneuvers (as vertical). The ERASMUS project [9] suggests velocity and heading angle regulations in small ranges from -6 to 3% of the current velocity and -30º to 30º for heading angle changes. However, for altitude level changes no guidelines are given, but due to comfort, it is desirable to be as less as possible.

The literature has tackled this problem by different approaches (optimal control, mathematical programming and metaheuristic approaches, among others). We refer to [19] for a survey up to year 2000 and to [20] for a survey up to year 2010. However, the literature taking the three maneuvers at once is limited and very often presents approaches that are too complex to provide an efficient answer in almost real-time, which is required by the Air Traffic Control officers (ATCos).

It is worth pointing out that recently, different groups, whose field of expertise is on mathematical optimization, are working on studies from that field in the CRP. To the best of our knowledge, the first works based on this discipline are [13], [16], [25] and [30] where different models are studied but allowing one or two maneuvers, at most. The work presented in [30] is not based on maneuvers but on waypoints. Nowadays, the guidelines for future Air Traffic Management are clearer and other works apply that discipline with the new specifications in advanced models as the ones presented in [10], [23], [24], [26], [27], [28], [29] and [32], among others.

The Variable Neighborhood Search (VNS) metaheuristic approach was firstly introduced in [21]. It has been applied to many problems, especially in the field of combinatorial optimization; see [17] and [22]. Its basic idea lies on changing neighborhood structures in search for a better solution, in both the perturbation (diversification) and the local search (intensification) phases. This combination allows exploring the solution space. One of the main advantages of this metaheuristic approach is that it needs only one parameter to tune the algorithm, making it more flexible to be applied to different problems.

In our work, that parameter depends on the number of aircraft under consideration, which makes our algorithm totally adaptable to be used by any ATCo. So, we present a VNS approach to deal with the CRP by performing any of the three maneuvers. VNS for CRP was introduced for the first time in [3], [5] by considering only heading angle changes. We extend those works by including the velocity and altitude changes as well as a multiobjective methodology to deal with their different nature.

The rest of the paper is organized as follows: Section II describes the geometric rules used to detect whether there is a conflict situation or not between a pair of aircraft; Section III presents the VNS methodology applied to solve the CRP; Section IV describes the three methods in multiobjective optimization that are used in our research; Section V reports a preliminary computational experiment; Finally, Section VI concludes and details the future research lines.

II. Conflict Avoidance

The geometric construction for detecting conflict situations to be considered in the study is shown in Figs. 1 and 2. Let $\vec{v}_i$ and $\vec{v}_j$ denote the velocity vectors of the pair of aircraft, say *i* and *j*, respectively. The main idea lies on constructing the relative velocity vector $\vec{v}_i - \vec{v}_j$. Observe in the figure two tangents to the safety circle of aircraft *j* that are parallel to the relative velocity of the pair of aircraft *i* and *j*. They define a region where the intersection with the trajectory of aircraft *i* is a segment named the shadow segment. Notice that if the intersection of that segment with the safety circle of aircraft *i* is empty, then, there is no conflict situation between the two aircraft. Otherwise, the conflict must be solved. Assume that ω_{ij} and α_{ij} denote the $\arctan\left(\frac{y_i - y_j}{x_i - x_j}\right)$ and $\arcsin\left(\frac{r_i + r_j}{d_{ij}}\right)$, respectively, where x_i and y_i are the abscissa and ordinate of the current position of aircraft *i*, respectively; r_i is the safety radius of aircraft *i*; and d_{ij} is the Euclidean distance between aircraft *i* and *j*. A conflict situation can be detected depending upon the tangent of angles $l_{ij} = \omega_{ij} + \alpha_{ij}$ and $g_{ij} = \omega_{ij} - \alpha_{ij}$ and the tangent of the relative velocity vector.

Notice that if different safety radii are considered for an aircraft (e.g., due to their expansion in case of bad weather conditions occur, due to aircraft performance, etc.), the following expression from [1], [2] and [4] should be used to compute the angle α_{ij},

$$\alpha_{ij} = \arctan \frac{r_i + r_j}{\sqrt{d_{ij}^2 - (r_i + r_j)^2}}.$$

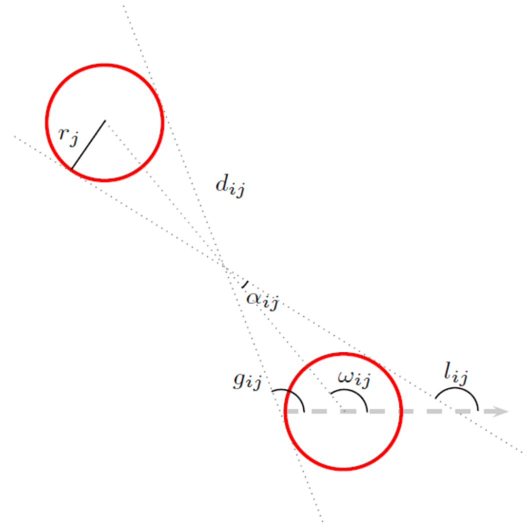

Fig. 1. Angles for conflict detection

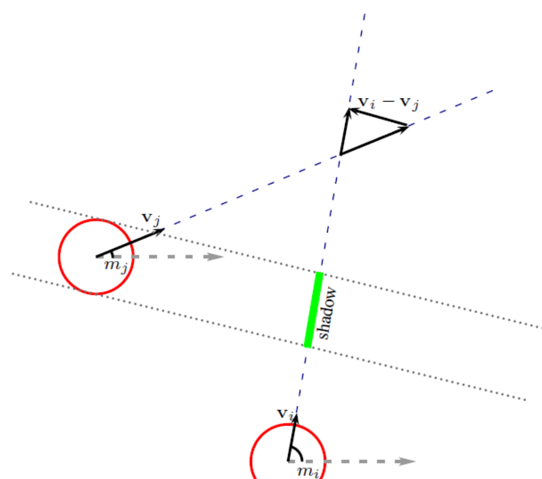

Fig. 2. Geometric construction for conflict detection

In the geometric construction the notation is based on vectors, but they can be decomposed into the two components, abscissa and ordinate, in the mathematical model. Therefore, no conflict occurs between aircraft *i* and *j* if any of the following expressions is satisfied,

$$\begin{aligned} &\frac{(v_i+\upsilon_i)\sin(m_i+\mu_i)-(v_j+\upsilon_j)\sin(m_j+\mu_j)}{(v_i+\upsilon_i)\cos(m_i+\mu_i)-(v_j+\upsilon_j)\cos(m_j+\mu_j)} \geq \tan(l_{ij}) \\ &\frac{(v_i+\upsilon_i)\sin(m_i+\mu_i)-(v_j+\upsilon_j)\sin(m_j+\mu_j)}{(v_i+\upsilon_i)\cos(m_i+\mu_i)-(v_j+\upsilon_j)\cos(m_j+\mu_j)} \leq \tan(\mathrm{g}_{ij}) \end{aligned} \qquad (1)$$

where m_i is the current direction of motion and μ_i and υ_i are the heading angle and velocity variations to be obtained, respectively, such that now the new angles of motion $m_i + \mu_i$ and $m_j + \mu_j$ and the new velocities $v_i + \upsilon_i$ and $v_j + \upsilon_j$ avoid the conflict between aircraft *i* and *j*. Two situations can be seen to arise in the expressions above.

Notice that the left-hand-side in the previous expression can involve a null denominator. We refer to this as a pathological situation which produces unstable solutions since a specific conflict between two aircraft may be wrongly solved, even leading the aircraft to crash in the worst scenario. These situations can be detected before the optimization phase. In order to solve these cases, parameters m_i and m_j can be replaced with $m_i + \pi/2$ and $m_j + \pi/2$, i.e., the elements of the configurations are turned to $\pi/2$ rad. So there is no longer a pathological situation. Note: This replacement is only made for the aircraft *i* and *j* under pathological situation.

The approach presented in this work assumes that the conflict situations are solved by performing instantaneously the corresponding maneuvers (VC, TC and AC). Notice that in real situations the problem should be solved each time instant at which an aircraft comes into an air sector. This is made to ensure that no conflict situations occur anymore.

If a TC is performed by a given aircraft, it should obviously be forced to return to its initial trajectory once the conflict situation has been resolved. This could be done by solving a set of unconstrained quadratic models, see [8]. The optimization of those models does not present any difficulty if an appropriate optimization solver is used. In our case the model takes less than a second to be resolved on a typical PC hardware platform.

III. VARIABLE NEIGHBORHOOD SEARCH

As has been stated in the introduction, VNS has been applied to many problems in combinatorial optimization. However, when additionally continuous variables are involved, it is more complicated as it happens in our approach (velocity and heading angle changes are related to continuous variables). One way commonly used in continuous optimization when using VNS is moving to another solution by adding a value at random forming the new neighborhood structure. This kind of movements are used in both, the local search and the shaking phases, see below.

A. Penalty function

In the previous section has been described the equations to detect if there is a conflict situation between two aircraft, see equations (1). We formulate the problem highly penalizing the infeasibility, so, expressions (1) can be expressed as the only one following inequality,

$$\tan(g_{ij}) \le \frac{(v_i + \upsilon_i)\sin(m_i + \mu_i) - (v_j + \upsilon_j)\sin(m_j + \mu_j)}{(v_i + \upsilon_i)\cos(m_i + \mu_i) - (v_j + \upsilon_j)\cos(m_j + \mu_j)} \le \tan(l_{ij})$$

If the previous inequality is satisfied there is a conflict situation between aircraft *i* and *j*. However, inequalities in (1) have difficulties in the computation when the denominator is null. We have referred above as pathological cases. In that case, the following expression is taken into account,

$$-\cot(g_{ij}) \le \frac{(v_i + \upsilon_i)\sin(m_i + \mu_i + \pi/2) - (v_j + \upsilon_j)\sin(m_j + \mu_j + \pi/2)}{(v_i + \upsilon_i)\cos(m_i + \mu_i + \pi/2) - (v_j + \upsilon_j)\cos(m_j + \mu_j + \pi/2)} \le -\cot(l_{ij})$$

where it can be observed that the angles involved in the geometric construction presented above are turned $\pi/2$ rad.

The penalization function must consider velocity, heading angle and altitude level changes and it is considered as follows,

$$g(\upsilon,\mu,\gamma) = \begin{cases} \sum\limits_{i<j\in\Im} \max\left\{0, \min\left\{\tan(l_{ij}) - t_{ij}, t_{ij} - \tan(g_{ij})\right\}\right\} \\ \qquad\qquad \text{if } cp_{ij} = 0 \text{ and } z_i + \gamma_i = z_j + \gamma_j \\ \sum\limits_{i<j\in\Im} \max\left\{0, \min\left\{\tan(l_{ij}) - t_{ij}, t_{ij} - \tan(g_{ij})\right\}\right\} \\ \qquad\qquad \text{if } cp_{ij} = 1 \text{ and } z_i + \gamma_i = z_j + \gamma_j \end{cases}$$

being cp_{ij} a 0-1 parameter which equals 1 if there is a pathological case between aircraft *i* and *j*; z_i the initial altitude level of aircraft *i*; γ_i the number of altitude levels that aircraft *i* changes (positive and negative); and,

$$t_{ij} = \frac{(v_i + \upsilon_i)\sin(m_i + \mu_i) - (v_j + \upsilon_j)\sin(m_j + \mu_j)}{(v_i + \upsilon_i)\cos(m_i + \mu_i) - (v_j + \upsilon_j)\cos(m_j + \mu_j)}$$

$$t'_{ij} = \frac{(v_i + \upsilon_i)\sin(m_i + \mu_i + \pi/2) - (v_j + \upsilon_j)\sin(m_j + \mu_j + \pi/2)}{(v_i + \upsilon_i)\cos(m_i + \mu_i + \pi/2) - (v_j + \upsilon_j)\cos(m_j + \mu_j + \pi/2)}$$

The function $g(\upsilon,\mu,\gamma)$ considers that if you aircraft *i* and *j* are flying at the same altitude level, the geometric construction must be applied to detect if there is a conflict situation or not. Otherwise, if the two aircraft fly at different altitude levels, there is not a conflict between them. Depending on whether a pathological case happens or not, the corresponding expression is applied.

Besides function $g(\upsilon,\mu,\gamma)$, the objective functions for each maneuver presented in IV.A must be aggregated.

B. Data structure

In our VNS algorithm, we have five vectors used to save the aircraft configuration: velocity (*V*), heading angle (*T*), altitude level (*Z*), abscissa (*X*) and ordinate (*Y*); three solution vectors (υ), (μ) and (γ) for velocity, heading angle and

Algorithm 1: Updating matrix A

```
Function Updating(j, A, ν, μ, γ, V, T, Z, CP, TL, TG,
CTL, CTG);
i = 0;
repeat
    if CP(i, j) = 0 and Z_i + γ_i = Z_j + γ_j then
        Use T, TL and TG to calculate A_ij
        by using g;
    else
        if CP(i, j) = 1 and Z_i + γ_i = Z_j + γ_j then
            Use T + π/2, CTL and CTG to
            calculate A_ij by using g;
        else
            A_ij = 0;
        end
    end
    i = i + 1;
until i = n;
```

altitude level changes, respectively; one 0-1 matrix to denote parameter cp_{ij}; five matrices *TL*, *TG*, *CTL*, *CTG*, *A*, to save the values for $\tan(l_{ij})$, $\tan(g_{ij})$, $-\cot(l_{ij})$, $-\cot(g_{ij})$ and the values of function $g(\upsilon,\mu,\gamma)$. The latter is a dynamic matrix which is changing during the VNS algorithm. The rest of parameters corresponds to the initial configuration.

C. Local Search

The Local Search phase is the intensification phase. Its aim is to improve a given solution. In our case the evaluation of new solutions is made by changing by a fixed quantity the velocity or the heading angle as well as changing one altitude level. The Local Search phase in our CRP is based on first improvement strategy, which consists on evaluate one solution and if it is improved then take the new one until no improvement is found instead of the mechanism for best improvement which consists of evaluate a set of values and take the best one. We have tried both strategies but we have chosen first improvement since no significant improvement in both time and solution's quality is found.

Algorithm 1 presents the general process for updating matrix *A* according to the new solution obtained whereas Algorithm 2 do the general process for Local Search based on first improvement strategy.

D. Shaking

In the diversification phase, the shaking procedure is used. This scheme requires neighborhood structures to be used. Our procedure consists of changing the aircraft configuration of a given number of aircraft, randomly obtained in each iteration of the algorithm. The velocity, heading angle or altitude level is modified in order to perturb the best solution found. The variations are also randomly decided to be positive or negative.

Algorithm 2: First improvement local search for the CDR problem

```
Function FirstImprovement(ν, μ, γ, vel, ang, alt,
A, V, T, Z, CP, TL, TG, CTL, CTG);
k = 1 ;
repeat
    j = 1 ;
    repeat
        Move aircraft j by vel nm/h if k = 1;
        Move aircraft j by ang rads. if k = 2;
        Move aircraft j by alt levels if k = 3;
        if f(x) < f(x') then
            j = 0; x' = x;
        else
            Move aircraft j by −vel nm/h if k = 1;
            Move aircraft j by −ang rads. if k = 2;
            Move aircraft j by −alt levels if k = 3;
            if f(x) < f(x') then
                j = 0; x' = x;
            else
                j = j + 1;
            end
        end
    until j > n;
until k > 3;
Updating(j, A, ν, μ, γ, V, T, Z, CP, TL, TG, CTL, CTG)
```

As the parameters *vel*, *ang*, and *alt* are randomly defined, it can occur that, in certain iteration, the parameters could be greater or smaller than the one in the previous iteration.

Algorithm 3 presents the procedure to perturb the solution.

E. Basic VNS algorithm

In the general procedure for the VNS scheme some parameters must be fixed. The maximum size allowed for the neighborhood structures for one VNS iteration, $k_{\max}$, is fixed to 30º as it is the bound for angle variations. The maximum time of computing, $t_{\max}$, which is fixed to as many seconds as the number of aircraft under consideration.

In the VNS scheme we allow to restarting procedure which consists of starting the VNS algorithm from scratch. In many situations, the algorithm is blocked in a local optimum. So, in order to take advantage of the computing time, we use this procedure to allow the algorithm to get other solutions in the solution space that, hopefully, will provide a better solution. Note that random parameters are taken into account in the diversification phase, so other solutions will be explored. The parameter t_r is the time such that if it is exceeded without finding any better solution, the VNS algorithm is restarted. It has been set to $t_{\max}/10$. For the first local search iteration, the parameters *vel*, *ang* and *alt* have been set to 1 nm/h, 1º (which is transformed to rads.) and 1 altitude level.

Algorithm 4 presents the basic VNS scheme for the CRP.

Algorithm 3: Shaking for the CDR problem

```
Function Shaking(ν, μ, γ, vel, ang, alt, A, V, M, CP,
TL, TG, CTL, CTG);
nn ← k  mod n/4 ;
u1 = Rand(0, 1); u2 = Rand(0, 1) ;
u3 = ⌈3 · Rand(0, 1)⌉ ;
vel ← u1k, ang ← u1k, alt ← ⌈u1k⌉ ;
j = 0;
repeat
    if u2 < 0.5 then
        Move aircraft j by vel nm/h if u3 = 1;
        Move aircraft j by ang rads. if u3 = 2;
        Move aircraft j by alt levels if u3 = 3;
    else
        Move aircraft j by −vel nm/h if u3 = 1;
        Move aircraft j by −ang rads. if u3 = 2;
        Move aircraft j by −alt levels if u3 = 3;
    end
    Updating(j, A, ν, μ, γ, V, T, Z, CP, TL, TG, CTL,
    CTG);
    j ← j + 1;
until j = nn;
```

IV. MULTIOBJECTIVE METHODS

Three different multiobjective methods are considered to deal with the aircraft CRP: (1) Lexicographic goal programming; (2) Compromise programming; (3) Double Compromise programming. These methods were firstly introduced for the CRP problem in [6,7].

A. *Objective functions*

We consider three objective functions, one for each maneuver. Furthermore, in some of the multiobjective methods we need additional objective functions as it is described below. The three objective functions related to each maneuver are as follows:

$$\min \sum_{f \in \mathfrak{F}} c_f^{\upsilon} \left| \upsilon_f \right| \tag{2}$$

for velocity changes, being υ_f the velocity variation and c_f^{υ} the unit cost for aircraft f,

$$\min \sum_{f \in \mathfrak{F}} c_f^{\mu} \left| \mu_f \right| \tag{3}$$

for heading angle changes, being μ_f the heading angle variation and c_f^{μ} the unit cost for aircraft f, and,

$$\min \sum_{f \in \mathfrak{F}} c_f^{\gamma} \gamma_f \tag{4}$$

for altitude level changes, being γ_f the number of altitude level changes and c_f^{γ} the unit cost for aircraft f.

Algorithm 4: Steps of the VNS for the CDR problem

```
Function VNS (x, k_max, t_r, t_max);
Calculate CP, TL, TG, CTL, CTG, A ;
FirstImprovement(ν, μ, γ, vel, ang, alt, A, V, T, Z, CP,
TL, TG, CTL, CTG) ;
repeat
    k ← 1;
    repeat
        x' ← Shake(x, k)        /* Shaking */;
        x'' ← FirstImprovement(ν', μ', γ', vel, ang,
        alt, A, V, T, Z, CP, TL, TG, CTL, CTG);
        if f(x'') < f(x) then
            x ← x'; k ← 1   /* Make a move */;
            t_li ← CpuTime() ;
        else
            k ← k + 1     /* Next neighborhood */;
        end
        t ← CpuTime();
    until k = k_max;
    if t − t_li > t_r then
        break;
    end
until t > t_max;
```

B. *Ideal and nonideal values*

In the multiobjective optimization methodology, an initial study is necessary on the different objectives. An ideal value for a single objective function is the best possible value when that objective is optimized subject to the set of constraints. On the opposite, a non-ideal value is the worst value for a single objective function when optimizing another function. These values are represented in the so-called pay-off matrix which provides information about the range in which each single objective varies for any feasible solution. The ideal values lie on the diagonal of the matrix whereas the non-ideal values do on the rest of matrix positions. This matrix is also commonly used to study the degree of conflict among the objectives under consideration.

C. *Lexicographic Goal Programming*

The Goal Programming method was firstly introduced in [12]. For good surveys, see [18] and [31]. It consists of sequentially solving the optimization problem for each objective function allowing a deviation from a given aspiration level for each previously optimized objective. The sequence is given by the priority order. In our case, we have selected the order: Altitude level, heading angle and velocity changes, since we are interested in comfort terms, following general guidelines in [11].

In this way, the procedure to obtain the solution given by the lexicographical Goal Programming is as follows:

1. Obtaining the ideal objective function values for each maneuver by independently optimizing the

corresponding objective function (2), (3) and (4) for VC, TC and AC minimization, respectively.

2. Computation of the maximum deviation that is allowed from the ideal values that have been obtained for AC and TC maneuvers. Those deviations are computed as $\varepsilon_t(z_t^{**}-z_t^*)$ for TC and $\lceil \varepsilon_a(z_a^{**}-z_a^*) \rceil$ for AC, where $\lceil x \rceil$ means the smallest integer value greater or equal than *x*, ε_t and ε_a are user-driven 0–1 fractional values to calibrate the deviations, and z_t^{**} and z_a^{**} are the maximum TC and AC values allowed by the user and known as non-ideal values.

   Notice that the deviation $\varepsilon_a(z_a^{**}-z_a^*)$ is rounded to the closer greater integer number since the difference $z_a^{**}-z_a^*$ in AC is always integer, and its multiplication by a value less than one produces a non-integer rhs and probably without allowing any other AC.

3. TC minimization (3) and the following constraint,

$$\sum_{f\in\mathfrak{F}} c_f^{\gamma}\gamma_f \le z_a^* + \lceil \varepsilon_a(z_a^{**}-z_a^*) \rceil$$

4. VC minimization (2) and the previous and following constraint,

$$\sum_{f\in\mathfrak{F}} (c_f^{\mu^+}\mu_f^+ + c_f^{\mu^-}\mu_f^-) \le z_t^* + \varepsilon_t(z_t^{**}-z_t^*).$$

   where the ideal value z_t^* used in the previous constraint has been obtained in the third step.

D. Compromise Programming

The Compromise Programming method was firstly introduced in [14]. Its main idea lies on the perception that the decision maker prefers the solution closer to the ideal value. In this method, as the objective function minimizes a distance, different norms can be taken into account (generally providing different solutions). In our case we are using the l_1 norm due to its linearity with the aim of normalize the objective function by taking into account the range of variation given by the ideal and the non-ideal values.

The procedure to obtain the solution by the Compromise Programming is as follows.

1. Obtaining the ideal objective function values for each maneuver by independently optimizing the corresponding objective function (2), (3) and (4) for VC, TC and AC minimization, respectively.

2. Minimizing the following objective function (it is the l_1 norm),

$$\frac{\sum_{f\in\mathfrak{F}} c_f^{\upsilon}\left|\upsilon_f\right| - z_v^*}{z_v^{**}-z_v^*} + \frac{\sum_{f\in\mathfrak{F}} c_f^{\mu}\left|\mu_f\right| - z_t^*}{z_t^{**}-z_t^*} + \frac{\sum_{f\in\mathfrak{F}} c_f^{\gamma}\gamma_f - z_a^*}{z_a^{**}-z_a^*} \qquad (5)$$

E. Double Compromise Programming

Sometimes, the solution provided by the Compromise Programming method may result in an excessive deviation from the ideal value, so, in [15] a two-step methodology is presented where firstly is used the l_∞ norm followed by the l_1 norm imposing the maximum deviation obtained by the l_∞ norm.

The procedure to obtain the solution by the Double Compromise Programming is as follows.

1. Obtaining the ideal objective function values for each maneuver by independently optimizing the corresponding objective function (2), (3) and (4) for VC, TC and AC minimization, respectively.

2. Minimizing the following objective function (it is the l_∞ norm),

$$\max\left\{ \frac{\sum_{f\in\mathfrak{F}} c_f^{\upsilon}\left|\upsilon_f\right| - z_v^*}{z_v^{**}-z_v^*}, \frac{\sum_{f\in\mathfrak{F}} c_f^{\mu}\left|\mu_f\right| - z_t^*}{z_t^{**}-z_t^*}, \frac{\sum_{f\in\mathfrak{F}} c_f^{\gamma}\gamma_f - z_a^*}{z_a^{**}-z_a^*} \right\}$$

3. Minimizing the objective function (5) and restricting the changes in each maneuver to be less or equal than the value obtained in the previous step.

V. Preliminary Computational Experiment

In this section we report a preliminary results obtained by applying the VNS algorithm to the CRP. In order to validate the approach, an illustrative example is used, although it is not realistic is commonly used to test the approaches to the CRP. We consider a circumference and locate a certain number of aircraft on it flying towards the circumference center as it is illustrated in Fig. 3. We compare the results with the ones obtained by using the exact mixed integer nonlinear and nonconvex optimization model presented in [6]. The experiment has been performed on a 4xIntel Core i5-2430M, 2.40 GHz, 8 GB RAM with a Linux Xubuntu 14.04 Operative System. The VNS algorithm has been implemented in C++.

Table I reports the solution values for each maneuver obtained by using the state-of-the-art solver Minotaur. The headings are as follows: *Case* represents the case instance Ca-z, where *a* denotes the number of aircraft under consideration and *z* the number of altitude levels that the aircraft are allowed to change; *nc* is the number of pairs of aircraft in conflict situation; z_v^*, z_t^* and z_a^* are the ideal values for VC (nm/h), TC (rad.) and AC (number of altitude levels), respectively;

TABLE I. IDEAL AND NON-IDEAL VALUES

Instance		Minotaur						Gap VNS					
Case	nc	z_v^*	z_t^*	z_a^*	z_v^{**}	z_t^{**}	z_a^{**}	g_v^*	g_t^*	g_a^*	g_v^{**}	g_t^{**}	g_a^{**}
C2-4	1	0.0000	0.0048	0	9.0000	0.0050	1	0.00	-4.17	0.00	-0.89	38.00	0.00
C3-4	3	0.0000	0.0000	0	2.0010	0.0144	3	0.00	0.00	0.00	4.33	21.53	0.00
C4-4	6	0.0000	0.0096	0	17.3330	0.0191	6	0.00	23.96	0.00	-82.82	5.76	0.00

TABLE II. COMPUTING TIMES

Instance	Minotaur						VNS					
Case	t_v^*	t_t^*	t_a^*	t_v^{**}	t_t^{**}	t_a^{**}	t_v^*	t_t^*	t_a^*	t_v^{**}	t_t^{**}	t_a^{**}
C2-4	0.56	0.08	0.02	0.03	0.03	0.03	0.00	0.00	0.00	1.12	0.00	0.00
C3-4	0.17	0.13	0.04	0.09	0.08	0.46	0.00	0.01	0.00	0.04	0.00	1.05
C4-4	71.27	1.09	0.10	1.20	0.21	4.78	0.00	2.12	0.00	1.62	0.00	1.30

z_v^{**}, z_t^{**} and z_a^{**} are the nonideal values for VC, TC and AC, respectively; g_v^*, g_t^*, g_a^*, g_v^{**}, g_t^{**} and g_a^{**} report the gaps between the solution obtained by Minotaur and the VNS approach computed as: $\frac{z_{VNS} - z_M}{z_M} \cdot 100\%$, where z_M and z_{VNS} denote the corresponding solutions obtained by Minotaur and VNS, respectively.

In our approach we avoid altitude levels if it is possible due to comfort reasons, so, it can be observed in Table I that if altitude level changes are not needed to avoid the conflict situations, some ideal values for velocity or heading angle maneuvers could be different from zero as in the cases with even number of aircraft.

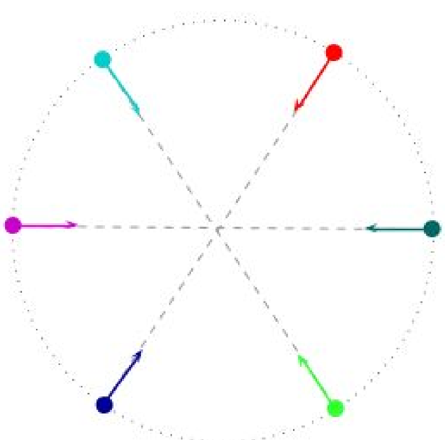

Fig. 3 Illustrative instance

For the cases with odd number of aircraft, velocity changes can avoid the conflict situations (if the aircraft are sufficiently far from each other).

It can be observed in Table I that the gaps between the solutions obtained by Minotaur are, in general, acceptable. VNS obtains reasonable solutions for the ideal values, except for the ideal value for heading angle changes in C4-4. For the non-ideal values, the optimization (even when exact schemes are used) is unstable due to the fact that some objectives are not penalized in the objective function.

Table II reports the computing times needed to solve each instance. The new headings are as follows: t_v^*, t_t^*, t_a^*, t_v^{**}, t_t^{**} and t_a^{**} are the computing times needed to obtain the ideal and nonideal values for each maneuver.

It can be observed in Table II that the computing times for Minotaur grow significantly as the number of aircraft does. For only four aircraft, the computing time exceeds one minute for obtaining the ideal value for velocity changes. However, VNS is able to obtain a good solution in less than 2 secs. except when minimizing heading angle changes for C4-4.

VI. CONCLUSIONS

A VNS metaheuristic approach has been presented to deal with the Conflict Resolution Problem in Air Traffic Management. Due to the nature of the problem, an answer in almost real-time is required. Previous works in the literature deal with the problem but they do not consider the three maneuvers or do need too much computing time to provide a solution. Furthermore, the three maneuvers are different among them and should be treated in a balanced way by considering comfort or economic terms, for instance. We integrate in the VNS algorithm a multiobjective methodology with the aim of providing useful information to the Air Traffic Control officers. We have reported preliminary results by applying our approach in a difficult case showing good results both in computing time and in solution's quality. This is a work-in-progress and some steps must be finished as the implementation of the multiobjective methods, its integration under the same algorithm and its validation in an extensive computational experiment.

ACKNOWLEDGMENT

This research is partially supported by the projects OPTIMOS3 MTM2012-36163-C06-06 and GROMA MTM2015-63710-P, both funded by the Spanish Ministry of Economy Affairs and Competitiveness.

REFERENCES

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

[2] A. Alonso-Ayuso, L.F. Escudero, and 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, 12(39), pp. 3136-3146, 2012.

[3] A. Alonso-Ayuso, L.F. Escudero, F.J. Martín-Campo, and N. Mladenović, "VNS based algorithm for solving a 0-1 nonlinear nonconvex model for the collision avoidance in air traffic management". Electronic notes in discrete mathematics, 39, pp. 115-120, 2012.

[4] A. Alonso-Ayuso, L.F. Escudero, and F.J. Martín-Campo, "On modeling the air traffic control coordination in the collision avoidance problem by mixed integer linear optimization". Annals of Operations Research 1(222), pp. 89-105, 2014.

[5] 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 63(3), pp. 583-596, 2015.

[6] A. Alonso-Ayuso, L.F. Escudero, and F.J. Martín-Campo, "An exact multi-objective mixed integer nonlinear optimization approach for aircraft conflict resolution", TOP, in press, 2015.

[7] A. Alonso-Ayuso, L.F. Escudero, and F.J. Martín-Campo, "Multiobjective optimization for aircraft conflict resolution. A metaheuristic approach". European Journal of Operational Research 248, pp. 691-702, 2016.

[8] A. Alonso-Ayuso, L.F. Escudero, and F.J. Martín-Campo, "Exact and approximate solving of the aircraft collision resolution problem by turn changes". Transportation Science 50(1), pp. 263-274, 2016.

[9] D. Bonini, C. Dupré, and G. Granger, "How ERASMUS can support an increase in capacity in 2020" in Proceedings of the 7^{th} international conference on computing, communications and control technologies Orlando (USA), 2009.

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

[11] C. Cetek, "Realistic speed change maneuvers for air traffic conflict avoidance and their impact on aircraft economics". International Journal of Civil Aviation 1(1), pp. 62-73, 2009.

[12] A. Charnes, W. Cooper, and R. Ferguson, "Optimal estimation of executive compensation by linear programming". Management Science 1(2), pp. 138-151, 1955.

[13] M.A. Christodoulou, and C. Costoulakis, "Nonlinear mixed integer programming for aircraft collision avoidance in free flight" in IEEE Melecon, Dubrovnik, Croacia, 2004.

[14] J. Cochrane, and M. Zeleny, "Multiple criteria decision making". University of South Carolina Press, Columbia, 1973.

[15] L.F. Escudero, "Robust proftolios for mortgage-backed securities" in Zenios SA, Quantitative methods, AI and supercomputers in finance, Unicom, pp. 201-228, 1995.

[16] G. Granger, N. Durand, and J.M. Alliot, "Optimal Resolution of En Route Conflicts", ATM 2001, 4^{th} USA/Europe Air Traffic Management Research and Development Seminar, Santa Fé, USA, 2001.

[17] P. Hansen, N. Mladenović, and J.M. Pérez, "Variable neighbourhood search: methods and applications", 4-OR 6, pp. 319-360, 2008.

[18] J. Ignizio, "Introduction to linear goal programming". Sage Publishing, Beverly Hills, 1985.

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

[20] F.J. Martín-Campo, "The collision avoidance problem: methods and algorithms". Lambert Academic Publishing, Germany, 2012.

[21] N. Mladenović, and P. Hansen, "Variable neighborhood search". Computers & Operations Research 24, pp. 1097-1100, 1997.

[22] N. Mladenović, and P. Hansen, "Variable neighborhood search: Principles and applications". European Journal of Operational Research 130(3), pp. 449-467, 2001.

[23] J. Omer, and J. Farges, "Hybridization of nonlinear and mixed-integer linear programming for aircraft separation with trajectory recovery". IEEE Transactions on Intelligent Transportation Systems 14(3), pp. 1218-1230, 2013.

[24] J. Omer, "A space-discretized mixed-integer linear model for air-conflict resolution with speed and heading maneuvers". Computers & Operations Research 58, pp. 75-86, 2015.

[25] L. Pallottino, E. Feron, and A Bicchi, "Conflict resolution problems for air traffic management systems solved with mixed integer programming". IEEE Transactions on Intelligent Transportation Systems 3(1), pp. 3-11, 2002.

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

[27] D. Rey, C. Rapine, V. Dixit, and S.T. Waller, "Equity-oriented aircraft collision avoidance model". IEEE Transactions on Intelligent Transportation Systems 16(1), pp. 172-183, 2015.

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

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

[30] A.G. Richards, and J.P. How, "Aircraft trajectory planning with collision avoidance using mixed integer linear programming" in American control conference, Anchorage (Alaska), USA, 2002.

[31] C. Romero, "Handbook of critical issues in goal programming". Pergamom Press, Oxford, 1991.

[32] A. Vela, S. Solak, W. singhose, and J. Clarke, "A mixed-integer program for flight level assignment and speed control for conflict resolution" in Proceedings of the joint 48^{th} IEEE conference on decision and control, Shanghai, 2009.