



**HAL**  
open science

## Résolution de conflits aériens par un algorithme à évolution différentielle

Charlie Vanaret, David Gianazza, Jean-Baptiste Gotteland, Nicolas Durand

► **To cite this version:**

Charlie Vanaret, David Gianazza, Jean-Baptiste Gotteland, Nicolas Durand. Résolution de conflits aériens par un algorithme à évolution différentielle. ROADEF 2012, 13ème congrès annuel de la Société Française de Recherche Opérationnelle et d'Aide à la Décision, Apr 2012, Angers, France. pp 550-553. hal-00864898

**HAL Id: hal-00864898**

**<https://hal.science/hal-00864898>**

Submitted on 23 Sep 2013

**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.

# Résolution de conflits aériens par un algorithme à évolution différentielle

Charlie Vanaret, David Gianazza, Jean-Baptiste Gotteland, Nicolas Durand

ENAC/MAIAA, IRIT/APO, Toulouse (France)

{vanaret}@cena.fr

{gianazza, gottelan, durand}@recherche.enac.fr

**Mots-clés :** *trafic aérien, optimisation, évolution différentielle.*

## 1 Introduction

La résolution de conflits entre trajectoires d'avions est un problème fortement combinatoire qui n'a encore jamais été résolu par des méthodes déterministes locales ou globales en gardant des hypothèses réalistes. Les seules approches centralisées permettant de résoudre efficacement des conflits impliquant plus d'une vingtaine d'avions en recherchant un optimum global, sont à ce jour la programmation linéaire mixte [1] [2] et les algorithmes évolutionnaires [3] [4].

Dans [1], les hypothèses sur la vitesse des avions sont fortes et peu réalistes dans le cadre d'une utilisation opérationnelle : vitesses constantes pour des résolutions par évitement latéral, ou changements de vitesses instantanés pour des résolutions en vitesse. Dans [3], un algorithme génétique fait évoluer une population d'individus encodant des manœuvres de déviation latérales pour les avions initialement en conflit. La modélisation proposée s'appuie sur des variables discrétisées : les temps de début et fin de manœuvres sont choisis avec un pas de temps fixé, et les angles de déviation sont pris parmi des valeurs discrètes. Dans [4], une modélisation similaire est utilisée, avec un algorithme de colonies de fourmis. Les algorithmes évolutionnaires donnent des résultats très convaincants, avec des hypothèses relativement réalistes, permettant notamment la prise en compte d'incertitudes sur les vitesses des avions, contrairement à l'approche par programmation linéaire mixte de [1] (vitesses constantes, manœuvres synchronisées). Cependant, les équipements aujourd'hui présents dans les centres de contrôle et à bord des avions ne sont pas prévus pour gérer des manœuvres définies en temps.

Dans la présente communication, nous proposons d'utiliser un algorithme à évolution différentielle [5] opérant sur des variables continues, avec des manœuvres latérales définies par des points géographiques directement utilisables par les FMS<sup>1</sup> actuels. Notons que les définitions de manœuvre en temps ou en distance sont équivalentes pour les trajectoires nominales, mais conduisent à des modélisations tout à fait différentes des incertitudes.

## 2 Modélisation

Chaque manœuvre d'évitement est modélisée par un segment rectiligne écartant l'avion  $i$  de sa trajectoire initiale, suivi d'un segment de retour. Une manœuvre est totalement déterminée par un triplet  $x_i = (d_i, \alpha_i, l_i)$  où  $d_i$  est la distance (le long de la route initiale) à laquelle débute la manœuvre,  $\alpha_i$  est l'angle de déviation, et  $l_i$  est la longueur du premier segment de déviation. Le segment de retour se fait par une nouvelle déviation d'angle  $(-2\alpha_i)$  jusqu'à interception de la route initiale. La modélisation fournit une suite de points géographiques (Fig. 1).

---

1. FMS : Flight Management System.

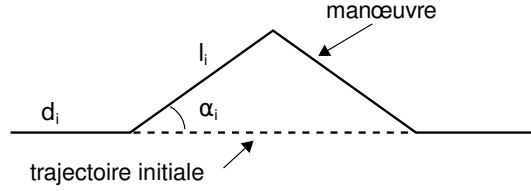


FIG. 1 – Modèle de point tournant

### 3 Critère à optimiser

Les contraintes du problème sont d'avoir des trajectoires séparées d'au moins une distance  $d$ , dite norme de séparation, à tout instant (Fig. 2). La fonction d'évaluation de ces contraintes est la durée cumulée des conflits entre les avions deux à deux.

Le coût associé à une déviation de trajectoire peut prendre plusieurs formes : rallongement de la trajectoire globale, longueur des segments de manœuvre, surface de l'écart à la route initiale, temps de déviation, etc.

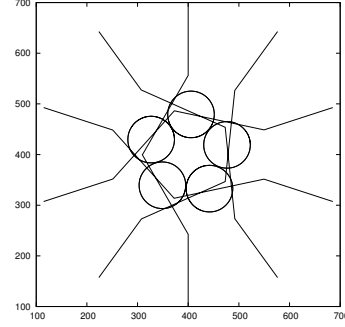


FIG. 2 – Norme de séparation entre 5 avions à un temps donné

#### 3.1 Conflits entre trajectoires

Chaque paire d'avions  $(i, j)$  étant caractérisée par des positions  $X_i(t)$  et  $X_j(t)$  évoluant au cours du temps  $t$ , la contrainte de séparation entre trajectoires se traduit simplement par  $\|X_j(t) - X_i(t)\|^2 > d^2, \forall t$ . Deux avions sont en conflit lorsque leurs trajectoires prévues ne respectent pas cette contrainte de séparation.

Afin d'évaluer la durée des conflits entre deux trajectoires  $i$  et  $j$ , on calcule les subdivisions respectives  $([P_i^{(k)} P_i^{(k+1)}])_k$  et  $([P_j^{(k)} P_j^{(k+1)}])_k$  de telle sorte que deux segments  $[P_i^{(k)} P_i^{(k+1)}]$  et  $[P_j^{(k)} P_j^{(k+1)}]$  soient rectilignes et parcourus en un temps identique  $\delta^{(k)}$  (Fig. 3).

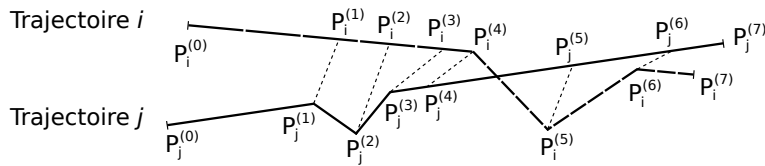


FIG. 3 – Subdivision de deux trajectoires

En notant  $D(t) = \|X_j(t) - X_i(t)\|^2 - d^2$ , on résout l'équation  $D(t) = 0$ . Lorsque les vitesses des avions sont constantes, on obtient une équation du second degré en  $t$  avec deux racines  $t_1 < t_2$  dans le cas où le discriminant est positif. En prenant  $t_{min}^{(k)} = \max(t^{(k)}, \min(t_1, t^{(k)} + \delta^{(k)}))$  et  $t_{max}^{(k)} = \max(t^{(k)}, \min(t_2, t^{(k)} + \delta^{(k)}))$ , on définit la durée de conflit entre  $[P_i^{(k)} P_i^{(k+1)}]$  et  $[P_j^{(k)} P_j^{(k+1)}]$  comme  $\Delta t_{ij}^{(k)} = t_{max}^{(k)} - t_{min}^{(k)}$ . La durée des conflits entre les trajectoires  $i$  et  $j$  est alors  $\Delta t_{ij} = \sum_k \Delta t_{ij}^{(k)}$ .

Lorsque les vitesses ne sont pas constantes, et en supposant une accélération constante par segments, la résolution de  $D(t) = 0$  se traduit par une équation du quatrième degré en  $t$ . Une autre approche, plus simple et largement utilisée, consiste à discrétiser les trajectoires avec un pas de temps constant, suffisamment faible, et à calculer les distances entre points des trajectoires pour déterminer quand elles deviennent inférieures à  $d$ .

### 3.2 Coût des manœuvres

Pour prendre en compte à la fois le retard dû à la manœuvre et la durée de la manœuvre elle-même, on mesure la surface  $S_i = l_i^2 \cos(\alpha_i) \sin(\alpha_i)$  occupée par la manœuvre de l'avion  $i$ .

### 3.3 Fonction objectif

La fonction objectif (ou fitness) à maximiser est donnée par :

$$f = \begin{cases} \frac{1}{2 + \sum_{i < j} \Delta t_{ij}} & \text{si } \exists (i, j), \Delta t_{ij} > 0 \\ \frac{1}{2} + \frac{1}{2 + \sum_i S_i} & \text{si } \forall (i, j), \Delta t_{ij} = 0 \end{cases} \quad (1)$$

Une situation avec conflits fournit une valeur de fitness inférieure à 0.5, tandis que la valeur de fitness d'une situation sans conflits est supérieure à 0.5.

## 4 Evolution différentielle

L'évolution différentielle est une métaheuristique à base de populations introduite par Storn et Price en 1997 [5], inspirée par les algorithmes génétiques (opérateurs de mutation et de croisement) et les stratégies géométriques de recherche (du type simplexe de Nelder-Mead [6]). Dans notre cas, chaque individu  $x$  de la population représente une situation à  $n$  avions, et est caractérisé par un vecteur  $(x_1, \dots, x_n)^T$  représentant les manœuvres latérales des  $n$  avions.

Une opération combinant mutation et croisement est appliquée à chaque individu afin de faire évoluer la population de manière progressive, jusqu'à obtenir des solutions satisfaisantes. Pour chaque vecteur  $x^{(k)}$  de la génération  $k$ , on construit le vecteur  $x^{(k+1)}$  à partir de trois vecteurs  $a^{(k)}, b^{(k)}, c^{(k)}$  aléatoirement choisis dans le reste de la population, tous différents et différents de  $x^{(k)}$ . Les composantes  $x_i^{(k+1)} = (d_i^{(k+1)}, \alpha_i^{(k+1)}, l_i^{(k+1)})$ ,  $i \in \{1, \dots, n\}$  sont calculées de la manière suivante :

$$x_i^{(k+1)} = \begin{cases} a_i^{(k)} + F \times (b_i^{(k)} - c_i^{(k)}) & \text{si } (i = R) \text{ ou } (r_i < CR) \\ x_i^{(k)} & \text{sinon} \end{cases} \quad (2)$$

où  $F \in [0, 2]$  est appelé poids différentiel,  $R$  tiré aléatoirement dans  $\{1, \dots, n\}$  garantit qu'au moins une des composantes de  $x^{(k+1)}$  diffère de celle de  $x^{(k)}$ ,  $r_i \sim \mathcal{U}(0, 1)$  est tiré aléatoirement pour chaque composante et comparé à la probabilité de croisement  $CR \in [0, 1]$ .  $x^{(k+1)}$  remplace alors  $x^{(k)}$  dans la population s'il améliore la fonction objectif.

## 5 Résultats et conclusion

Les figures 4 à 7 illustrent différentes étapes de la résolution d'un conflit à 50 avions, pour des positions initiales disposées en cercle, et des trajectoires convergeant simultanément vers le centre. Deux solutions optimales existent : dans la première, tous les avions tournent à droite, dans la seconde, tous les avions tournent à gauche (configurations dites en rond-point). Le fait que le problème est fortement contraint peut être illustré en générant aléatoirement un million de situations à  $n$  avions et en dénombrant le nombre de situations sans conflits. Pour  $n = 5$ , 497 situations sont sans conflits. Pour  $n \geq 8$ , aucune situation générée n'est sans conflits.

L'algorithme a été testé avec les paramètres du tableau 1, pour un temps d'exécution de 14 secondes et 40040 évaluations de la fonction objectif. La figure 8 montre que les conflits ont été résolus à l'itération 950. Le tableau 2 présente les résultats obtenus en moyenne pour 20 exécutions successives de l'algorithme pour différents nombres d'avions. La colonne  $S$  donne le nombre d'exécutions pour lesquelles les conflits ont été résolus.

Ces résultats préliminaires montrent que l'algorithme à évolution différentielle opérant dans le domaine continu est une alternative valable aux méthodes exposées en introduction. La

modélisation choisie permet par ailleurs de traduire directement les manœuvres proposées en une suite de points géographiques.

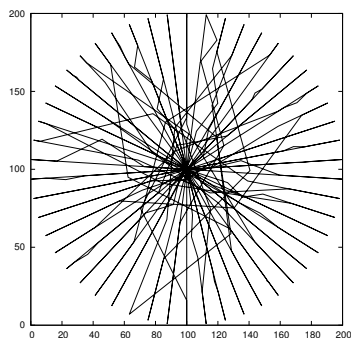


FIG. 4 – Itération 1

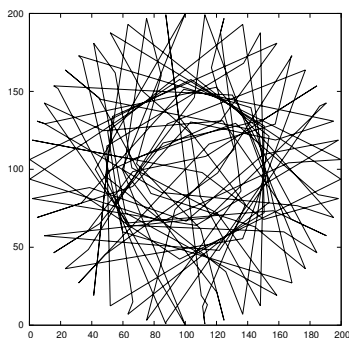


FIG. 5 – Itération 250

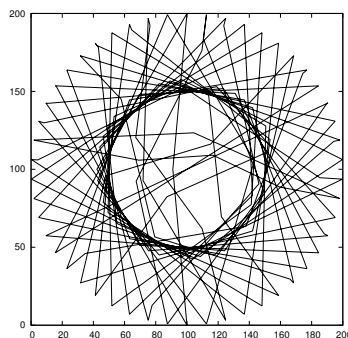


FIG. 6 – Itération 500

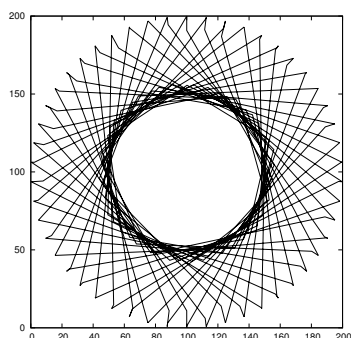


FIG. 7 – Itération 1000

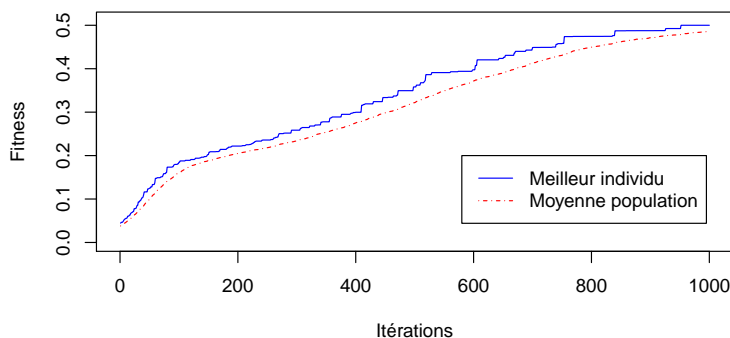


FIG. 8 – Evolution de la fitness au cours des itérations

Paramètres	Valeur
$NP$ : taille de la population	40
$F$ : poids différentiel	0.7
$CR$ : probabilité de croisement	0.05
Nombre d'itérations	1000
$d$ : norme de séparation	5 NM
$\alpha$ : angle de la manœuvre	$[-\frac{\pi}{6}, \frac{\pi}{6}]$

TAB. 1 – Paramètres de l'algorithme de résolution de conflits à 50 avions

Avions	Max	Min	Moyenne	Ecart type	$S$
3	0.507403	0.507403	0.507403	0.000000	20
5	0.504614	0.501771	0.502987	0.000953	20
10	0.500372	0.500181	0.500268	0.000048	20
20	0.500033	0.500021	0.500028	0.000003	20
30	0.500010	0.500009	0.500010	0.000000	20
40	0.500006	0.482338	0.498730	0.004130	18
50	0.500004	0.436032	0.496805	0.013942	19

TAB. 2 – Résultats pour 20 exécutions de l'algorithme (2000 itérations,  $NP = 100$ ,  $F = 0.7$  et  $CR = 0.1$ )

## Références

- [1] 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, Mars 2002.
- [2] S. Caferi, P. Brisset, and N. Durand. A mixed-integer optimization model for air traffic deconfliction. In *Proceedings of the Toulouse Global Optimization Workshop*, pages 27–30, 2010.
- [3] N. Durand, J.M. Alliot, and J. Noailles. Automatic aircraft conflict resolution using genetic algorithms. In *Proceedings of the Symposium on Applied Computing*, 1996.
- [4] N. Durand and J.M. Alliot. Un algorithme de colonie de fourmis pour résoudre des conflits aériens. In *ROADEF*, 2010.
- [5] R. Storn and K. Price. Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, pages 341–359, 1997.
- [6] J. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, pages 308–313, 1965.