

Algorithmes Génétiques et Programmation Linéaire appliqués à la résolution de conflits ATC.

Frédéric Médioni Nicolas Durand Jean-Marc Alliot

ENAC*

CENA†

ENAC

1 Introduction

La génération de trajectoires optimales d'évitement pour des aéronefs en route est un problème complexe qui n'est pour l'instant résolu que de manière empirique par les contrôleurs. Cependant plusieurs études ont été menées pour automatiser la résolution de ce problème, qui s'attachent plus ou moins à la notion d'optimalité :

- Le projet SAINTEX [AL92] utilise une approche de type *système expert*, avec l'introduction dans certains cas d'une tentative d'optimisation.
- Karim Zeghal propose dans [Zeg93] des techniques réactives pour l'évitement robustes au perturbation, mais qui laissent de côté la recherche de l'optimalité des trajectoires.
- Le projet AERA 3 [NFC⁺83], [Nie89b], [Nie89a], ne recherche l'optimalité que dans le cas d'un conflit à deux avions.
- ARC 2000 [K⁺89], [FMT93], optimise les trajectoires des avions au fur et à mesure de leur prise en compte, en considérant comme définitivement fixées les trajectoires des avions traitées précédemment : il ne recherche pas l'optimalité globale.
- Nicolas Durand (citer) utilise les algorithmes génétique pour la résolution des conflits en route, avec recherche de l'optimalité globale.

Nous présentons ici une autre manière d'utiliser les algorithmes génétiques dans ce but, en les couplant à un programme d'optimisation linéaire.

2 Position du problème.

2.1 Hypothèses.

Les trajectoires d'évitement de conflits à proposer à des avions doivent répondre à plusieurs critères : elles doivent être compatibles avec les performances des avions considérés et être assez simples, pour pouvoir

être facilement transmises aux pilotes, que cette transmission soit effectuée par des contrôleurs, ou qu'elle soit elle aussi automatisée. Ainsi le nombre de changement de cap, d'altitude, de vitesse doit être limité.

Les hypothèses que nous retiendrons sont en fait plus restrictives : nous considérerons dans toute la suite que les trajectoires des avions sont contenues dans un plan horizontal. Nous nous limiterons de plus aux conflits entre des avions volant à la même altitude.

2.2 La trajectoire d'évitement optimale

Nous allons voir que sous les hypothèses décrites ci-dessus, deux modes d'évitement apparaissent comme les plus naturels : l'évitement *par un point tournant* et l'évitement *par offset*. Ces modes d'évitement ont été présentés et comparés dans [Dur94]. Nous rappelons ici l'essentiel de ces résultats. Nous nous limiterons d'abord au cas d'un conflit à deux avions, un seul avion modifiant sa trajectoire, avant d'étendre les résultats concernant l'évitement *par offset* à un conflit à n avions, où $n > 2$.

On considère deux avions, notés a_1 et a_2 , dont les positions à un instant t sont donnés par les couples de coordonnées $(x_1(t), y_1(t))$ et $(x_2(t), y_2(t))$. A l'instant t_o les avions a_1 et a_2 sont respectivement aux points O_1 et O_2 , et souhaitent atteindre à t_f les points D_1 et D_2 . On suppose qu'en l'absence de déviation les trajectoires des avions se coupent en un point C . Les trajectoires d'évitement doivent alors satisfaire la contrainte suivante :

$$\forall t \in [t_o, t_f], (x_1(t) - x_2(t))^2 + (y_1(t) - y_2(t))^2 \geq d^2 \quad (1)$$

où d est la norme de séparation horizontale choisie.

On cherche alors la trajectoire optimale d'évitement pour l'avion a_1 (on ne modifie pas celle de l'avion a_2), c'est à dire la trajectoire satisfaisant la contrainte (1) et entraînant l'allongement minimal par rapport à la trajectoire non déviée (les vitesses étant constantes en norme, minimiser l'allongement revient à minimiser le retard). On montre que dans ce cas la trajectoire de l'avion a_1 dans le référentiel lié à l'avion a_2 comporte trois phases :

1. un segment de droite sur lequel la contrainte (1) n'est pas saturée ;
2. un arc de cercle sur lequel la contrainte (1) est saturée ;

*Ecole Nationale de l'Aviation Civile

†Centre d'Etudes de la Navigation Aérienne

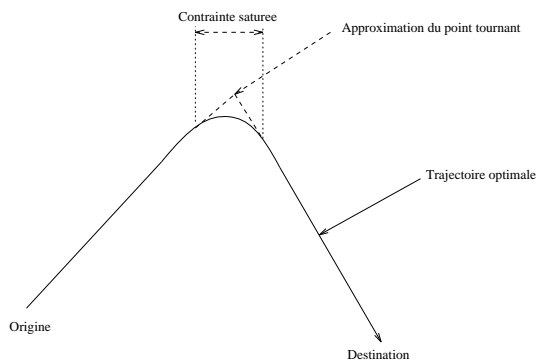


Figure 1: Méthode du point tournant

- un deuxième segment de droite sur lequel la contrainte 1 n'est pas saturée.

Les deux segments de droite sont tangents au cercle de centre C et de rayon d . (voir figure)

Ce type de trajectoire présente l'inconvénient d'être difficile à suivre avec précision par un pilote, c'est pourquoi on introduit d'autres modes d'évitement, entraînant un allongement légèrement supérieur à celui entraîné par la trajectoire optimale décrite ci-dessus, mais plus facilement réalisables que celle-ci.

2.3 L'évitement par point tournant

Le principe de l'évitement par point tournant consiste à prolonger les deux parties rectilignes de la trajectoire décrite ci-dessus, et à remplacer l'arc de cercle par un unique point tournant (voir figure 1).

Nicolas Durand montre dans [Dur94] que cette modélisation permet d'obtenir une bonne approximation de la trajectoire optimale (allongement augmenté de moins de 5%) pourvu que la résolution du conflit commence suffisamment tôt et que l'angle d'incidence ne soit pas trop faible. Ces conditions ne sont pas très contraignantes : ainsi, avec un angle de 90° entre les trajectoires des deux avions, le *temps d'anticipation* (c'est à dire le temps restant avant le conflit à l'instant où on commence à dévier l'avion a_1) suffisant pour que l'augmentation de l'allongement par rapport à la trajectoire optimale soit inférieure à 5% n'est que de 2 fois le temps nécessaire pour parcourir la norme de séparation.

2.4 L'évitement par offset

L'évitement par offset (voir figure 2) consiste à faire suivre à l'avion dévié une trajectoire parallèle à sa trajectoire optimale, décalée par rapport à celle-ci d'un côté ou de l'autre. L'évitement par offset comporte trois phases ; durant chacune de ces phases la trajectoire de l'avion est portée par un segment de droites :

- La phase de mise en offset : l'avion est dévié de sa trajectoire suivant un angle appelé *angle de mise en offset*, jusqu'à ce qu'il atteigne l'écartement

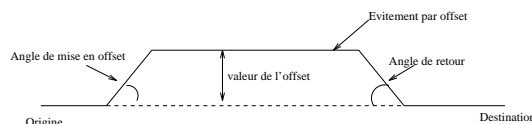


Figure 2: Méthode de l'offset

désiré par rapport à sa trajectoire d'origine. On définit ainsi le *sens de l'offset*, et l'angle de mise en offset est toujours compté positivement.

- L'avion suit ensuite une trajectoire parallèle à sa trajectoire d'origine, pendant un temps variable, mais en tout cas suffisant à l'évitement du deuxième avion. La distance entre la trajectoire dévié et la trajectoire d'origine sera appelée dans la suite de ce mémoire la *valeur de l'offset*.
- L'avion revient sur sa trajectoire initiale avec un angle par rapport à celle-ci appelé *angle de retour*.

L'angle de mise en offset comme l'angle de retour sur la trajectoire peuvent être fixés. Dans la suite, ils seront fixés et égaux entre eux. Cet angle sera noté β . La durée de la seconde phase (temps pendant lequel la trajectoire de l'avion reste parallèle à sa trajectoire) n'a pas d'incidence directe sur l'allongement de la trajectoire. Cependant, il est préférable que ce temps soit le plus réduit possible, pour éviter que pendant cette phase, la trajectoire de l'avion dévié n'interfère avec celle d'un autre avion, et parce que il est préférable que l'avion dévié ait rejoint sa trajectoire d'origine au moment de changer de *secteur de contrôle*. (voir à ce propos (Daniel))

Comme on va le voir dans la section suivante, la modélisation de l'évitement *par offset* mène à un problème d'optimisation linéaire avec contraintes linéaires, aux prix de simplifications supplémentaires que nous précisons. Nous étendrons ce résultat au cas d'un conflit à n avions. Cette linéarité est le principal avantage de la modélisation par offset par rapport à la modélisation par point tournant, qui donne, elle, un retard plus proche de celui qu'entraîne la trajectoire d'évitement optimale. (voir [Dur94], p. 21)

3 La modélisation par offset pour un conflit à n avions

On considère n avions (a_1 à a_n) volant dans un secteur donné pendant l'intervalle de temps $[t_o, t_f]$, et susceptibles d'entrer en conflit les uns avec les autres. La condition nécessaire et suffisante pour que les n avions soient séparés sur l'intervalle $[t_o, t_f]$ est clairement que les avions soient séparés deux à deux, c'est à dire que la contrainte (1) soit satisfaite pour chaque couple d'avions.

On supposera dorénavant que $t_o = 0$. On s'intéressera à deux avions, a_i et a_j . On note ϕ_{ij} l'angle entre les trajectoires non déviées, de a_i et a_j (voir figure ??),

t_{ij}^i (resp. t_{ij}^j) l'instant auquel la trajectoire non déviée de l'avion a_i (resp. a_j) coupe celle de l'avion a_j (resp. a_i), et v_i et v_j les normes des vecteurs vitesse des avions (par hypothèse constantes sur $[t_o, t_f]$).

En exprimant en fonction du temps $t \in [t_o, t_f]$ les coordonnées des avions a_i et a_j , dans le repère ortho-normé de centre C_{ij} (point d'intersections deux trajectoires) et dont l'axe des abscisses est dirigé par le vecteur vitesse de a_j , on obtient la condition de séparation suivante, pour $t \in [t_o, t_f]$:

$$\begin{aligned} & (v_i - v_j)^2 t^2 \\ & + (2v_i v_j \cos(\phi_{ij})(t_{ij}^i + t_{ij}^j) \\ & - 2v_i^2 t_{ij}^i - 2v_j^2 t_{ij}^j) t \\ & + v_i^2 t_{ij}^i{}^2 + v_j^2 t_{ij}^j{}^2 \\ & - 2v_i t_{ij}^i v_j t_{ij}^j \cos(\phi_{ij}) - d^2 \geq 0 \end{aligned} \quad (2)$$

Cette condition est une inéquation du second degré en t . Comme elle est vérifiée pour t assez grand, elle est vérifiée pour tout t ssi son discriminant est négatif, ce qui donne, après simplification :

$$v_i^2 v_j^2 \sin(\phi_{ij})^2 (t_{ij}^i - t_{ij}^j)^2 \geq d^2 (v_i^2 + v_j^2 - 2v_i v_j \cos(\phi_{ij})) \quad (3)$$

Cette condition donne les deux conditions suivantes, selon l'ordre de passage des avions a_i et a_j au point C_{ij} , point d'intersection de leurs deux trajectoires :

- Si l'avion a_i passe derrière l'avion a_j , on obtient :

$$(t_{ij}^i - t_{ij}^j) v_i v_j \sin(\phi_{ij}) \geq d \sqrt{v_i^2 + v_j^2 - 2v_i v_j \cos(\phi_{ij})} \quad (4)$$

- Si l'avion a_j passe derrière l'avion a_i , on obtient :

$$(t_{ij}^j - t_{ij}^i) v_i v_j \sin(\phi_{ij}) \geq d \sqrt{v_i^2 + v_j^2 - 2v_i v_j \cos(\phi_{ij})} \quad (5)$$

Il faut noter que la condition (3) est une condition nécessaire et suffisante de séparation des deux avions pour $t \in]-\infty, +\infty[$, alors que nous cherchions une CNS de séparation pour $t \in [t_o, t_f]$. Sur cet intervalle (3) est une condition suffisante, mais pas nécessaire, de séparation. On applique ainsi des contraintes trop strictes, et on risque de passer à côté de solutions optimales. On peut cependant limiter ce risque en n'appliquant cette méthode que dans le cas où un risque de conflit entre a_i et a_j a été repéré sur l'intervalle de temps $[t_o, t_f]$. (Voir 9 !!! pour des détails sur la prévision et l'évaluation de tels risques.)

La mise en offset des avions a_i et a_j a pour effet d'introduire un décalage d'une trajectoire par rapport à l'autre et d'influer ainsi sur l'instant auquel la trajectoire de chacun des deux avions coupe celle de l'autre.

Considérons l'effet de l'évitement par offset pour un couple d'avions (a_i, a_j) , où $1 \leq i < j \leq n$. Pour chacun des deux avions a_i et a_j , l'évitement par offset a trois effets sur les temps $t_{i,j}^i$ et $t_{i,j}^j$. Ces effets dépendent de l'angle de mise en offset, noté β , du sens de l'offset, et de sa valeur. Ainsi un offset de l'avion a_i d'une valeur d_i aura les effets suivants :

- $t_{i,j}^i$ est augmenté, quelque soit le sens de l'offset, du retard dû à la mise en offset, c'est à dire de $\frac{d_i \tan(\frac{\beta}{2})}{v_i}$.
- Si la trajectoire de l'avion a_i est déviée vers l'extérieur de l'angle formé par les trajectoires des deux avions, $t_{i,j}^i$ est augmenté du retard dû au déplacement de l'avion a_i , c'est à dire de $\frac{d_i \cot(\phi_{i,j})}{v_i}$. Si la trajectoire de a_i est déviée vers l'intérieur de cet angle, $t_{i,j}^i$ est diminué de cette valeur.
- De même, si la trajectoire de l'avion a_i est déviée vers l'extérieur de l'angle formé par les trajectoires des deux avions $(a_i$ et $a_j)$, $t_{i,j}^j$ est aussi augmenté du retard dû au déplacement de la trajectoire de l'avion a_j , c'est à dire de $\frac{d_i}{v_j \sin(\phi_{i,j})}$.

Ainsi, en modifiant $t_{i,j}^i$ et $t_{i,j}^j$ dans les conditions (4) et (5), on obtient une inéquation linéaire en d_i et d_j , dont les coefficients dépendent du sens de l'offset de chaque avion, et de l'ordre de passage des avions.

Par exemple si les offset des deux avions sont à l'extérieur, et si a_i passe derrière a_j , on obtient :

$$\begin{aligned} & (t_{i,j}^i + \frac{d_i \tan(\frac{\beta}{2})}{v_i} + \frac{d_i \cot(\phi_{i,j})}{v_i} \\ & + \frac{d_j}{v_i \sin(\phi_{i,j})} - t_{i,j}^j - \frac{d_j \tan(\frac{\beta}{2})}{v_j} \\ & - \frac{d_j \cot(\phi_{i,j})}{v_j} - \frac{d_i}{v_j \sin(\phi_{i,j})}) v_i v_j \sin(\phi_{i,j}) \\ & - d \sqrt{v_i^2 + v_j^2 - 2v_i v_j \cos(\phi_{i,j})} \geq 0 \end{aligned} \quad (6)$$

La condition évoquée ci-dessus est, comme on l'a vu, une condition suffisante de séparation pour le couple d'avions a_i et a_j pendant la seconde phase de l'évitement par offset. Nous ne nous intéresserons dans la suite qu'à ces contraintes linéaires, et nous ne prendrons pas en compte celles qui traduisent la séparation des avions pendant les phases de mise en offset et de retour à leurs trajectoires d'origines, qui ne sont pas linéaires. (Dans la plupart des cas, on pourra considérer que si les avions sont mis en offset suffisamment tôt avant le conflit, et maintenus en offset suffisamment longtemps, ils seront séparés pendant les phases de mise en offset et de retour sur leurs trajectoires initiales.)

On a considéré jusqu'ici que les droites portant les trajectoires non déviées des avions a_i et a_j étaient sécantes (et comme on l'a vu que les trajectoires des avions se coupaient pour $t \in [t_o, t_f]$). Dans le cas où les

trajectoires sont parallèles, on obtient aussi des conditions linéaires. La discussion sur l'ordre de passage des deux avions est alors remplacée par le choix entre "a_i passe à gauche de a_j" et "a_j passe à gauche de a_i".

4 Un problème linéaire, mais très fortement combinatoire

On a donc obtenu des contraintes linéaires, rendant compte, après les simplifications décrites ci-dessus, de séparation des avions deux à deux, ce qui donne $\frac{n(n-1)}{2}$ contraintes. On leur ajoute les contraintes : $d_i \geq 0$ pour $1 \leq i \leq n$, ce qui fait donc en tout pour un conflit à n avions $\frac{n(n+1)}{2}$ contraintes. (On aura le plus souvent n contraintes supplémentaires : en effet on peut borner la valeur des offset pour ne retenir que des solutions "raisonnables".)

La fonction à minimiser est la somme des retards pour les n avions. Pour chaque avion, le retard entraîné par l'évitement par offset est dû aux phases de mise en offset et de retour sur sa trajectoire d'origine. Ce retard dépend des angles de mise en offset et de retour sur la trajectoire. Ainsi, pour l'avion a_i le retard entraîné par un offset de valeur d_i est égal à :

$$2 \frac{d_i \tan(\frac{\beta}{2})}{v_i}$$

La somme des retards pour tous les avions est donc :

$$S(d_1, \dots, d_n) = 2 \sum_{i=1}^n \frac{d_i \tan(\frac{\beta}{2})}{v_i}$$

Cette fonction est linéaire en d_i pour $1 \leq i \leq n$ On a donc, quand on a fixé le sens de l'offset de chaque avion, et l'ordre de passage des avions, pour obtenir les valeurs d'offset entraînant un retard global minimal, à résoudre un problème d'optimisation linéaire à $\frac{n(n+1)}{2}$. Ce problème peut se résoudre facilement, à l'aide par exemple d'un algorithme du type "simplexe". cependant on n'obtiendra alors qu'un optimum local. On peut même n'obtenir aucun résultat : il est possible que pour des sens de déviations et des ordres de passage donnés, le problème linéaire correspondant n'ait pas de solution. (En fait c'est même souvent le cas, surtout si on borne la valeur des offsets)

Les coefficients des contraintes linéaires dépendant des sens de déviation et de l'ordre de passage de chaque avion, il faut fixer ces données avant de lancer l'optimisation. On a alors $2^{\frac{n(n+1)}{2}}$ combinaison possible, où n est le nombre d'avion en conflit. Dès que le nombre d'avion impliqués dans le conflit augmente, il devient impossible d'essayer toute ces combinaisons possible : pour 5 avions, il y en a 32768, et pour 6 avions, il y en a 2097152. Cette combinatoire est trop importante pour qu'on puisse utiliser des algorithmes

d'énumération exhaustive couplés à des programmes d'optimisation linéaire.

A ce point, deux questions se posent :

1. N'y a-t-il pas un moyen simple et rapide d'établir que toute une classe de combinaisons des sens de déviation et d'ordre de passage ne mène à aucune solution et de l'éliminer ?
2. Ne peut-on pas regrouper plusieurs de ces combinaisons en une même composante connexe, quitte à prendre des contraintes non linéaires ?

Une réponse affirmative à ces questions aurait permis de réduire le nombre de combinaisons à prendre en compte, soit en éliminant certaines, soit en regroupant plusieurs. Malheureusement dans le cas, pourtant le plus simple, d'un conflit à deux avions, la réponse à ces questions est très complexe (elle dépend notamment du rapport entre l'angle de mise en offset et de l'angle entre les trajectoires des avions), elle le devient encore plus pour trois avions, et ne permet pas, quel que soit le cas de figure de réduire la combinatoire de manière significative.

5 L'utilisation des algorithmes génétiques.

Par sa combinatoire très importante, ce problème nous a semblé pouvoir être résolu par une utilisation des algorithmes génétiques, couplé à un algorithme de programmation linéaire du type "simplexe".

5.1 les algorithmes utilisés

Pour la résolution des problèmes linéaires, nous utilisons le programme lp_solve, élaboré par la *Design Automation Section, Eindhoven University of Technology*. Il fonctionne selon un algorithme du type *simplexe*, tel qu'il est décrit dans [OH68].

Nous utilisons des algorithmes génétiques tels qu'ils sont décrits dans [Gol89] et [Mic92] : des éléments de population sont dans un premier temps générés aléatoirement. Puis, pour chaque élément est calculée une fonction appelée *fitness*, évaluant l'adéquation de l'élément au problème à résoudre. Dans un second temps, on opère une sélection sur les éléments de la population, en fonction de leur *fitness*. Puis on applique aléatoirement, avec les probabilités P_c et P_m, des opérateurs de croisement et de mutation, pour diversifier la population. A ce stade, une nouvelle population est créée, et on applique à nouveau ce procédé, de manière itérative, à chaque itération, on obtient une nouvelle population, par analogie avec l'évolution des espèces vivantes, on parlera alors de nouvelle *génération*.

Pour éviter les optima locaux, nous utilisons avec l'algorithme génétique la technique du *sharing*, décrite dans [eXY].

5.2 Codage pour l'algorithme génétique

Nous codons le sens de déviation de chaque avion, et l'ordre de passage des avions, dans le chromosome d'un élément de la population par un entier. Les n premiers bits de cet entier codent les sens des offset des n avions en cas d'évitement (1 pour un affset à gauche, 0 pour un offset à droite, si l'avion n'a pas besoin d'être dévié on prévoit tout de même un sens de déviation, et on aura 0 pour la valeur de l'offset). Les $\frac{n(n-1)}{2}$ bits suivant codent les ordres de passage pour chaque couple d'avion. On a pris pour coder le chromosome un entier contenant 32 bits, ce qui limite l'utilisation du programme à la résolution d'un conflit impliquant 7 avions. Pour des conflits impliquant plus de 7 avions, on peut utiliser une structure contenant deux entiers (ou plus).

L'opérateur de mutation consiste à choisir un élément de la population avec la probabilité P_m , et de modifier un des $\frac{n(n+1)}{2}$ premiers bits de l'entier codant son chromosome (choisi de manière uniformément aléatoire).

L'opérateur de croisement consiste à choisir avec la probabilité P_c deux éléments de la population, les "parents", et à les remplacer par deux autres éléments, les "enfants", dont les chromosomes sont obtenus à partir de ceux des "parents" de la manière suivante : on détermine de manière aléatoire, pour chaque bit, duquel des deux parents chaque enfant héritera le bit correspondant. Nous utiliserons en fait, pour favoriser la diversification au début du processus, une variante de l'opérateur de croisement inspirée des techniques de recuit simulé : On compare, après avoir appliqué le croisement, chaque enfant au meilleur des parents. Si le "fils" est meilleur que son "père", il prend sa place ; s'il est moins bon, il prend sa place, avec une probabilité de loi $e^{-\frac{\Delta}{T}}$, où Δ est la différence entre la fitness du père et celle du fils, et où T décroît avec les générations. T est appelé la *température de recuit*.

5.3 Utilisation

Le chromosome d'un élément de population code un sens de déviation donnée pour chaque avion, et pour chaque couple d'avions à un ordre de passage donné. On peut donc obtenir, grâce au programme d'optimisation linéaire, pour chaque élément de population, les valeurs des offsets des n avions menant au retard minimal, et la valeur de ce retard.

On peut ainsi prendre pour la fitness d'un élément de la population décroissante du retard minimal correspondant à cet élément. Cependant on a vu qu'une combinaison de sens de déviation et d'ordre de passage donnée pouvait mener à un problème d'optimisation linéaire infaisable, c'est à dire un problème dont toutes les contraintes ne peuvent pas être simultanément satisfaites. Si on affecte à tous les éléments de la population correspondant une fitness nulle, les algorithmes linéaires perdent beaucoup de leur efficacité (pour un conflit à plus de trois avions, dès qu'on introduit une borne supérieure pour la valeur des offsets,

les éléments correspondant à un problème infaisable sont très majoritaires).

Pour différencier ces éléments, nous supprimons alors, une à une, (toujours de manière aléatoire, pour éviter d'éventuels effets de symétrie) des contraintes. Nous prenons alors pour la fitness une fonction décroissante du plus petit nombre de contraintes à supprimer pour obtenir un problème faisable. Ces différentes fonction sont ajustées selon les données du problème (nombre d'avions, vitesses des avions, norme de séparation choisie) afin que la fitness des éléments de la population correspondant à des problèmes faisables sans suppression de contraintes soit toujours nettement supérieure à celle des éléments nécessitant la suppression de contraintes.

Une variante de cette méthode, plus adaptée au le problème considéré, et aussi, plus efficace, consiste à regrouper d'abord pour chaque avion, les contraintes dans lesquelles la valeur de l'offset de cet avion intervient, et à supprimer les contraintes, non plus une à une, mais "avion par avion". Cela revient à retirer simultanément toutes les contraintes dans lesquelles intervient une variable donnée. Les résultats présentés plus loin ont été obtenus en utilisant cette dernière méthode.

6 Résultats

L'évaluation des résultats obtenus pose le problème suivant : il est difficile d'obtenir par un autre méthode une solution dont on sache qu'elle est optimale, pour la comparer à la solution obtenue grâce à l'algorithme ci-dessus. Dans le cas d'un conflit mettant en jeu 6 avions, il est encore possible (en un peu plus de trois heures) de traiter par un programme d'optimisation linéaire de manière systématique et déterministe toutes les combinaisons possibles de sens de déviation et d'ordre de passage (il y en a 2097152), et de comparer tous les résultats obtenus, et donc de déterminer la meilleure solution.

C'est ce que nous avons fait, en considérant la configuration suivante : les avions vont à la même vitesse (400 kts), ils sont au temps t_0 régulièrement répartis sur un demi cercle de centre C (et de rayon 100 Nm), et leurs trajectoires se coupent en C . Il est à noter que les symétries de ce problème, utiles à l'homme pour se le représenter, et permettant d'être sûr qu'il y aurait bien, sans déviation, un conflit impliquant les cinq avions, ne changent pas la manière dont il est traité par la machine.

On obtient alors deux solutions optimales équivalentes : soit les cinq premiers avions sont déviés vers la droite, tandis que le sixième n'est pas dévié, et si $i < j$, a_i passe derrière a_j , soit le premier avion n'est pas dévié, et les cinq suivants sont déviés vers la gauche, et si $i < j$, a_i passe devant a_j .

La méthode décrite dans cette article mène aux deux mêmes solutions. Ainsi, on peut aussi comparer le coût en temps de la méthode présentée ici dans le cas d'un conflit à 6 avions, au coût de la méthode consistant à évaluer les retards correspondant à toutes les

	Moyenne	Ecart type
Temps (s)	99.0	51.8
Nb d'évaluation	9696	6707
Nb d'appels	23193	11818

Table 1: Nombre d'appels au programme linéaire

combinaisons de sens de déviation et d'ordre de passage.

Pour un conflit à 6 avions, nous avons lancé notre programme 50 fois sur l'exemple présenté ci-dessus, en prenant à chaque fois une initialisation différente pour les fonctions aléatoires. Nous avons fait tourner notre programme pendant 98 générations, avec 150 éléments de population. Sur ces 50 fois, nous avons obtenu une des deux solutions optimales en moins de 20 générations 14 fois, 9 fois entre 20 et 40 générations, 13 fois entre 40 et 80 générations, 3 fois entre 80 et 98 générations. 11 fois (sur 50), nous n'avons pas obtenu une de ces solutions optimales après 98 générations. En fait, le nombre de génération au bout duquel on obtient une solution optimale n'est pas très significatif. Nous présentons dans le tableau 1 la moyenne et l'écart-type, sur les 39 fois où une solution optimale a été atteinte, du temps CPU auquel elle a été atteinte, du nombre d'évaluations de chromosomes effectués (Nb d'évaluations), et du nombre d'appel au programme d'optimisation linéaire (Nb d'appels).

Ces résultats ont été obtenus sur une station HP 720. Un parcours exhaustif de tous les cas de figure nécessiterait, comme nous l'avons vu précédemment, 2097152 appels au programme d'optimisation linéaire. Il est à noter que chaque génération mène à environ 250 évaluations (chacun des 150 éléments de la population est évalué, et d'autres évaluations sont nécessaires au moment des croisements et des mutations). Les premières génération peuvent nécessiter jusqu'à 1500 appels au programme linéaire, tandis que pour des générations ultérieures, ce nombre descend à moins de 350. En effet, comme nous l'avons vu, lorsque le cas de figure codé par le chromosome d'un élément de population mène à un problème linéaire infaisable, nous prenons comme *fitness* pour cet élément le nombre d'avions à supprimer pour obtenir un problème linéaire qui ait une solution. (Nous rappelons que par la *suppression* d'un avion, nous entendons celle des contraintes dans lesquelles la valeur de l'offset de cet avion intervient). Pour obtenir ce nombre d'avions à *supprimer*, nous appelons le programme linéaire en supprimant d'abord un avion, puis deux, etc jusqu'à obtenir une solution. De génération en génération, les éléments de la population s'améliorent globalement, le nombre de ceux qui correspondent à des problèmes infaisable diminue : pour un nombre stable d'évaluation, le nombre d'appels au programme linéaire diminue alors fortement.

7 Conclusion

Les résultats obtenus pourront sans doute être améliorés, en affinant les différents coefficients qui entrent en jeu dans les algorithmes décrits ci-dessus (notamment dans le calcul de la fitness des éléments). Ils sont cependant déjà, dans l'état actuels, encourageant, et montrent que l'utilisation conjointe des algorithmes génétiques et de la programmation linéaire peut, dans un tel cas être intéressante.

Cette méthode permet d'allier aux avantages de l'optimisation linéaire (notamment l'obtention rapide d'un optimum), ceux des algorithmes génétiques, qui permettent notamment, d'éviter les optima locaux, et de proposer plusieurs solutions à un problème, à travers la diversité des éléments de la population, ce qui est un avantage non négligeable, si l'on veut dans un premier pas, non pas automatiser le contrôle aérien, mais aider les contrôleurs, en leur proposant des trajectoires d'évitements.

Bibliographie

- [AL92] Luc Angerand et Hervé LeJeannic. Bilan du projet SAINTEX. Rapport technique, CENA, 1992. CENA/R92009.
- [Dur94] Nicolas Durand. Modélisation des trajectoires d'évitement pour la résolution de conflits en route. Rapport technique, Centre d'Etudes de la Navigation Aérienne, Février 1994.
- [eXY] Noel Germy et Xiaodong Yin. A fast genetic algorithm with sharing scheme using cluster analysis methods in multimodal function optimization. Rapport technique, Université Catholique de Louvain, Laboratoire d'Electronique et d'Instrumentation.
- [FMT93] Xavier Fron, Bernard Maudry, et Jean-Claude Tumelin. Arc 2000 : Automatic radar control. Rapport technique, EUROCONTROL, 1993.
- [Gol89] David Goldberg. *Genetic Algorithms*. Addison Wesley, 1989. ISBN : 0-201-15767-5.
- [K+89] Fred Krella et al. Arc 2000 scenario (version 4.3). Rapport technique, EUROCONTROL, April 1989.
- [Mic92] Zbigniew Michalewicz. *Genetic algorithms + data structures = evolution programs*. Springer-Verlag, 1992. ISBN : 0-387-55387.
- [NFC+83] W.P. Niedringhaus, I. Frolow, J.C. Corbin, A.H. Gisch, N.J. Taber, et F.H. Leiber. Automated En Route Air Traffic Control Algorithmic Specifications: Flight Plan Conflict Probe. Rapport technique, FAA, 1983. DOT/FAA/ES-83/6.

- [Nie89a] W.P. Niedringhaus. Automated planning function for AERA3: Manoeuver Option Manager. Rapport technique, FAA, 1989. DOT/FAA/DS-89/21.
- [Nie89b] W.P. Niedringhaus. A mathematical formulation for planning automated aircraft separation for AERA3. Rapport technique, FAA, 1989. DOT/FAA/DS-89/20.
- [OH68] W. Orchard-Hays. *Advanced Linear Programming Computing Techniques*. McGraw-Hill, 1968.
- [Zeg93] Karim Zeghal. Techniques réactives pour l'évitement. Rapport technique, ONERA, June 1993.