Résolution de conflits aériens par algorithmes génétiques

Nicolas Durand, Jean-Marc Alliot

▶ **To cite this version:**

Nicolas Durand, Jean-Marc Alliot. Résolution de conflits aériens par algorithmes génétiques. Revue de l'Association Aéronautique et Astronautique de France, 1996, pp xxxx. hal-00935204

HAL Id: hal-00935204
https://enac.hal.science/hal-00935204

Submitted on 25 Apr 2014

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.

Résolutions de conflits aériens par algorithmes génétiques

Nicolas Durand Jean-Marc Alliot

20 Juin 1996

1 Introduction

Le trafic aérien a connu une progression très importante pendant les dernières décennies. Le nombre de données à traiter augmentera encore avec la mise en place de nouveaux moyens de communication entre le sol et les avions (Data-Link). Parallèlement, les avions s'équipent de moyens sophistiqués de navigation (FMS[1], GPS[2]) permettant des tenues plus précises de trajectoires.

Dans ce cadre, il faut s'interroger sur l'évolution des techniques de contrôle pour les années à venir. On se retrouve alors pris entre deux positions caricaturales : l'automatisation complète ou le refus complet de toute automatisation. Les tenants de l'une et l'autre approche ont des discours qui relèvent plus de l'émotionnel que du rationnel.

La voie moyenne consiste sans doute à penser un système qui, dans un premier temps, serait susceptible de fournir une aide à la détection et à la résolution des conflits, et pourrait, à la demande de l'opérateur, effectuer automatiquement certaines résolutions. Le Centre d'Etudes de la Navigation Aérienne s'est intéressé à cette hypothèse dans le cadre de certains de ses projets, comme SPECTRA[Pla93].

D'autres projets, comme ARC-2000 (développé par le Centre Expérimental Eurocontrol de Brétigny), ont exploré des voies plus futuristes. Les problèmes d'évolution des systèmes de contrôle sont discutés plus en détail dans [DAM93], et l'on trouvera un large panorama des différents projets d'automatisation dans [Dur96].

Dans tous les cas, il faut savoir prévoir les trajectoires des avions, détecter les conflits, regrouper ces conflits par paquets, et donner des manœuvres simples les résolvant, etc. Ces problèmes sont aujourd'hui identifiés, mais les techniques à employer pour les résoudre le sont moins. On peut globalement classer les approches employés pas les différentes équipes en deux catégories, qui ne sont d'ailleurs pas sans rappeler deux écoles de pensée du monde de l'Intelligence Artificielle :

- la première approche consiste à construire un modèle, dit "cognitif", du contrôleur, spécialement dans ses modes d'appréhension des paramètres du conflit, et d'utiliser ce modèle dans un calculateur afin de construire des outils de filtrage de l'information et d'assistance électronique au contrôleur. Cette problématique souffre directement des limitations propres aux approches dites "expertes" : coût très élevé de développement et de maintenance, problème de généralisation des maquettes au cas général, dégradation brutale de l'expertise aux limites du domaine, faillibilité du système expert, etc ([Dre84, Dre79, Dre87, Fox90, Dav82])

- La seconde approche, que nous qualifierons de pragmatique, consiste à étudier mathématiquement le problème et à appliquer les algorithmes susceptibles de donner les meilleurs résultats possibles sans trop se soucier de savoir si ces algorithmes reproduisent ou non le raisonnement humain.

Cette dernière approche a été utilisée par le projet AERA-III [Cel90, SPSS83, NFC+83, Nie89b, Nie89a, PA91] aux Etats-Unis, ou plus récemment le projet ARC-2000 [K+89, FMT93, MG94] du Centre Expérimental Eurocontrol. Le travail que nous avons réalisé depuis 1992, et que nous allons présenter dans cet article, s'inscrit également dans cette lignée.

[1] Flight Management System
[2] Global Positionning System

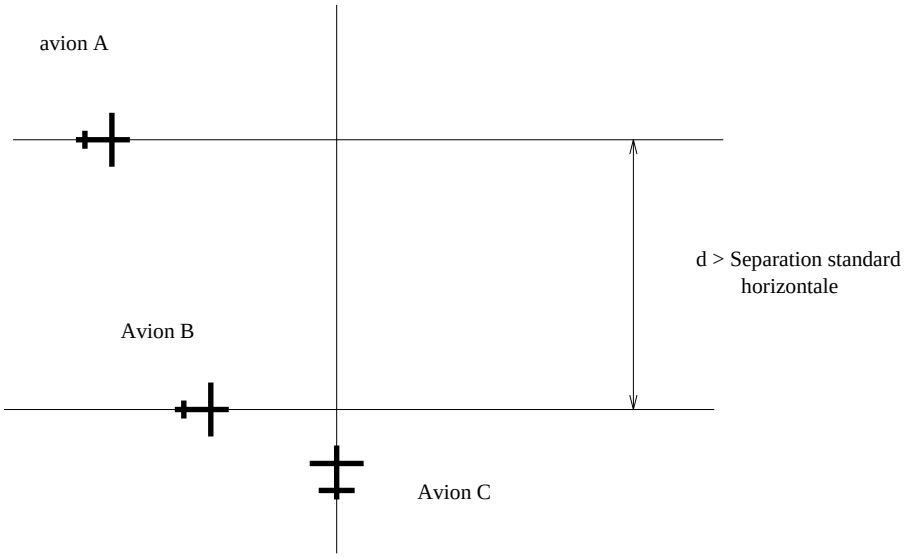

FIG. 1: Exemple de cluster.

2 Quelques définitions

Le but premier du contrôle du trafic aérien est d'assurer la sécurité des aéronefs. Avant toute chose, nous allons introduire le vocabulaire indispensable à la description du problème :

Contrôle en route : il s'agit du contrôle à l'extérieur des zones entourant les aéroports (dans ces zones, on parle de *contrôle d'approche*). Les avions utilisent alors le plus souvent des routes préétablies, les *couloirs aériens* (en anglais *airways*). Ces couloirs sont des *tubes* de sections rectangulaires.

Ces routes contournent les parties d'espace réservées aux militaires et permettent aux contrôleurs d'avoir une visualisation plus aisée de la situation spatiale des avions.

Plan de vol : il contient tous les éléments indicatifs décrivant le vol prévu pour un avion (heure de départ, niveau de vol, route prévue).

Séparations : on définit une distance horizontale exprimée en milles nautiques[3] (NM), la *séparation horizontale*[4], et une distance verticale exprimée en pieds[5] (ft) : la *séparation verticale*[6]. On dit que deux avions sont *séparés* quand la distance qui sépare leurs projections sur un plan horizontal est supérieure à la séparation horizontale OU quand la distance qui sépare leurs projections sur un plan vertical est supérieure à la séparation verticale.

Conflit élémentaire : deux avions sont dits en *conflit* quand ces avions ne sont plus *séparés*. Si l'on se fixe une durée T, deux avions seront dits en conflit potentiel pendant T, si durant le temps T, ils ont une probabilité non nulle d'être en conflit.

Cluster : un *cluster* d'avions est une fermeture transitive d'avions en conflits potentiels. Si un avion A est en conflit avec B à l'instant t et B est en conflit avec C à l'instant $t + \Delta t$ avec $\Delta t < H$ alors A, B et C appartiennent au même cluster. Ainsi dans la figure 1, les avions A et B ne sont jamais en conflit mais ils appartiennent au même cluster. Une déviation de C pour éviter B peut lui permettre d'éviter A. L'optimisation du conflit devra être globale.

L'expression *conflit à n avions* signifie en fait cluster à n avions.

[3] 1 mille nautique vaut 1852 m (1 minute d'angle sur un méridien).
[4] Elle est de 8 NM en France mais devrait bientôt descendre à 5 NM.
[5] 1 pied vaut 30,48 cm.
[6] Elle vaut 1000 ft au dessous du FL 290 et 2000 au dessus.

3 Conflit à deux avions : approche mathématique.

Il est indispensable de tenter de résoudre analytiquement un problème et d'évaluer sa complexité théorique avant d'aborder n'importe quelle autre approche. Les principaux travaux existants dans ce domaine ont été effectués par N. Durand dans sa thèse [Dur96, DAAS94, DAN94, DAN96b, DCA96, DAN96a, DA96]. N. Durand montre notamment [Dur96] qu'il n'est pas possible d'envisager de résoudre des conflits en modifiant la vitesse des avions excepté dans les phases de descente. En effet, les avions ont des marges de manœuvre réduites sur leurs vitesses lorsqu'ils sont en croisière. La résolution en vitesse sera donc réservée au contrôle d'approche. On supposera dans la suite les avions contraints en vitesse et stables en croisière.

3.1 Résolution par commande optimale

Si l'on s'intéresse à minimiser l'allongement de trajectoires, la théorie de la commande optimale [Cle90, MZ88, BH91, ME88, Sch90] permet de dégager certains résultats généraux : lorsque la contrainte de séparation n'est pas saturée, les trajectoires des avions sont rectilignes uniformes, leurs caps sont modifiés uniquement lorsque la contrainte de séparation est saturée. La contrainte sur les vitesses ne permet pas de résoudre analytiquement le conflit. Cependant, si l'on ne dévie que l'un des deux avions, on peut mettre en évidence une équation implicite donnant l'allongement minimal de la trajectoire de l'autre avion. On montre alors qu'un conflit est d'autant plus difficile à résoudre que l'angle d'incidence des trajectoires est faible et que les avions ont des vitesses voisines.

3.2 Complexité de l'ensemble des solutions sans conflit.

Si l'on ne dévie que l'un des deux avions et que la direction du vecteur vitesse de l'avion dévié n'est modifiée de façon discrète qu'en un nombre fini d'instants, on peut montrer, en utilisant le theórème des résidus, que l'espace des trajectoires admissibles pour l'avion dévié se compose de deux composantes connexes qui correspondent aux trajectoires obtenues quand l'avion dévié passe devant ou derrière l'avion non dévié.

Un conflit comprenant n avions peut contenir $\frac{n\,(n-1)}{2}$ conflits élémentaires. L'espace des solutions admissibles peut contenir $2^{\frac{n\,(n-1)}{2}}$ composantes connexes. Dans la pratique, toutes les composantes connexes n'ont pas besoin d'être explorées. Néanmoins, la présence théorique d'autant d'espaces disjoints et l'impossibilité de connaître à priori lequel contient la solution optimale nous a orienté vers des méthodes d'optimisation globale plutôt que vers des méthodes locales.

3.3 Approche numérique

On peut observer les trajectoires optimales grâce à un outil d'optimisation locale sous contraintes non linéaires tel que LANCELOT [CGT92][7]. Pour l'appliquer, les trajectoires d'avions sont discrétisées en 20 points. Les contraintes de séparation des avions sont vérifiées en chaque point de discrétisation. Dans les exemples présentés, la norme de séparation est de 8 nautiques et les avions parcourent en moyenne 120 nautiques à une vitesse de 400 nœuds.

La figure 2 donne six façons de résoudre un conflit à cinq avions. 6 points de départ différents ont été donnés à LANCELOT pour la recherche de la solution optimale donnant 6 solutions localement optimales situées dans des composantes connexes différentes. On remarquera que les solutions, apparemment très différentes, génèrent des allongements moyens très voisins.

Le temps de calcul nécessaire est très long (de l'ordre de plusieurs heures). On ne peut envisager une résolution en temps réel par ce type d'algorithme.

[7] Large And Non linearly Constrained Extended Lagrangian Optimization Technique

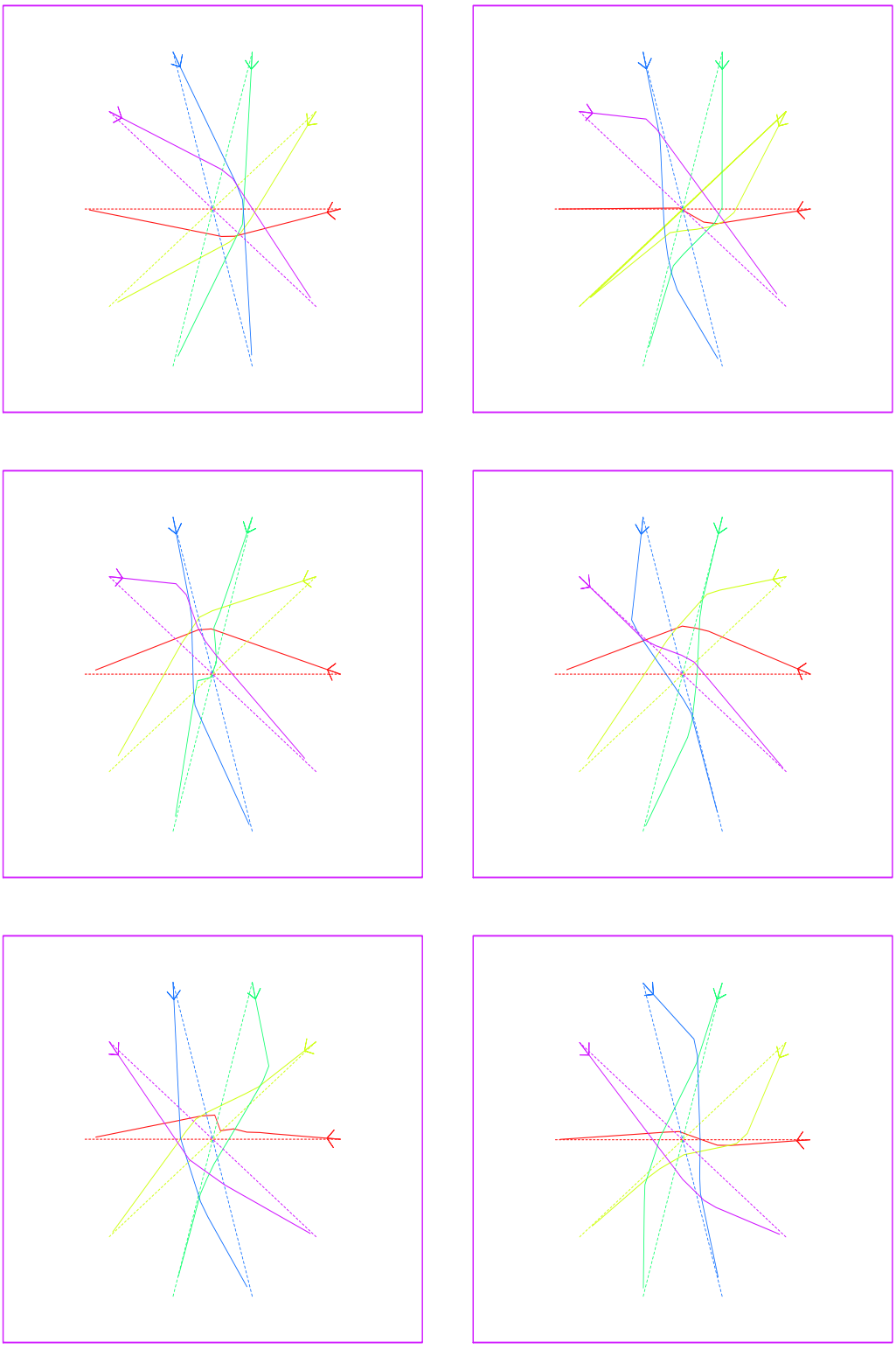

FIG. 2: De haut en bas et de gauche à droite, allongements moyens : 2.16, 2.21, 2.22, 2.20, 2.19 et 2.17 degrés.

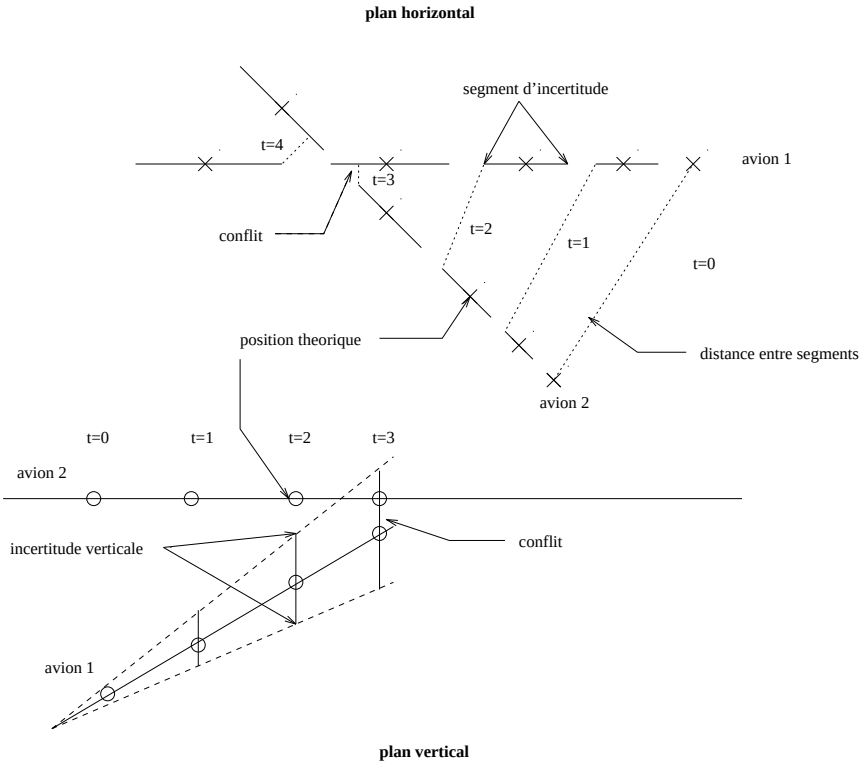

FIG. 3: Modélisation de l'incertitude.

3.4 Influence des incertitudes

Si un avion de ligne est capable de suivre précisément une route et une altitude, maîtriser précisément sa vitesse sol n'est pas possible en raison notamment des vents. Il est d'autre part impossible de corriger en permanence la vitesse car les réacteurs ont un régime de consommation idéal dont il ne faut pas trop s'écarter. Enfin, ils ne doivent pas, pour des raisons mécaniques, être soumis à des changements de régime permanents. Pour tenir compte des incertitudes sur les vitesses, on peut modéliser les avions non par des points matériels mais par des segments de droites. Il suffit alors de vérifier, pour que la contrainte horizontale soit respectée, que les segments de droite qui modélisent les avions soient séparés par la séparation standard (voir figure 3). Dans le plan vertical, l'incertitude est beaucoup plus forte, on maîtrise très mal le taux de montée d'un avion qui dépend de paramètres aussi variés que la masse, où la mise en route de la climatisation. En conséquence, le même principe que dans le plan horizontal est repris mais les pourcentages d'incertitude seront beaucoup plus forts.

4 Modélisation simplifiée des trajectoires

4.1 Modélisation point tournant et offset

La modélisation par Contrôle Optimal permet de dégager certaines propriétés sur les trajectoires optimales et sur la structure de l'espace des solutions admissibles : les trajectoires non contraintes sont rectilignes et uniformes à l'optimum. Dans la pratique, la durée de saturation des contraintes, compte-tenu de la vitesse élevée des avions et de la faible densité de trafic, est très faible comparée à la durée de vol sans saturation de contrainte (dans la mesure où les conflits ne sont pas à angle trop faible). En conséquence, la plupart des trajectoires optimales peuvent être approchées par des trajectoires rectilignes par morceaux. On peut alors définir deux types de manœuvres simples qui permettent de résoudre la plupart des conflits élémentaires. La modé-

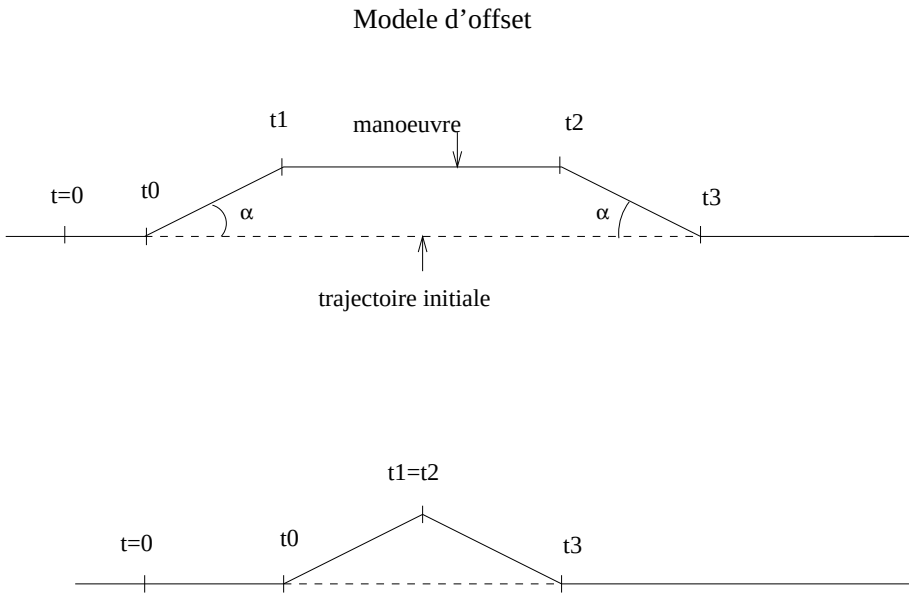

FIG. 4: Modèles de manœuvre.

lisation par point tournant[8] consiste à modifier le cap d'un avion de plus ou moins 10, 20 ou 30 degrés[9] et de ramener ensuite l'avion sur sa trajectoire initiale en modifiant le cap de l'avion de deux fois la déviation précédente, mais dans le sens inverse. La modélisation par offset reprend la précédente mais on rajoute une branche parallèle avant la phase de retour sur la trajectoire. Cette modélisation est indispensable pour résoudre des conflits de rattrapage. Ces deux types de manœuvres sont simples pour le pilote et pour le contrôleur et peuvent être décrites par quelques paramètres. On pourra trouver dans [Dur96] des mesures quantitatives donnant la perte d'optimalité liée à ces manœuvres. Elle est généralement très faible. Nous ne présentons dans cet article que des manœuvres d'évitement dans le plan horizontal. Le modèle peut-être facilement généralisé au plan vertical, sans augmentation notable de la complexité.

Une manœuvre est donc déterminée par les 4 paramètres suivants :

- la date de début du virage d'éloignement t_0 ;
- la date de fin du virage d'éloignement t_1 ;
- la date de début du virage de retour sur la trajectoire initiale t_2.
- l'angle de déviation de la manœuvre : α ;

Les différents t_i représentent des dates relatives. L'instant $t = 0$ correspond au début de la prévision. A cette date, la position de l'avion est connue sans incertitude. On la considère comme *la position initiale* de l'avion. Le virage de retour a le même angle α mais dans le sens opposé.

Un tel modèle réduit la taille du problème de façon significative. Pour un conflit impliquant n avions, la dimension de l'espace de recherche sera de $4n$. Ceci permet de résoudre des gros conflits sans avoir à explorer un domaine de solutions admissibles trop vaste.

[8] L'incertitude sur les vitesses des avions et sur leurs taux de montée fait apparaître dans le cas de la modélisation point tournant un temps "idéal" de début de manœuvre [Dur96].

[9] Les angles de déviation sont discrétisés de façon à rendre les manœuvres standard.

4.2 Déroulement en temps réel

Un des objectifs retenus est de pouvoir détecter et résoudre les conflits en temps réel. Pour ce, fixons un délai Δ. La situation sera revue toutes les Δ secondes à partir de nouvelles données. Le système dispose alors d'un temps inférieur à Δ pour analyser la nouvelle situation et trouver les trajectoires optimales sans conflit pour l'ensemble des avions impliqués dans la fenêtre d'anticipation F_a. Le reste de l'intervalle Δ est consacré à la transmission des manœuvres du sol vers le bord. Le système ne peut pas changer la trajectoire prévue de chacun des avions pendant les Δ premières secondes de F_a. Ces trajectoires correspondent à la prévision effectuée au tour précédent. Par contre, le système reste libre de modifier toutes les trajectoires entre Δ et F_a à l'exception des branches de retour en point tournant ou en offset. En effet, cette partie de la manœuvre est complètement déterminée par le début de la manœuvre.

5 Résolution par algorithmes génétiques

Nous avons pu mettre en évidence l'inefficacité d'une méthode locale pour la résolution de conflits. Parmi les méthodes d'optimisation globale, les algorithmes génétiques ont la particularité de ne requérir que peu d'information sur la fonction d'évaluation et de pouvoir détecter simultanément plusieurs optima locaux proches de l'optimum. C'est principalement ce dernier point qui a motivé le choix de tels algorithmes pour notre problème. En effet, en vue de créer un outil d'aide à la décision, il peut être intéressant de rechercher non pas la solution optimale mais un ensemble de solutions différentes proches de l'optimum.

De plus, les AGs sont relativement robustes et ne requièrent *a priori* aucune hypothèse de régularité sur la fonction à optimiser (leur efficacité dépend cependant fortement de la régularité de cette dernière). La fonction d'évaluation que l'on veut optimiser peut être le résultat d'une simulation, ce qui sera le cas pour le problème de résolution de conflits.

5.1 Principes généraux des algorithmes génétiques

Les algorithmes génétiques sont des algorithmes d'optimisation s'appuyant sur des techniques dérivées de la génétique et de l'évolution naturelle : croisements, mutations, sélection, etc. Ils ont déjà une histoire relativement ancienne puisque les premiers travaux de John Holland sur les systèmes adaptatifs remontent à 1962 [Hol62]. L'ouvrage de David Goldberg [Gol89] a largement contribué à les vulgariser.

Un algorithme génétique recherche le ou les extrema d'une fonction définie sur un espace de données. Pour l'utiliser, on doit disposer des cinq éléments suivants :

1. Un principe de codage de l'élément de population. Cette étape associe à chacun des points de l'espace d'état un *chromosome* (structure de données).

2. Un mécanisme de génération de la population initiale. Ce mécanisme doit être capable de produire une population d'individus non homogène qui servira de base pour les générations futures.

3. Une fonction à optimiser. Celle-ci retourne une valeur de $\Re^+$ appelée *fitness* ou fonction d'évaluation de l'individu.

4. Des opérateurs permettant de diversifier la population au cours des générations et d'explorer l'espace d'état. L'opérateur de croisement recompose les gènes d'individus existant dans la population, l'opérateur de mutation a pour but de garantir l'exploration de l'espace d'états.

5. Des paramètres de dimensionnement : taille de la population, nombre total de générations ou critère d'arrêt, probabilités d'application des opérateurs de croisement et de mutation.

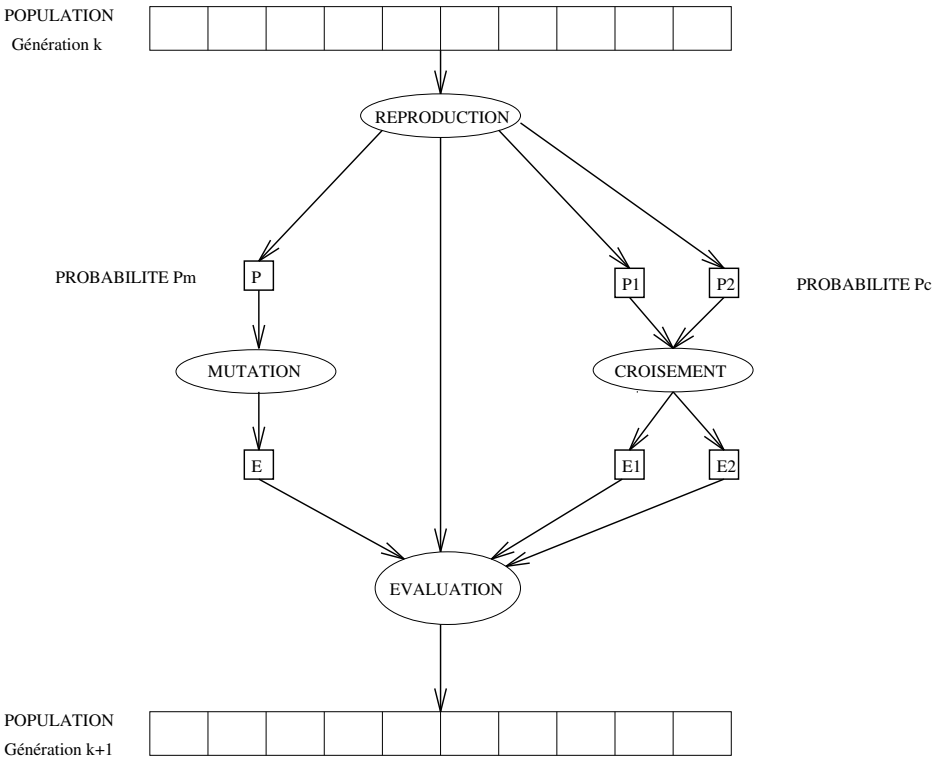

FIG. 5: Principe général des algorithmes génétiques

Le principe général du fonctionnement d'un algorithme génétique est représenté sur la figure 5 : on commence par générer une population d'individus de façon aléatoire. Pour passer d'une génération k à la génération $k+1$, les trois opérations suivantes sont répétées pour tous les éléments de la population k. Des couples de parents P_1 et P_2 sont sélectionnés en fonction de leurs adaptations. L'opérateur de croisement leur est appliqué avec une probabilité P_c (généralement autour de 0.6) et génère des couples d'enfants C_1 et C_2. D'autres éléments P sont sélectionnés en fonction de leur adaptation. L'opérateur de mutation leur est appliqué avec la probabilité P_m (P_m est généralement très inférieur à P_c) et génère des individus mutés P'. Le niveau d'adaptation des enfants ($C1$,C_2) et des individus mutés P' sont ensuite évalués avant insertion dans la nouvelle population. Différents critères d'arrêt de l'algorithme peuvent être choisis :

- Le nombre de générations que l'on souhaite exécuter peut être fixé à priori. C'est ce que l'on est tenté de faire lorsque l'on doit trouver une solution dans un temps limité.
- L'algorithme peut être arrêté lorsque la population n'évolue plus ou plus suffisamment rapidement.

5.2 Critères à optimiser

Nous sommes face à un problème d'optimisation multi-critères. Nous avons retenu les critères suivants :

- Les trajectoires doivent respecter les normes de séparation.
- Le retard induit pour chaque avion doit être aussi faible que possible.
- Le nombre d'avions déviés et le nombre total de manœuvres doivent être aussi faibles que possibles, il se peut donc que les retards ne soient pas partagés par tous les avions.

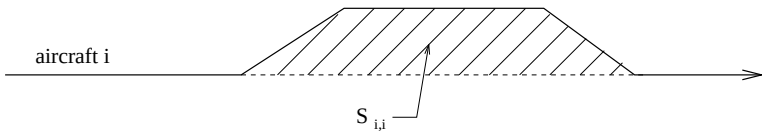

FIG. 6: Surface occupé par la manœuvre

– La durée d'une manœuvre doit être aussi faible que possible pour pouvoir rendre le plus tôt possible un avion disponible pour une autre manœuvre.

Pour un cluster comprenant n avions, nous construisons une matrice F de taille $n \times n$. Si $i \neq j$, alors $F_{i,j}$ mesure la gravité du conflit entre l'avion i et l'avion j. S'il n'y a pas de conflit, $F_{i,j} = 0$, $F_{i,j}$ croit avec la gravité du conflit. $F_{i,i}$ mesure la pénalisation de la trajectoire de l'avion i provoqué par ses manœuvres.

5.2.1 Calcul des éléments de la diagonale

Pour prendre en compte à la fois le retard dû à la manœuvre et la durée de la manœuvre elle-même, on peut mesurer la surface S occupée par la résolution (figure 6). Cette surface augmente à la fois avec le retard induit et la durée de la manœuvre[10]. Pour minimiser le nombre de manœuvres, on ajoute à S le nombre de manœuvres multiplié par un certain coefficient k. Pour un point tournant, il faut donc rajouter $2\,k$ à S, pour un offset, il faut rajouter $3\,k$.

$$F_{i,i} \quad = \quad S_i + k\,N_i$$

5.2.2 Évaluation des termes non diagonaux

A chaque pas de temps t, on évalue la différence (si elle est positive) $C_{t,i,j}$ entre la séparation standard et la distance entre les segments i et j représentant les positions des avions i et j à t. Ces valeurs sont additionnées pour tous les temps t pour donner $F_{i,j}$, mesure du conflit entre i et j.

$$F_{i,j} \quad = \quad \sum_{t=0}^{total\,time} (C_{t,i,j})$$

A partir de la matrice F, et pour les besoins de l'opérateur de sélection, on est en mesure de calculer une fitness f (ou fonction à optimiser) scalaire de la façon suivante :

$$\exists (i,j),\ F_{i,j} \neq 0 \ \Rightarrow \ f \quad = \quad \frac{1}{2 + \sum_{i \neq j} F_{i,j}}$$

$$\forall (i,j),\ F_{i,j} = 0 \ \Rightarrow \ f \quad = \quad \frac{1}{2} + \frac{1}{1 + \sum_{i} F_{i,i}}$$

On remarquera que cette fitness permet de distinguer les individus pour lesquels il reste des conflits (leurs fitness sont inférieures à 0.5) des individus où il n'y a plus de conflit (leurs fitness sont supérieures à 0.5).

5.2.3 Exemple d'utilisation

Le conflit présenté ici implique 20 avions convergeant simultanément sur un même point. Ce conflit a très peu de solutions car tous les avions sont en conflit deux à deux. En fait, deux solutions optimales existent : dans la première, tous les avions tournent à droite, dans la seconde, tous les avions tournent à gauche. La figure 7 donne le résultat donné par l'algorithme génétique.

[10] Cette notion de surface occupée est d'ailleurs un des critères utilisés par les contrôleurs.

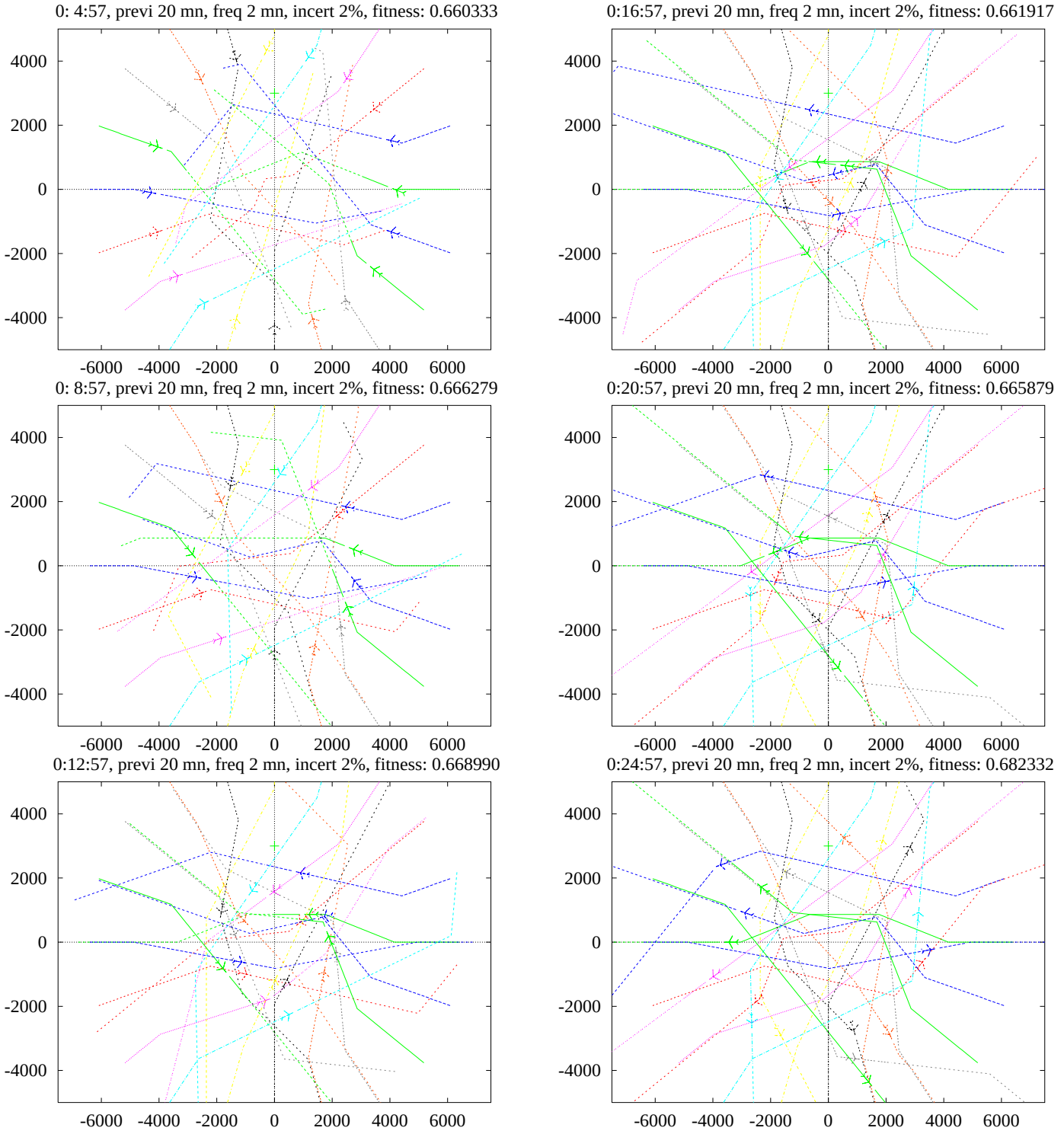

FIG. 7: 20 avions sans conflit

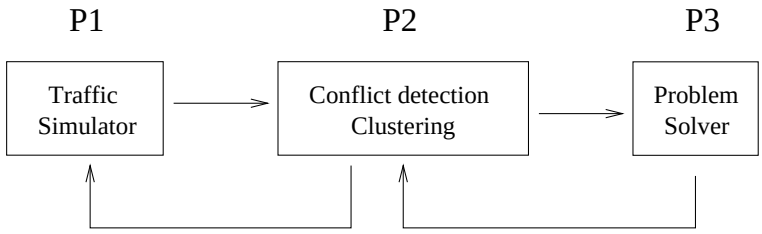

FIG. 8: Architecture générale

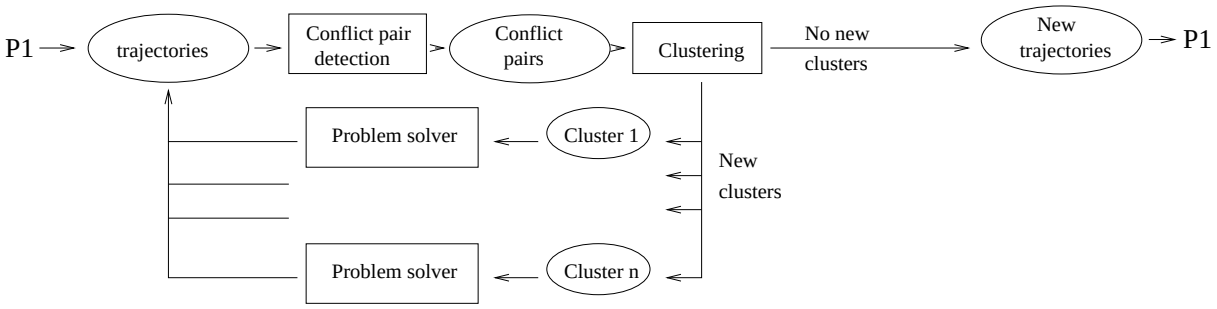

FIG. 9: Architecture détaillée du simulateur de trafic

5.3 Architecture générale du simulateur de trafic

Les tests effectués précédemment montrent que l'algorithme génétique utilisé est très performant sur des cas d'école. Il convenait donc de tester l'algorithme dans un environnement réel.

L'architecture du simulateur de trafic (figure 8) est la suivante :

- Le processus $P1$ est le simulateur de trafic proprement dit, qui gère l'ensemble des vols passant au dessus de la France sur une journée.
- Le processus $P2$ est chargé de détecter les paires d'avions en conflit, de fabriquer les clusters d'avions par fermeture transitive des paires d'avions en conflits. Il vérifie également les trajectoires nouvelles proposées par $P3$.
- Le processus $P3$ est l'algorithme de résolution de conflits décrit dans le chapitre précédent.

L'architecture du simulateur de trafic est détaillée figure 9.

Le processus $P1$ envoie les positions courantes des avions et les plans de vols au processus $P2$. $P2$ construit alors les trajectoires prévues pour les 20 minutes à venir et détecte les paires de conflits. $P2$ prend en compte les incertitudes sur les vitesses. Après avoir détecté les paires d'avions, $P2$ construit les clusters d'avions par fermeture transitive des paires d'avions sur les 20 minutes à venir. Les clusters sont en fait les classes d'équivalence de la relation "est en conflit avec". Tous les clusters sont ensuite résolus par le résolveur de conflits de façon indépendante. Le résolveur de conflits retourne à $P2$ les trajectoires modifiées. $P2$ vérifie alors que les trajectoires modifiées d'un cluster ne génèrent pas de nouveaux conflits avec les avions d'un autre cluster. Si tel est le cas, les ordres de résolutions pour les 5 minutes à venir sont envoyés à $P1$. Si le résolveur a généré de nouveaux conflits entre des avions de deux clusters différents, ces deux clusters sont regroupés, formant un nouveau cluster sur lequel on applique le résolveur de conflits. Ce processus est répété jusqu'à ce que tous les conflits disparaissent. Ce processus converge nécessairement. Dans le pire des cas, tous les avions en conflits dans les 20 prochaines minutes se retrouvent dans le même cluster. Cependant, le processus est d'autant plus efficace qu'il est en présence de nombreux clusters que l'on résoudra en parallèle[11].

[11] Dans la pratique, on utilise un réseau de stations de travail reliées par Ethernet et PVM [GBD+94] pour passer des messages entre les différentes applications

5.4 Résultats sur une journée de trafic réelle

Les expérimentations pratiques sur le simulateur de trafic sont toujours en cours mais certains résultats sont déjà disponibles [Cha95].

Nous décrivons ici un test effectué avec les plans de vols du 12 novembre 1992. 4835 vols ont été enregistrés ce jour-là. Le pourcentage d'incertitude sur les taux de montée et sur la vitesse ont été fixés à 1% et les séparations standards fixées à 6 nm et 1000 pieds. Les routes directes ont été utilisées (les avions ne transitant pas par des balises, mais rejoignant directement leur destination). Seuls les conflits "en route" (au dessus de 6000 pieds) ont été considérés.

Lorsque cette journée est simulée avec un algorithme de détection ne prenant pas en compte l'incertitude, 665 conflits au dessus de 6000 pieds sont détectés.

Lorsque une simulation complète est exécutée, le processus $P2$ détecte 4408 paires d'avions en conflits ne représentant en fait que 908 conflits différents (un certain nombre de conflits durent plus de 5 minutes et sont donc détectés plusieurs fois). 1888 clusters ont étés proposés au résolveur de conflits. Il en a résolu 1541. 260 ont été presque résolus (le conflit prévu n'a finalement pas eu lieu en raison de l'incertitude adoptée). Le résolveur a échoué sur 80 conflits représentant en fait 23 différents conflits (répétés dans le temps). En observant ces conflits, on a pu remarquer qu'ils étaient tous dus à des décollages, atterrissages ou entrées dans l'espace aérien français simultanés (ces conflits apparaissent car des données plan de vol non régulées ont été utilisées). Dans ce cas, le résolveur de conflit ne peut bien évidemment pas séparer les avions. Ne considérer que les conflits au dessus de 6000 pieds ne suffit pas à s'affranchir de ce problème. Un algorithme de pré-régulation des plans de vol est en cours d'élaboration pour résoudre ce problème.

6 Conclusion

En quelques années, les cockpits d'avion, et les systèmes de commande et de contrôle du vol, sont devenus de véritables centres informatiques, et l'automatisation a été poussée au maximum, peut-être d'ailleurs trop vite. On peut en revanche se préoccuper du retard pris en matière de modernisation du contrôle aérien, et les centres de contrôle donnent par comparaison l'impression d'en être encore à l'âge de pierre. Alors que les constructeurs aériens font des efforts démesurés pour réduire les coûts d'opération des appareils de quelques dixièmes de pourcent, les compagnies estiment actuellement que l'influence du contrôle sur leurs coûts d'opérations est de l'ordre de 10%.

Avec l'augmentation régulière de la densité du trafic, il est probable que la seule augmentation du nombre de contrôleurs ne permettra pas d'obtenir un écoulement fluide. Le problème se pose de façon d'autant plus aiguë pour la France, qui se trouve à la croisée des grands flux de trafic Nord-Sud et Est-Ouest passant au dessus de l'Europe. Elle sera donc un des pays les plus touchés par le problème du contrôle en route, et des délais et coûts induits[12].

Il est donc nécessaire de commencer à réfléchir non pas à une automatisation du trafic, mais à la mise en place progressive d'aides à la résolution permettant sur le court terme d'améliorer la capacité de l'espace et sur le long terme de préparer une possible transition vers une automatisation plus poussée.

Références

[BH91] Betts (John T.) et Huffman (William P.). – Trajectory optimization on a parallem processor. *Journal of Guidance, Control and Dynamics*, vol. 14, n° 2, 1991, pp. 431–439.

[Cel90] Celio (Joseph C.). – *Controller perspective of AERA2.* – Rapport technique, MITRE, February 1990. MP-88W00015.

[12] Les problèmes du contrôle américain sont bien différents : il s'agit surtout de problèmes de régulation de flux au niveau des aéroports, qui sont aux Etats-Unis l'élément limitant du système.

[CGT92] Conn (A.R.), Gould (Nick) et Toint (Ph. L.). – *A comprehensive description of LANCELOT.* – Rapport technique, IBM T.J. Watson research center, 1992. Report 91/10.

[Cha95] Chansou (Olivier). – *Résolution Automatisée de Conflits en Route.* – DEA, Ecole Nationale de l'Aviation Civile (ENAC), 1995.

[Cle90] Clements (John C.). – Minimum-time turn trajectories to fly-to points. *Optimal Control Applications and Methods*, vol. 11, 1990, pp. 39–50.

[DA96] Durand (N.) et Alliot (J.M.). – Genetic algorithms for partially separable functions. *In : Submitted to the Fourth International Conference on Parallel Problem Solving from Nature, PPSN96, Berlin.*

[DAAS94] Durand (Nicolas), Alech (Nicolas), Alliot (Jean-Marc) et Schoenauer (Marc). – Genetic algorithms for optimal air traffic conflict resolution. *In : Proceedings of the Second Singapore Conference on Intelligent Systems.* SPICIS.

[DAM93] Dujardin (Patrick), Alliot (Jean-Marc) et Mourlon (Paul-Henri). – Different paths to automation. *In : IFAC'93.*

[DAN94] Durand (Nicolas), Alliot (Jean-Marc) et Noailles (Joseph). – Algorithmes genetiques : un croisement pour les problemes partiellement separables. *In : Proceedings of the Journees Evolution Artificielle Francophones.* EAF.

[DAN96a] Durand (N.), Alliot (J.M.) et Noailles (J.). – Collision avoidance using neural networks learned by genetic algorithms. *In : Ninth International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, Fukuoka.*

[DAN96b] Durand (Nicolas), Alliot (Jean-Marc) et Noailles (Joseph). – Automatic aircraft conflict resolution using genetic algorithms. *In : Proceedings of the Symposium on Applied Computing, Philadelphia.* ACM.

[Dav82] Davis (Randall). – Expert Systems : Where Are We ? And Where Do We Go From Here ? *The AI Magazine*, Printemps 1982.

[DCA96] Durand (N.), Chansou (O.) et Alliot (J.M.). – An optimizing conflict solver for atc. *ATC Quarterly*, 1996.

[Dre79] Dreyfus (Hubert). – *What computers can't do : the limits of artificial intelligence.* – Harper and Row, 1979. ISBN : 0-06-090624-3.

[Dre84] Dreyfus (Hubert). – *Intelligence Artificielle, mythes et limites.* – Flammarion, 1984. La traduction française correspond à la deuxième édition américaine [Dre79].

[Dre87] Dreyfus (Hubert). – Mind over machine. *In : Artificial Intelligence, the case against*, éd. par Born (Rainer). – Croom Helm, 1987. ISBN : 0-312-00439-7.

[Dur96] Durand (Nicolas). – *Optimisation de trajectoires pour la résolution de conflits en route.* – Thèse de Doctorat, INPT, 1996.

[FMT93] Fron (Xavier), Maudry (Bernard) et Tumelin (Jean-Claude). – *Arc 2000 : Automatic Radar Control.* – Rapport technique, Eurocontrol, 1993.

[Fox90] Fox (Mark). – AI and Expert Systems : Myths, Legends and Facts. *IEEE Expert*, Fevrier 1990.

[GBD+94] Geist (A.), Beguelin (A.), Dongarra (J.), Jiang (W.), Manchek (R.) et Sunderam (V.). – *PVM 3 User's Guide and Reference Manual.* – Rapport technique, Oak Ridge National Laboratory, 1994.

[Gol89] Goldberg (D.E). – *Genetic Algorithms in Search, Optimization and Machine Learning.* – Reading MA Addison Wesley, 1989.

[Hol62] Holland (John). – Outline for a logical theory of adaptive systems. *Journal of the Association of Computing Machinery*, vol. 3, 1962.

[K+89] Krella (Fred) et al. – *Arc 2000 scenario (version 4.3).* – Rapport technique, Eurocontrol, April 1989.

[ME88] Mahmoud (Magdi S.) et Eid (Shawki Z.). – Optimization of freeway traffic control problems. *Optimal Control Applications and Methods*, vol. 9, 1988, pp. 37–49.

[MG94] Meckiff (Dr. C.) et Gibbs (Dr. P.). – *PHARE : Highly Interactive Problem Solver.* – Rapport technique, Eurocontrol, 1994.

[MZ88] Mikhailov (L.) et Zaprianov (J.). – Multilevel hierarchical optimization of aircraft trajectories. *In : IFAC/IMACS Symposium.*

[NFC+83] Niedringhaus (W.P.), Frolow (I.), Corbin (J.C.), Gisch (A.H.), Taber (N.J.) et Leiber (F.H.). – *Automated En Route Air Traffic Control Algorithmic Specifications : Flight Plan Conflict Probe.* – Rapport technique, FAA, 1983. DOT/FAA/ES-83/6.

[Nie89a] Niedringhaus (W.P.). – *Automated Planning Function for AERA3 : Manoeuver Option Manager.* – Rapport technique, FAA, 1989. DOT/FAA/DS-89/21.

[Nie89b] Niedringhaus (W.P.). – *A mathematical formulation for planning automated aircraft separation for AERA3.* – Rapport technique, FAA, 1989. DOT/FAA/DS-89/20.

[PA91] Perry (Tekla S.) et Adam (John A.). – Improving the world's largest, most advanced system ! *IEEE Spectrum*, February 1991.

[Pla93] Planchon (Pascal). – Use of advanced technologies in ATM domain. *In : AGARD conference.* – Berlin, 1993.

[Sch90] Schultz (Robert L.). – Three-dimensional trajectory optimization for aircraft. *Journal of Guidance, Control and Dynamics*, vol. 13, n° 6, 1990, pp. 936–943.

[SPSS83] Schuster (E. M.), Petroski (F. R.), Sciambi (R. K.) et Stokrp (M. MC). – *AERA 2 Functional Design and Performance Description.* – Rapport technique, MITRE, September 1983. MtR-83W136.