

Algorithmes génétiques appliqués à la gestion du trafic aérien

Nicolas Durand, Jean-Baptiste Gotteland

▶ **To cite this version:**

Nicolas Durand, Jean-Baptiste Gotteland. Algorithmes génétiques appliqués à la gestion du trafic aérien. Journal sur l'enseignement des sciences et technologies de l'information et des systèmes, 2003, 2 (Hors-Série 1), pp xxxx. 10.1051/bib-j3ea:2003506 . hal-00991623

HAL Id: hal-00991623

https://enac.hal.science/hal-00991623

Submitted on 15 May 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.

J3eA, Journal sur l'enseignement des sciences et technologies de l'information et des systèmes,
Volume 2, Hors-Série 1, 6 (2003)
DOI : http://dx.doi.org/10.1051/bib-j3ea:2003506
© EDP Sciences, 2003

Algorithmes génétiques appliqués à la gestion du trafic aérien

N. Durand et J.-B. Gotteland

Laboratoire d'Optimisation Globale
Centre d'études de la navigation aérienne / École Nationale de l'Aviation Civile

durand@recherche.enac.fr
gottelan@recherche.enac.fr

Algorithmes génétiques appliqués à la gestion du trafic aérien

Nicolas Durand / Jean-Baptiste Gotteland
Laboratoire d'Optimisation Globale
Centre d'études de la navigation aérienne / Ecole Nationale de l'Aviation Civile
durand@recherche.enac.fr / gottelan@recherche.enac.fr

7 mars 2003

La progression quasi-constante du trafic aérien depuis le début de l'aviation commerciale génère aujourd'hui des problèmes de saturation tant sur les plate-formes d'aéroport, que dans les zones d'approches ou dans l'espace aérien supérieur. Si les avions sont aujourd'hui largement optimisés et automatisés, on peut s'étonner que les tâches de contrôle soient restées pour la plupart artisanales, faisant appel à l'expérience humaine plus qu'à la puissance de calcul d'un ordinateur.

Nous présentons dans ce document deux problèmes de gestion du trafic aérien pour lesquels un algorithme génétique permet de proposer des solutions. La première application se situe au niveau du trafic en route, et plus particulièrement du contrôle tactique. La deuxième application s'intéresse à la gestion du trafic sur une plate-forme aéroportuaire.

1 Résolution de conflit en route

Le contrôle du trafic aérien actuel peut être vu comme un système de filtres, chaque filtre ayant des objectifs différents et gèrant des espaces et des horizons temporels distincts. On peut grossièrement distinguer cinq niveaux :

A long terme (plus de 6 mois), le trafic est organisé de façon macroscopique. Sont concernés par exemple les schémas d'orientation de trafic, les mesures du comité des horaires ou encore, les accords inter-centres et les accords avec les militaires qui permettent aux civils d'utiliser leurs zones aériennes pour écouler les pointes de trafic du vendredi après-midi.

A plus court terme, on parle souvent de pré-régulation : elle consiste à organiser une journée de trafic, la veille ou l'avant-veille. A ce stade, on dispose d'une

grosse partie des plans de vols, on connaît la capacité de contrôle que peut offrir chaque centre[1] en fonction de ses effectifs, le débit maximal d'avions pouvant pénétrer dans un secteur[2], appelé capacité du secteur. C'est le rôle de la CFMU[3].

Le jour même, des ajustements sont réalisés en fonction des derniers événements. Le trafic transatlantique, par exemple, peut être pris en compte à ce stade, les avions supplémentaires se voient affecter leurs routes et heures de décollage, on peut également ré-allouer des créneaux horaires non utilisés et tenir compte de la météo du jour. Ce rôle est en général joué par les FMP[4] dans chaque centre.

Le dernier filtre de la chaîne du contrôle aérien est le filtre tactique : il s'agit du contrôle à l'intérieur d'un secteur. Le temps moyen passé par un avion dans un secteur est de l'ordre d'une quinzaine de minutes. La visibilité du contrôleur est un peu supérieure puisqu'il dispose des plans de vol quelques minutes avant l'entrée de l'avion dans le secteur. Le contrôleur assure les tâches de surveillance, de résolution de conflit et de coordination avec les secteurs voisins. Il convient de préciser la définition d'un conflit aérien : deux avions sont dits en conflit lorsque la distance horizontale qui les sépare est inférieure à 5 milles nautiques[5] et leur différence d'altitude est inférieure à 1000 pieds[6]. Les méthodes de résolution de conflits appliquées par les contrôleurs aériens font appel avant tout à leur expérience et surtout pas à leur créativité. Lorsque plusieurs couples d'avions interagissent dans le même conflit, ils commencent par séparer les problèmes pour n'avoir que des conflits élémentaires à résoudre.

Le filtre d'urgence n'est censé intervenir que lorsque le système de contrôle est absent ou a été défaillant : pour le contrôleur, le *filet de sauvegarde* prédit la trajectoire de chaque avion avec un horizon temporel de quelques minutes à l'aide des positions radar passées et d'algorithmes de poursuite et déclenche une alarme en cas de conflit. Il ne propose pas de solution aux conflits détectés. A bord des avions, le TCAS[7] a pour rôle d'éviter une col-

[1] La France est divisée en cinq centres de contrôle (Paris, Reims, Brest, Bordeaux et Aix en Provence) qui gèrent chacun une partie de l'espace aérien.

[2] Chaque centre de contrôle gère entre 15 et 20 secteurs élémentaires qui peuvent être regroupés en fonction de la densité de trafic et des équipes de contrôleurs disponibles.

[3] «Central Flow Management Unit», située à Bruxelles.

[4] «Flow Management Position»

[5] 1 mille nautique équivaut à 1852 mètres

[6] un pied équivaut à 30,48 cm.

[7] «Traffic alert and Collision Avoidance System» : système embarqué à bord de certains avions que les Etats-Unis ont rendu obligatoire pour tous les avions de plus de 30 passagers.

lision présumée. La prédiction temporelle est inférieure à la minute et varie entre 25 et 40 secondes. Il est alors trop tard pour que le contrôleur intervienne puisque l'on estime qu'il lui faut entre 1 et 2 minutes pour analyser une situation, trouver une solution et la communiquer aux avions. Actuellement, le TCAS détecte les avions environnants et donne un avis de résolution au pilote (pour le moment dans le plan vertical). Ce filtre doit résoudre les conflits non prévisibles comme, par exemple, un avion dépassant un niveau de vol donné par le contrôle, ou un accident technique qui dégraderait notablement les performances de l'avion.

L'application que nous proposons dans ce chapitre se situe au niveau du filtre tactique : connaissant les positions des avions à un instant donné et leurs positions futures (avec une précision donnée), quelles sont les manoeuvres à donner à ces avions afin que les trajectoires ne génèrent aucun conflit et que le retard engendré soit minimal.

Un certain nombre d'hypothèses doivent toutefois être précisées.

- Un avion ne peut pas modifier sa vitesse (ou très faiblement), sauf dans sa phase de descente ;
- On ne peut pas considérer qu'un avion vole à vitesse constante, sauf éventuellement lorsqu'il est en croisière et qu'il n'y a pas de vent. De plus en montée et en descente, sa trajectoire n'est pas rectiligne. On ne peut donc pas en donner de description analytique. L'évaluation des positions futures d'un avion requiert l'utilisation d'un simulateur ;
- Les avions sont contraints en taux de virage, ils préfèrent généralement manoeuvrer latéralement que verticalement, sauf dans les phases de montée ou de descente ;
- Bien que les pilotes automatiques soient aujourd'hui largement plus performants que les pilotes humains (dans les situations normales de vol), il ne parait par réaliste pour l'instant d'envisager des trajectoires qui ne soient pas exécutables par un pilote humain.
- L'incertitude sur les taux de montée et de descente est très forte (entre 10% et 50% de la vitesse verticale). En croisière l'incertitude sur le vitesse est plus réduite (voisine de 5%). Latéralement l'incertitude ne croît pas avec le temps, de même qu'un avion en croisière tient en général bien son altitude.

La nécessité d'avoir recours à un simulateur pour calculer les positions futures des avions rend impossibles la recherche de solutions analytiques au problème de résolution de conflits aériens ainsi que l'utilisation de méthodes d'optimisation classiques ayant recours au gradient ou au hessien du critère à optimiser. Toutefois, la principale difficulté tient plus à la complexité du problème lui-même qu'aux contraintes citées précédemment.

Nous introduisons dans la première partie de ce chapitre quelques définitions permettant de comprendre la complexité du problème de résolution de conflits. La deuxième partie est consacrée à un bref historique des algorithmes testés sur ce problème et leurs limites. La partie trois détaille la modélisation du problème. La mise en oeuvre de l'algorithme génétique est développée dans la quatrième partie et des résultats numériques sont présentés dans une cinquième partie.

1.1 Complexité du problème de résolution de conflit

Pour une fenêtre temporelle de prévision de trajectoire donnée, on appelle conflit potentiel entre deux avions tout conflit détecté pendant la durée de la prévision, et ce en tenant compte des incertitudes sur les trajectoires.

La relation «est en conflit avec», ou «est en conflit potentiel avec», définit une relation d'équivalence. Les classes d'équivalence associées à cette relation seront appelées «clusters» d'avions en conflit ou tout simplement «clusters».

Si l'on se restreint au plan horizontal, dans le cas où l'on s'intéresse à un cluster comprenant n avions, on peut se retrouver en présence de $\frac{n\,(n-1)}{2}$ conflits potentiels. On peut montrer [Dur96] que l'ensemble des solutions admissibles contient $2^{\frac{n\,(n-1)}{2}}$ composantes connexes, ce qui suppose, si l'on veut utiliser une méthode d'optimisation locale (déformation continue de trajectoires) qu'il faut autant d'exécutions de l'algorithme de recherche. Ainsi, pour un cluster à 6 avions, cela représente 32768 composantes connexes. Dans la pratique, compte tenu des performances des avions, toutes les composantes connexes n'ont pas besoin d'être explorées. Néanmoins, la présence théorique d'autant d'ensembles disjoints et l'impossibilité de connaître a priori lequel contient la solution optimale rend le problème fortement combinatoire. En relaxant la contrainte de séparation, on reste tout de même en présence d'un problème d'optimisation globale comportant au moins autant d'optima locaux que de composantes connexes.

L'ajoût de la dimension verticale ne permet pas de réduire le caractère combinatoire du problème dans la mesure où l'on ne propose pas à un avion une manoeuvre simultanément dans les deux plans.

1.2 Les méthodes de résolution existantes

1.2.1 Approches opérationnelles

Historiquement, les américains furent les premiers, à la fin des années 80, à s'intéresser au problème de l'automatisation du contrôle aérien avec le projet AERAIII[8] [Nie89] qui n'apporta pas de solution satisfaisante aux problèmes à plus

[8] «Automated En-Route Air Traffic Control»

de deux avions. Le projet européen ARC2000[9] [FMT93] proposa une méthode de déformation continue de tubes 4D pour optimiser la trajectoire du $n + 1^{ieme}$ avion dans un environnement de n trajectoires contraintes. La modélisation ne prenait pas en compte les incertitudes sur les trajectoires et la méthode s'est avérée rapidement inefficace lorsque le trafic était dense, les gros conflits n'étant pas gérés globalement, mais avec une méthode itérative (la trajectoire du premier avion est fixée, puis celle du deuxième, en prenant la trajectoire du premier avion comme contrainte…). Enfin le projet européen FREER[10] [DF98], apparu en 1995 propose de déporter la tâche de résolution de conflit à bord de l'avion. Le problème de coordination entre avions est géré par des règles de priorité, ce qui revient à utiliser une méthode de résolution 1 contre n, inefficace pour des gros clusters [Gra02].

1.2.2 Approches théoriques

Parmi les approches plus théoriques du problème, on peut citer tout d'abord les méthodes de forces répulsives proposées par Zeghal [Zeg94] : les avions sont «attirés » par leur objectif et repoussés par les avions voisins. La méthode semble bien fonctionner dans des espaces peu denses, mais devient chaotique lorsque le trafic se densifie. Par ailleurs, le modèle suppose que le pilotage soit totalement automatisé, ce qui rend difficile la présence d'un pilote dans l'avion. Des approches similaires utilisant des champs de potentiel sont testées par le département d'aéronautique de Berkeley [GT00], mais elles ne permettent pas pour l'instant de résoudre des conflits à plus de trois avions, tout comme les méthodes neuronales testées au LOG(CENA-ENAC)[11] [DAN96] qui n'ont pu être étendues aux cas de conflits complexes. Enfin, parmi les méthodes qui ont tenté une approche globale de la résolution sur des conflits complexes (plus de cinq avions), on peut citer tout d'abord les travaux de Féron [FMF99] : une méthode utilisant la programmation semi-définie est utilisée pour déterminer le sens de résolution de chaque paire d'avions en conflits. Une fois ce sens de résolution déterminé, une méthode d'optimisation convexe sous contraintes convexes permet de calculer les modifications de cap à imposer aux avions. Cette méthode ne permet néanmoins pas de trouver une solution admissible dans tous les cas. Un processus de bruitage ou randomisation permet d'améliorer son taux de réussite sans toutefois rendre la méthode infaillible. Le cadre très simplifié du modèle choisi (vitesses constantes, avions dans un plan horizontal manœuvrant tous au même moment, pas de prise en compte de l'incertitude) ne laisse que peu d'espoir pour des applications concrètes. Enfin,

[9] «Automatic Radar Control for the 21^{th} Century»

[10] «Free-Route Experimental Encounter Resolution»

[11] Laboratoire d'Optimisation Globale du Centre d'Etudes de la Navigation Aérienne et de l'Ecole Nationale de l'Aviation Civile

le LOG a également testé une méthode de type «Branch and Bound» par intervalles [Méd98], permettant de résoudre de façon optimale des petits clusters (4 avions) sans toutefois pouvoir l'étendre à des clusters plus importants, la taille de l'espace de recherche devenant trop importante.

A ce jour, seul un algorithme génétique s'est révélé capable de trouver des solutions dans un temps raisonnable pour des clusters de plus de 6 avions et jusqu'à une trentaine d'avions.

1.3 Modélisation du problème

1.3.1 Prise en compte de l'incertitude

On définit tout d'abord une horizon temporel de détection T_w pendant lequel un simulateur évalue les positions futures des avions. Le simulateur prend en compte les incertitudes sur les vitesses horizontales et verticales des avions comme le montre la figure 1. Le temps est discrétisé (dans la pratique, par pas de 15 secondes). Dans le plan horizontal, l'avion est représenté par un point à l'instant initial. Le point devient un segment dont la longueur augmente au cours du temps. Lorsque l'on change de direction (à $t = 4$), le segment se déforme en suivant le nouveau vecteur vitesse : l'avion est alors représenté par un parallélogramme. Un nouveau changement de cap (à $t = 7$) transforme le parallélogramme en hexagone et plus généralement en un «convexe». Dans le plan vertical, on définit un cylindre dont la hauteur croît avec le temps. Lorsque l'avion atteint son altitude de croisière (à $t = 8$), le haut du cylindre ne change plus d'altitude et le bas du cylindre continue à monter jusqu'à l'altitude de croisière.

1.3.2 Détection de conflits

Pour détecter les conflits potentiels entre avions, il suffit de mesurer à chaque pas de temps la distance horizontale entre les convexes et la distance verticale entre les cylindres représentant les deux avions. Il y a conflit lorsque les normes verticales et horizontales sont simultanément violées.

1.3.3 Modélisation des manoeuvres d'évitement

On définit des manoeuvres dans le plan vertical et horizontal qui respectent les performances des avions et des pilotes. Dans le plan horizontal, on peut donner au pilote un changement de cap de 10, 20 ou 30 degrés à droite ou à gauche de sa route. La manoeuvre débute à un instant t_0 et se termine à un instant t_1.

Dans le plan vertical, les manoeuvres proposées dépendent de la phase de vol dans laquelle se situe l'avion. Ainsi, comme le montre la figure 2, lorsque l'avion

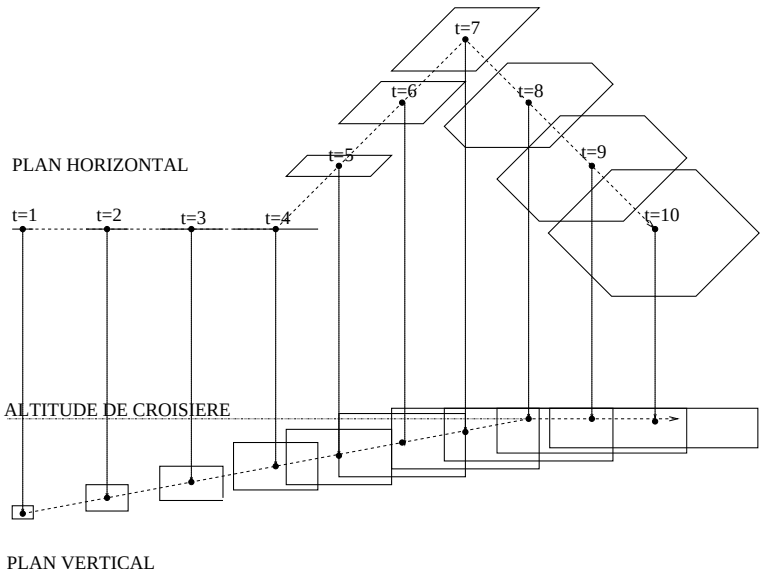


FIG. 1 – Modélisation de l'incertitude.

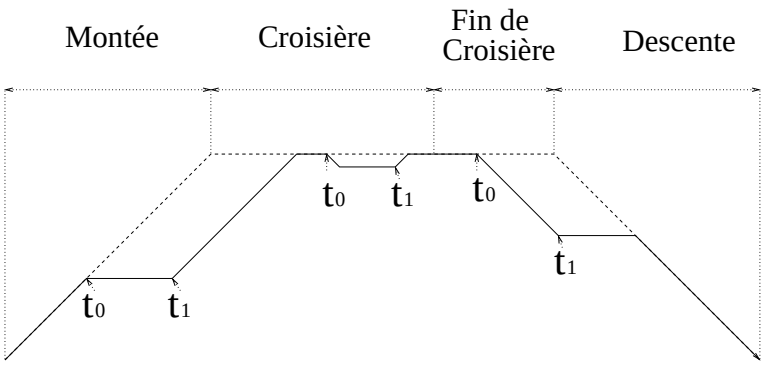


FIG. 2 – Manoeuvres dans le plan vertical.

est en montée, il peut interrompre sa montée à t_0 et la reprendre à t_1. Dans la phase de croisière, il peut descendre d'un niveau de vol (1000 pieds) à t_0 et reprendre son altitude initiale à t_1. Lorsque l'avion est à moins de 50 nautiques de son début de descente, il peut anticiper sa descente à t_0 et faire un palier à t_1 pour rejoindre sa trajectoire de descente. Afin de rendre la manoeuvre exécutable, une seule manoeuvre sera donnée au pilote à la fois. Une nouvelle manoeuvre ne pourra lui être proposée que lorsque la première manoeuvre sera terminée.

Une manoeuvre est donc modélisée par trois variables, la première est une variable discrète indiquant le type de manoeuvre (10, 20, 30, -10, -20, -30 degrés, ou manoeuvre verticale), les deux autres, t_0 et t_1, sont des variables entières indiquant le début et la fin de la manoeuvre. Une résolution d'un cluster à n avions est donc modélisée par $3\,n$ variables.

1.3.4 Gestion en temps réel

La résolution est opérée sur la fenêtre de prévision T_w (fixée entre 10 et 15 minutes) et la situation est reconsidérée toutes les δ minutes (2 ou 3 minutes en pratique). La figure 3 détaille la modélisation en temps réel. Trois périodes sont distinguées dans l'horizon temporel. La première période d'une durée de δ minutes est la période verrouillée. Aucune modification de trajectoire ne peut être opérée pendant cette période. En effet, pendant le temps nécessaire à l'évaluation de la situation, la résolution d'éventuels conflits et la transmission des ordres de manoeuvres, les avions continuent à voler. Il n'est par conséquent pas possible de modifier leurs trajectoires. La période suivante est appelée période définitive, car les ordres de manoeuvres donnés pendant cette période ne pourront pas être modifiés au cours de la prochaine itération. La dernière période est la période de manoeuvres prévues. Ces manoeuvres seront reconsidérées au cours de la prochaine itération. En raison de l'incertitude, certains conflits peuvent disparaître lorsque l'on se rapproche du point de conflit.

1.3.5 Fonctionnement général du simulateur de trafic

Le simulateur de trafic contrôlé CATS[12] est un simulateur arithmétique qui utilise un modèle tabulé pour faire voler les avions. Il prend en compte les plans de vol d'une journée de trafic. Toutes les δ minutes (2 ou 3 minutes en pratique), le simulateur effectue une prévision de trajectoires pour les T_w minutes à venir. Il réalise une détection de conflits par paires, puis construit les clusters d'avions en conflit. Chaque cluster est résolu par un solveur utilisant un algorithme génétique

[12] «Complete Air Traffic Solver »

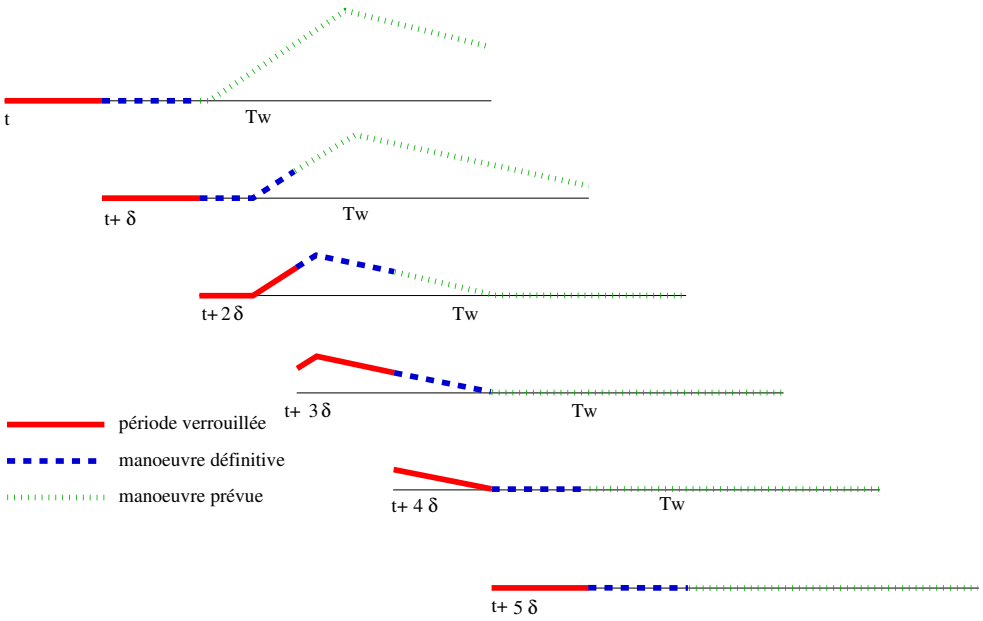


FIG. 3 – Modélisation en temps réel.

qui propose des manoeuvres aux avions. Une nouvelle prévision de trajectoire tenant compte des trajectoires modifiées est alors réalisée afin de détecter des conflits pouvant apparaître en deux avions n'appartenant pas au même cluster. Lorsque deux avions de deux clusters différents sont en conflit, les deux clusters sont réunis et une nouvelle résolution est opérée. Si un avion isolé interfère avec un avion d'un cluster, il est intégré au cluster et une nouvelle résolution est opérée. Le processus est répété tant que des conflits sont observés entre des avions n'appartenant pas au même cluster.

1.4 Mise en oeuvre de l'algorithme génétique

La fonction à optimiser pour chaque cluster tient compte de plusieurs critères différents :

- assurer toutes les séparations entre les avions.
- minimiser les délais imposés aux avions.
- minimiser le nombre de manœuvres nécessaires ainsi que le nombre d'avions déviés.
- minimiser les temps de manœuvres pour que l'avion soit libre aussitôt que possible.

1.4.1 Description générale

L'algorithme génétique utilisé pour réaliser l'optimisation est un algorithme classique tel que décrit par [Gol95].

Une population initiale de $3\,n$ variables est créée de façon aléatoire (la taille de la population est proportionnelle au nombre d'avions en présence avec un maximum de 200 individus). Chaque individu (représentant une configuration de manoeuvres) est évalué. Les meilleurs individus sont ensuite reproduits et sélectionnés en fonction de leur adaptation (la technique de sélection utilisée est le «Stochastic Reminder Without Replacement». Une partie de la population (50%) est ensuite croisée : à partir de deux «parents», deux «enfants» sont créés et remplacent les parents dans la population. Un certain nombre d'individus sont mutés (15%). La mutation consiste généralement à modifier la manoeuvre d'un avion dans le cluster. La distance utilisée pour distinguer deux configurations lors du «sharing» est simple. Deux manoeuvres sont considérées égales si elles sont toutes deux verticales ou horizontales et dans ce dernier cas si elles s'effectuent du même côté. Pour mesurer la distance entre deux configurations on compte le nombre de manoeuvres différentes. L'élitisme est utilisé : à chaque génération, les meilleurs individus de la population sont protégés de façon à ne pas disparaître au cours d'un croisement ou d'une mutation.

Compte tenu des impératifs temporels imposés par une gestion en temps réel du trafic, le critère d'arrêt utilisé consiste à interrompre l'optimisation au bout d'un certain nombre de générations (une vingtaine généralement). Toutefois ce nombre est augmenté tant que l'algorithme ne parvient pas à trouver de solution sans conflit (un maximum de quarante générations est toutefois fixé).

1.4.2 Prise en compte de l'effet horizon

La solveur n'a qu'une vision à court terme des trajectoires des avions. Avec une fonction de coût qui consiste à simplement limiter le retard engendré par une manoeuvre, le solveur est parfois tenté de reporter un conflit au-delà de la fenêtre temporelle sans pour autant le résoudre. Afin de contrer cet «effet horizon », on peut mesurer l'efficacité de résolution d'un conflit et modifier la fonction de coût de l'algorithme de résolution.

Pour toute paire d'avions impliqués dans un cluster :

- Si les avions ne sont pas en conflit, il n'y a pas lieu de pénaliser la fonction de coût.
- Dans le cas contraire, si les trajectoires entre les positions courantes des avions et leurs destinations se croisent, la fonction de coût est pénalisée lorsque les avions ne sont toujours pas croisés en fin de fenêtre : l'algo-

rithme de résolution n'aura alors fait que repousser le conflit au-delà de la fenêtre de résolution.

1.4.3 Fonction à optimiser

Pour chaque configuration, une matrice F de taille $(n \times n)$ permet de stocker les informations suivantes :

- Le terme de la diagonale $F_{i,i}$ mesure l'allongement de la trajectoire de l'avion i. Il est nul si aucune manoeuvre n'est donnée à l'avion i.
- Le terme $F_{i,j}$ avec $i < j$ mesure la violation de séparation entre l'avion i et l'avion j. Il est nul lorsque les deux avions ne sont pas en conflit.
- Le terme $F_{i,j}$ avec $i > j$ mesure l'efficacité de la résolution de conflit entre l'avion i et l'avion j.

La fonction d'évaluation proposée est de la forme :

$$\begin{aligned} \exists(i,j),\, i \neq j,\, F_{i,j} \neq 0 \;\Rightarrow\; F &= \frac{1}{2 + \sum_{i \neq j} F_{i,j}} \\ \forall(i,j),\, i \neq j,\, F_{i,j} = 0 \;\Rightarrow\; F &= \frac{1}{2} + \frac{1}{1 + \sum_{i} F_{i,i}} \end{aligned}$$

Elle garantit qu'une configuration sans conflit est toujours mieux évaluée qu'une configuration où un ou plusieurs conflits subsistent.

L'algorithme génétique rencontre des difficultés pour résoudre les gros clusters.

Nous allons montrer que l'utilisation de la structure partiellement séparable de la fonction d'évaluation permet de définir des opérateurs de croisement et de mutation adaptés au problème.

1.4.4 Utilisation de la séparabilité partielle

Nous nous intéressons dans ce paragraphe à la minimisation d'une fonction F de n variables x_1, x_2, .., x_n, somme de m termes positifs F_i qui ne dépendent chacun que d'un sous-ensemble de variables.

Une telle fonction (que nous dirons partiellement séparable) peut s'exprimer :

$$F(x_1, x_2, .., x_n) = \sum_{i=1}^{m} F_i(x_{j_1}, x_{j_2}, .., x_{j_{n_i}})$$

1.4.5 Opérateur de croisement adapté :

L'idée intuitive est la suivante : pour un problème complètement séparable, le minimum global est obtenu lorsque l'on minimise la fonction sur chacune des variables séparément. La fonction à minimiser peut s'écrire :

$$F(x_1, x_2, .., x_n) = \sum_{i=1}^{n} F_i(x_i)$$

Il suffit de minimiser séparément chacune des fonctions F_i pour obtenir le minimum global de la fonction.

L'opérateur de croisement qui choisit, pour chaque variable x_i, celle des deux parents qui minimise la fonction F_i, permet de construire un individu toujours meilleur (au sens large) que ses deux parents.

On adapte cette stratégie aux fonctions partiellement séparables. Pour créer un individu enfant à partir de deux parents, l'idée est de choisir, pour chaque variable, celle qui minimise la somme des fonctions partielles F_i dans laquelle elle intervient.

On définit pour cela tout d'abord une fonction d'adaptation locale ou ***fitness locale*** $G_k(x_1, x_2, .., x_n)$ associée à chaque variable x_k comme suit :

$$G_k(x_1, x_2, .., x_n) = \sum_{i \in S_k} \frac{F_i(x_{j_1}, x_{j_2}, .., x_{j_{n_i}})}{n_i}$$

où S_k est l'ensemble des indices i tels que x_k est une variable de F_i et n_i est le nombre de variables de F_i.

La fitness locale associée à une variable isole la contribution de cette variable dans la fitness globale.

Le principe de l'opérateur de croisement adapté est le suivant :

Si

$$G_k(parent_1) < G_k(parent_2) - \Delta$$

alors on retient pour l'enfant 1 la variable x_k du parent 1. Sinon, si

$$G_k(parent_1) > G_k(parent_2) + \Delta$$

alors on retient pour l'enfant 1 la variable x_k du parent 2. Enfin, si

$$|G_k(parent_1) - G_k(parent_2)| \leq \Delta$$

alors on choisit aléatoirement la variable x_k du parent 1 ou du parent 2 pour l'enfant 1. On peut également choisir une combinaison linéaire des variables x_k des deux parents lorsque l'on utilise des variables réelles. Si l'on utilise la même stratégie pour l'enfant 2 que pour l'enfant 1, on peut produire deux enfants semblables, surtout si Δ est petit. On peut éviter ce problème en utilisant une nouvelle paire de parents pour fabriquer chaque enfant.

Soit la fonction complètement séparable :

$$F(x_1, x_2, x_3) = x_1 + x_2 + x_3$$

avec x_1, x_2 et x_3 entiers de l'intervalle $[0, 2]$. La fitness locale de la variable k s'exprime : $G_k(x_1, x_2, x_3) = x_k$. Si l'on applique l'opérateur de croisement sur les individus $(1\ ,\ 0\ ,\ 2)$ et $(2\ ,\ 1\ ,\ 0)$ qui ont la même fitness $F = 3$. Avec $\Delta = 0$, l'enfant 1 s'écrit $(1\ ,\ 0\ ,\ 0)$: $F = 1$. Avec $\Delta = 1$, l'enfant 2 peut être $(2\ ,\ 1\ ,\ 0)$, $(2\ ,\ 0\ ,\ 0)$, $(1\ ,\ 1\ ,\ 0)$, ou $(1\ ,\ 0\ ,\ 0)$.

Les fitness des enfants sont toujours meilleures que celles des parents quand $\Delta = 0$, ce qui n'est pas le cas avec un opérateur de croisement classique.

1.4.6 Application au problème de résolution de conflits

Pour le problème de résolution de conflits, la «fitness locale» associée à chaque avion est définie de la façon suivante :

$$F_i = \sum_{j=1}^{n} (F_{i,j})$$

L'opérateur de croisement adapté est décrit par la figure 4. Pour chaque avion, si la fitness locale de l'avion 1 du parent A est nettement inférieure à celle du parent B, alors c'est la manoeuvre du parent A qui sera retenue. Dans le cas contraire (par exemple pour l'avion 3), c'est la manoeuvre du parent B qui sera retenue. Lorsque les fitness locales sont voisines, une combinaison des deux manoeuvres est utilisée.

Un opérateur de mutation adapté est également utilisé (figure 4). Un avion est tiré parmi ceux dont la fitness locale est supérieure à un seuil donné. Il peut par exemple s'agir des avions qui sont encore en conflit.

1.5 Application numérique

1.5.1 Un exemple de conflit complexe

Dans cet exemple issu d'une simulation, à 10h42 (figure 5), 5 avions sont en croisière au niveau de vol 350 (35 000 pieds). 4 conflits sont détectés entre les avions A et B, B et C, C et D, D et E.

L'algorithme génétique utilise une population de 100 individus, une solution sans conflit est obtenue au bout de 5 à 10 générations (sans l'utilisation de l'opérateur de croisement adapté, il en faut une soixantaine)[13]. L'algorithme est arrêté 20

[13] Ces ordres de grandeurs ont été obtenus en répetant une centaine de fois l'algorithme génétique à partir d'une génération aléatoire initiale différente.

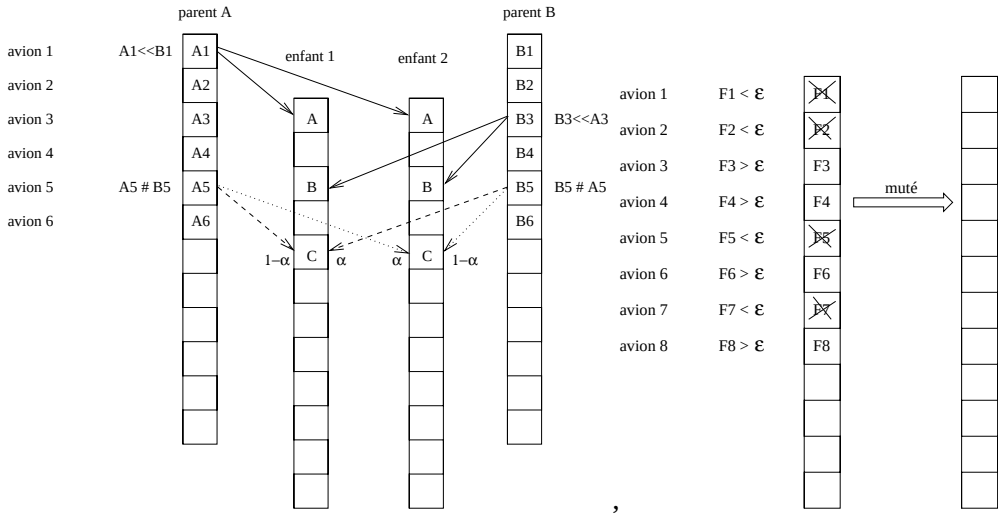


FIG. 4 – Opérateurs de croisement et de mutation adaptés

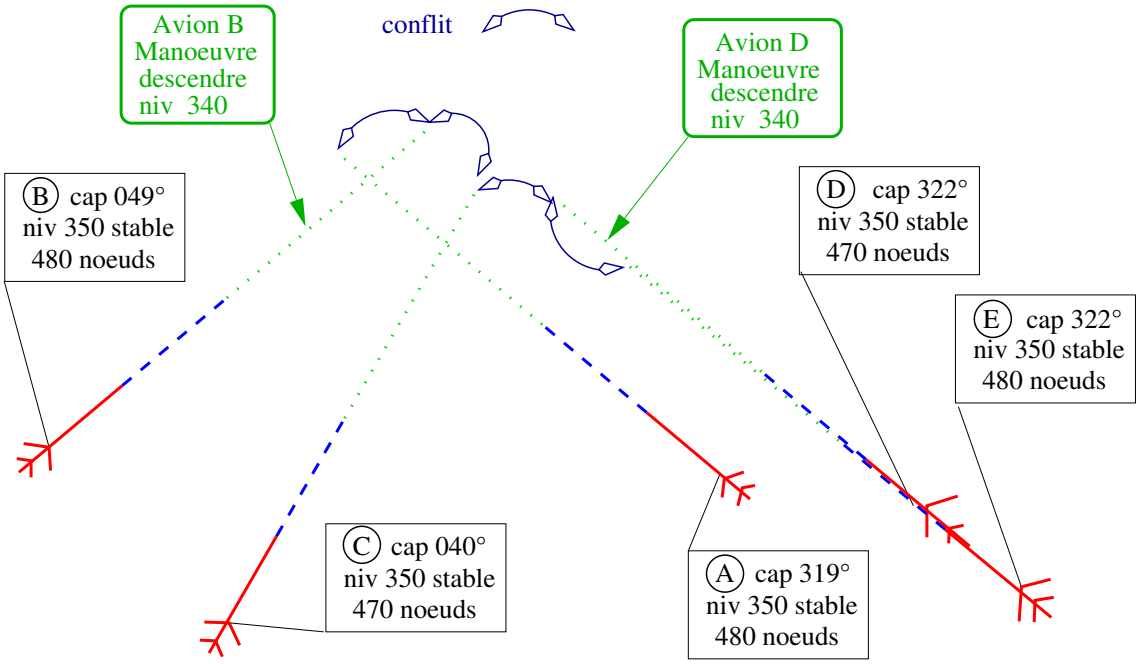


FIG. 5 – Situation à 10h42

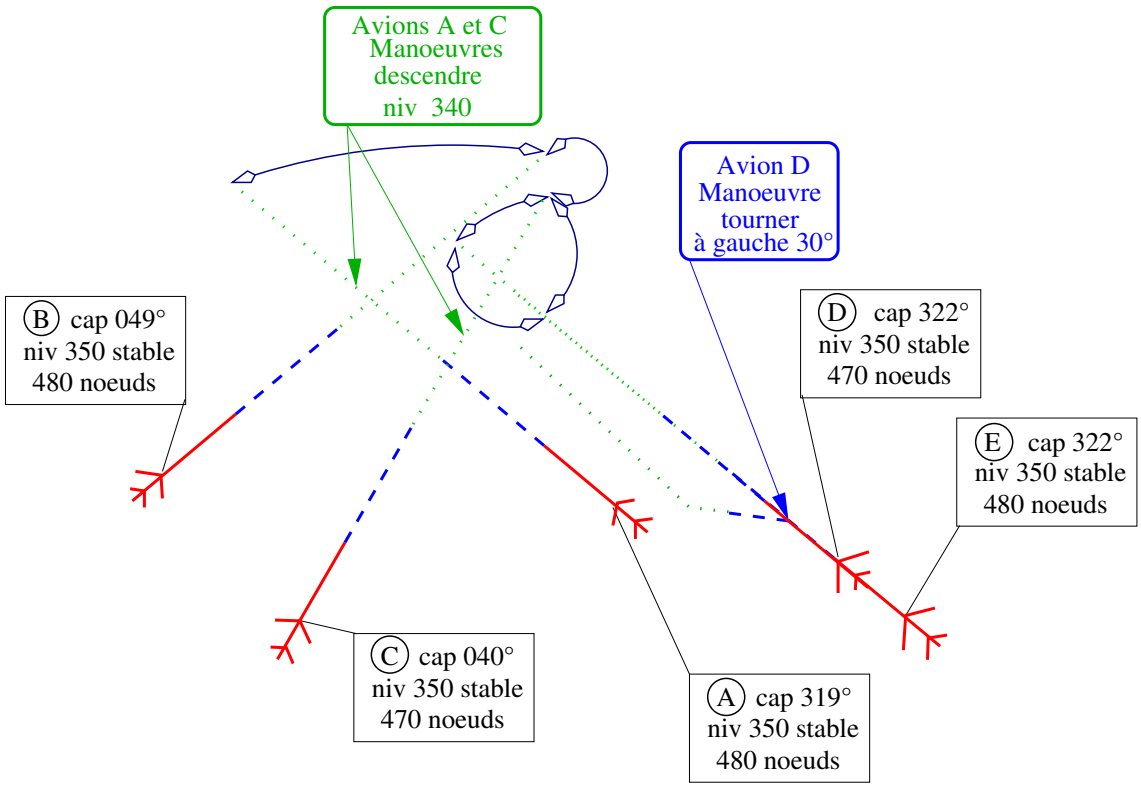


FIG. 6 – Situation à 10h45 (3 minutes plus tard)

générations après l'apparition d'une solution sans conflit (à savoir entre la 25^{ieme} et 30^{ieme} génération.

Le meilleur individu propose de résoudre le conflit avec seulement deux manoeuvres : il propose de faire descendre de 1000 pieds les avions B et D. Toutefois, les manoeuvres débutant dans la période des manoeuvres prévues, elles pourront être remises en cause à la prochaine détection de conflit, 3 minutes plus tard, et ne sont donc pas communiquées au pilote. A 10h45 (figure 6), on retrouve les 5 avions avec 5 conflits (les 4 conflits précédents et un conflit supplémentaire entre l'avion C et l'avion E). Les manoeuvres précédemment calculées ne suffisent plus car elles ne résolvent pas le conflit entre l'avion C et l'avion E. L'algorithme génétique propose désormais 3 manoeuvres dont une qui débute dans la période définitive (virage à gauche pour l'avion D) et deux qui débutent dans la période des manoeuvres prévues. 3 minutes plus tard, en raison de la diminution de l'incertitude, les conflits ont disparu. Finalement, seul l'avion D a subi un virage par la gauche pendant une minute. Cet exemple permet de détailler le fonctionnement du simulateur. Une simulation complète sur une journée de trafic est proposée dans le paragraphe suivant.

1.5.2 Statistiques sur une journée de trafic

Nous présentons dans ce paragraphe les résultats obtenus avec le simulateur sur une journée de trafic français (vendredi 21 mai 1999 : 7540 vols réalisés). Des résultats plus complets peuvent être trouvés dans [Gra02]. La simulation a été réalisée avec trois niveaux d'incertitude :

- 2% dans le plan horizontal et 5% dans le plan vertical
- 5% dans le plan horizontal et 15% dans le plan vertical
- 10% dans le plan horizontal et 30% dans le plan vertical

2140 conflits réels sont observés ce jour-là au dessus de 10000 pieds (niveau 100) lorsque le simulateur ne fait pas de résolution.

Pour chaque niveau d'incertitude le solveur résout tous les conflits. Il est important de noter que le simulateur bruite les trajectoires réelles des avions (en respectant les incertitudes fixées) afin que ceux-ci ne respectent pas exactement leurs trajectoires nominales. Le tableau 1 donne le nombre d'appels du solveur, le nombre d'avions manoeuvrés, la durée moyenne des manoeuvres par avion, la proportion du vol contraint par les manoeuvres et la durée d'exécution de la simulation[14] pour les différents niveaux d'incertitudes. On observe qu'avec une incertitude faible, le nombre de manoeuvres exécutées (2461) est légèrement supérieur au nombre de conflits réels (2140). On peut supposer que l'incertitude provoque alors très peu

[14]Les simulations ont été réalisées avec une douzaine de machines dont la plus puissante est un Pentium IV 2,53 MHz, les résolutions sont exécutées en parallèle.

de manoeuvres inutiles. Avec 10% et 30% d'incertitude, le nombre de manoeuvres est presque trois fois plus important qu'avec 2% et 5% d'incertitude, et le nombre d'appels au solveur est plus de deux fois plus important.

Le tableau 2 montre l'influence de l'incertitude sur la taille des clusters. On observe que l'augmentation de l'incertitude joue un rôle très important sur la taille des clusters à résoudre et donc sur la difficulté des problèmes à résoudre.

Incertitude	Nombre de clusters	Nombre de manoeuvres	Durée moy par avion	Proportion du vol contraint	Durée de la simulation
2% et 5%	8539	2461	34 s	$1,27\%$	26 mn
5% et 15%	12831	3881	78 s	$2,85\%$	35 mn
10% et 30%	19390	6819	236 s	$8,43\%$	55 mn

TAB. 1 – Résultats numériques

taille	2	3	4	5	6	7	8	9	10	11-17	18-37
2%-5%	7205	1021	224	56	23	6	3	1			
5%-15%	9970	1855	586	218	100	42	24	14	11	11	
10%-30%	12859	3326	1317	741	388	245	153	81	77	157	46

TAB. 2 – Influence de l'incertitude sur la taille des clusters.

1.6 Conclusion

L'automatisation du contrôle du trafic aérien n'est certes pas pour demain. Alors que les différentes phases d'un vol peuvent être totalement automatisées, la gestion du trafic reste un problème complexe pour lequel aucune méthode d'optimisation classique n'apporte de réponse satisfaisante. Les algorithmes génétiques permettent de tenir compte des contraintes opérationnelles du problème : obligation de simuler les trajectoires des avions, prise en compte des incertitudes, modélisation des manoeuvres par des variables discrètes. Par ailleurs, la structure partiellement séparable du problème nous a permis de développer des opérateurs de croisement et de mutation très efficaces permettant d'augmenter la taille des problèmes traités jusqu'à une trentaine d'avions. A ce jour, aucune autre méthode testée par notre laboratoire ou d'autres équipes dans le monde n'a permis de résoudre des clusters de cette taille. Il est donc difficile de comparer les performances de l'algorithme génétique avec celles d'un autre algorithme. L'outil développé n'a pour l'instant qu'une vocation de simulation. Un modèle a été adapté récemment afin

de tenir compte de la structure actuelle des routes aériennes. Il s'agit de bien comprendre que cet outil ne contrôle pas du tout de la même façon qu'un contrôleur aérien. L'exemple détaillé à 5 avions est difficilement gérable en temps réel par un opérateur humain qui préfèrera scinder le problème en amont en deux clusters plus petits. Cet outil de simulation permet néanmoins de faire un certain nombre de mesures sur la complexité du trafic et de comparer différents espaces aériens (l'espace européen et l'espace américain). Sa rapidité d'exécution permet d'en faire un outil statistique capable d'absorber un grand nombre de données.

2 Gestion du roulage sur les grands aéroports.

Le développement récent de nombreux hubs commerciaux est à l'origine d'une nouvelle forme de congestion sur les grandes plates-formes aéroportuaires : la plupart des mouvements ont tendance à être programmés aux mêmes heures «stratégiques».

Les nombreux retards occasionnés sont la source d'une incertitude croissante sur les heures de décollage et d'atterrissage. Les retards peuvent atteindre plusieurs dizaines de minutes aux heures de pointe, ce qui est extrêmement pénalisant pour l'ensemble des acteurs du transport aérien. Cette application présente une modélisation du trafic au roulage sur un aéroport. Elle est intégrée dans le simulateur décrit dans la première partie.

Différentes méthodes d'optimisation sont comparées par corrélation entre le délai qu'elles génèrent et le nombre de mouvements impliqués. Elles utilisent des algorithmes génétiques et des algorithmes de parcours de graphe pour trouver le meilleur chemin et/ou les meilleurs points d'attente pour chaque avion, en respectant les normes de séparation au sol, pistes comprises.

2.1 Modélisation

Le problème est de trouver un ensemble optimal de trajectoires admissibles pour le trafic au roulage.

Une trajectoire est définie par une heure de départ (ou d'arrivée), **un chemin** et des **points d'attente** sur ce chemin.

Le caractère optimal de l'ensemble des trajectoires peut avoir différentes définitions, et sera considéré globalement comme le minimum d'une fonction de coût précisée ci-dessous.

Les trajectoires sont admissibles si d'une part les chemins empruntés sont conformes avec les contraintes opérationnelles de l'aéroport (décrites au paragraphe 2.3), et si d'autre part les avions sont séparés (comme détaillé dans les parties suivantes).

2.2 Fonction de coût

La fonction de coût évaluant un ensemble de trajectoires admissibles peut faire intervenir différents facteurs, comme par exemple la durée et/ou la longueur de chaque trajectoire. Une attente peut être jugée plus ou moins coûteuse qu'une déviation. On peut aussi privilégier les départs soumis à un créneau de décollage en rendant leurs retards plus coûteux...

Dans la version actuelle, la fonction de coût f_c retenue prend en compte, pour chaque avion i ($1 \leq i \leq N$), son temps de roulage r_i et le délai d_i dû à sa déviation par rapport au plus court chemin :

$$f_c = \sum_{i=1}^{N} f_{c_i} \quad \text{avec} \quad f_{c_i} = r_i + d_i$$

Les déviations sont ainsi deux fois plus coûteuses que les attentes.

2.3 L'aéroport

Afin d'attribuer à chaque mouvement un ensemble de chemins admissibles, l'aéroport est modélisé par un graphe reliant ses parkings, ses taxiways et ses pistes.

Le coût pour aller d'un taxiway à un autre élément est le temps nécessaire pour parcourir ce taxiway. Ce temps est calculé en fonction du rayon de virage et des procédures spécifiques liées aux pistes (point d'arrêt obligatoire, sorties lentes, normales ou rapides...) et aux parkings (vitesses restreintes, attentes après push-backs...)

Certains taxiways sont des «sens-interdits » utilisables à titre plus ou moins exceptionnel. Leur coût est alors pondéré par un coefficient représentant la tolérance de passage par ce taxiway, conformément à la pratique opérationnelle.

Les différents chemins admissibles pour un mouvement peuvent alors être obtenus avec des algorithmes classiques de parcours de graphe.

L'algorithme du Dijkstra [AMO93] donne tous les meilleurs chemins et les coûts minimums correspondants (i.e. les temps de parcours minimaux) pour aller d'un nœud donné à tous les autres. A partir du résultat du Dijkstra, il est possible de trouver les k_0 meilleurs chemins pour aller d'un point à un autre par Enumération Récursive [JM96]. En réitérant m fois ces deux algorithmes (Dijkstra et Enumération Récursive), et en augmentant à chaque fois le coût des nœuds empruntés par les chemins trouvés, on peut obtenir jusqu'à mk_0 chemins «suffisamment» différents les uns des autres.

Chaque mouvement se voit ainsi attribué k chemins possibles ($k \leq mk_0$) allant du parking à la piste demandée pour un départ, et de la piste d'atterrissage au parking pour une arrivée.

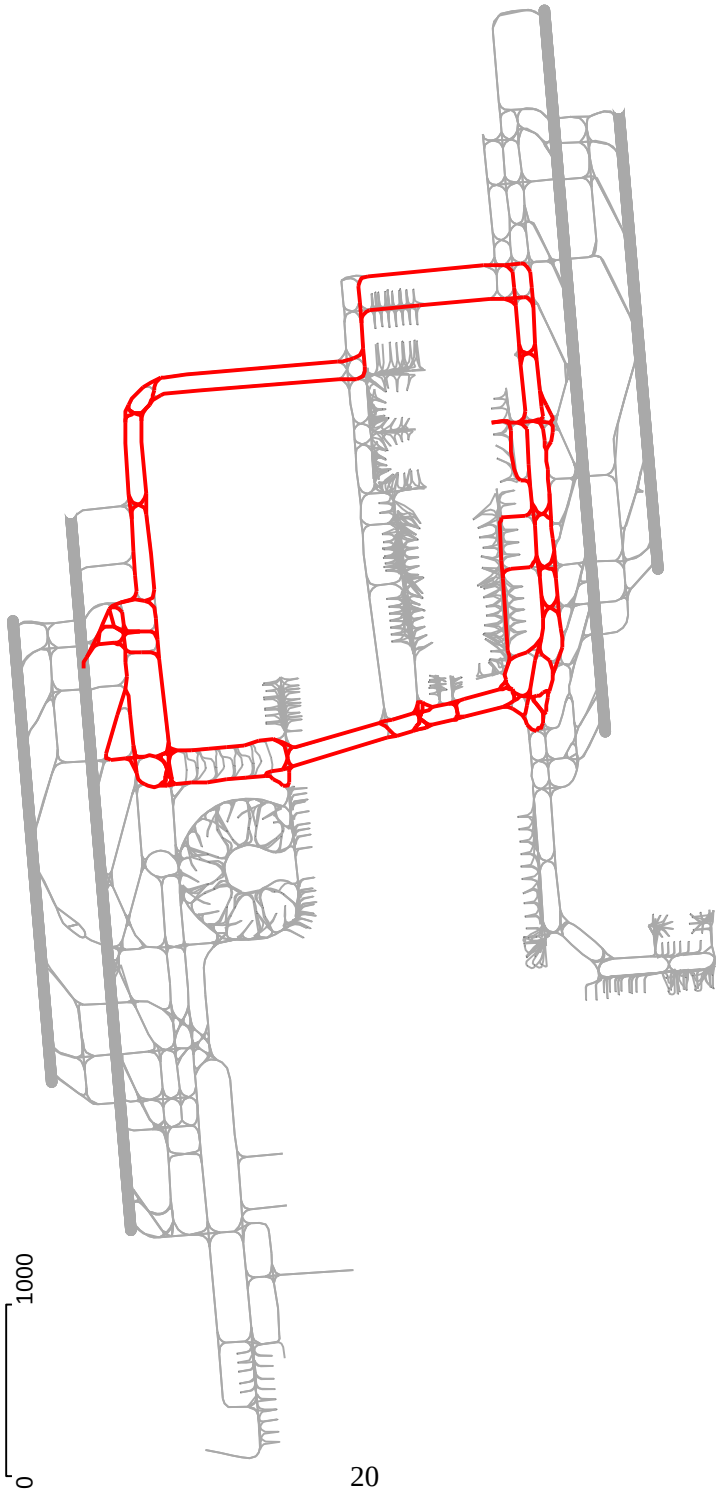


FIG. 7 – Graphe de Roissy

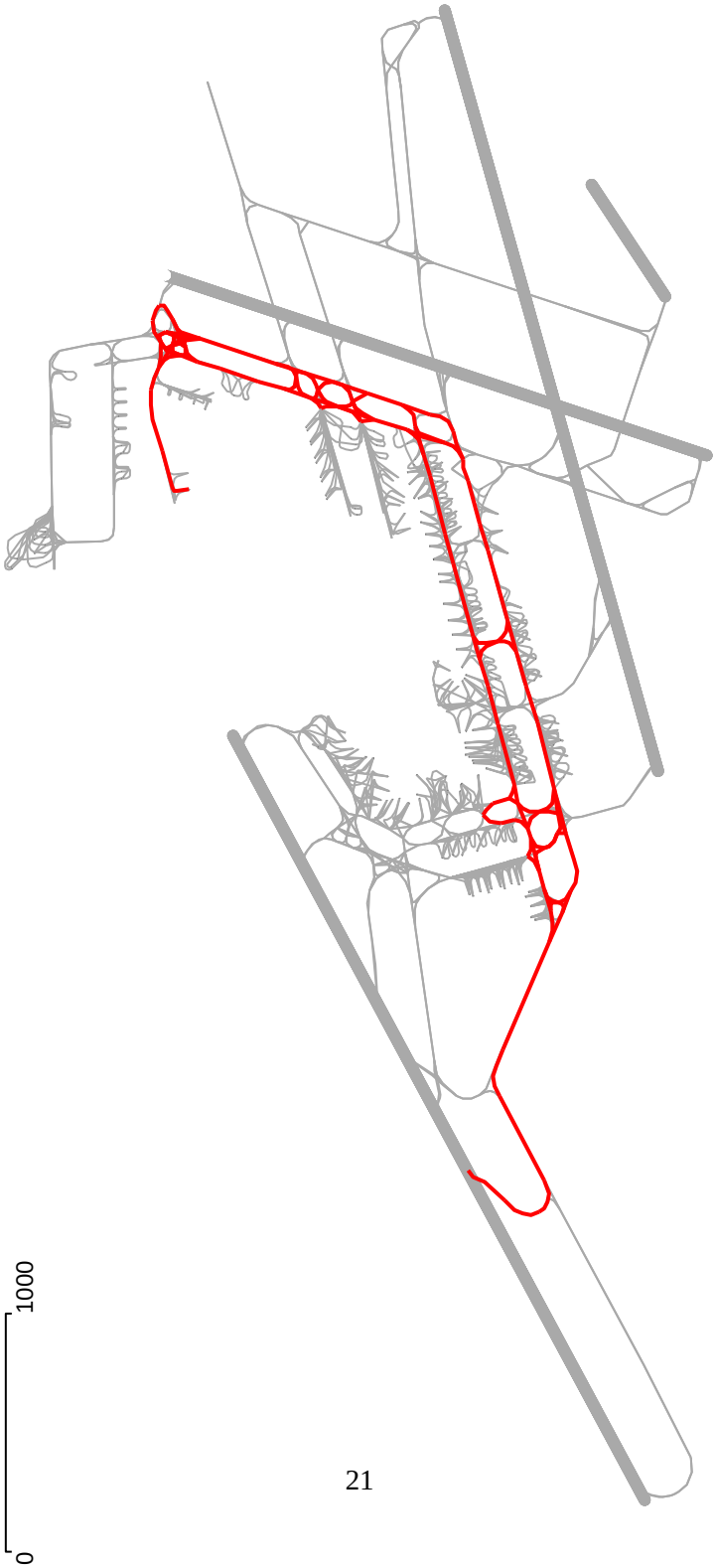


FIG. 8 – Graphe d'Orly

Les figures 7 et 8 représentent les graphes de Roissy et d'Orly et donnent un exemple de la trace des k meilleurs chemins allant d'un point à un autre.

2.4 Le trafic

2.4.1 Les plans de vol

Les mouvements sont décrits par leur plan de vol. Celui-ci fournit notamment l'heure de départ ou d'arrivée, le parking et la piste demandés, et le type avion. Le type avion permet d'appréhender la distance de décollage ou d'atterrissage, donc le(s) point(s) d'entrée ou de sortie de la piste envisageables. Il donne également la catégorie de turbulence de sillage (faible, moyenne ou élevée) utilisée pour les règles de séquencement de piste.

2.4.2 Séparation des avions

Les règles opérationnelles de séparation des avions sont les suivantes :
- la distance entre deux avions en mouvement doit toujours être supérieure à 60 mètres ;
- il ne peut y avoir qu'un seul décollage ou atterrissage à la fois sur chaque piste ;
- après un décollage, une séparation de 1, 2 ou 3 minutes (selon la catégorie de turbulence de sillage) est nécessaire avant le décollage ou l'atterrissage suivant ;
- un avion peut circuler dans l'aire de piste (90 mètres de part et d'autre de la piste) pendant un décollage ou un atterrissage à condition qu'il soit derrière l'avion concerné.

On dira qu'il y a conflit entre deux avions si leurs trajectoires prévues ne respectent pas ces règles.

2.4.3 Incertitudes sur la vitesse

Dans la simulation, les règles de séparation déterminent l'admissibilité des trajectoires, moyennant une incertitude relative sur la vitesse de déplacement des avions. Cette incertitude transforme la position d'un avion en un segment (et non un point). Les règles de séparation sont alors appliquées à tous les points du segment.

Deux particularités doivent cependant être prises en compte :
- Dans le cas où deux avions se suivent, le deuxième pilote est supposé adapter sa vitesse à celle du premier, et seules les positions principales (sans incertitudes) sont considérées.

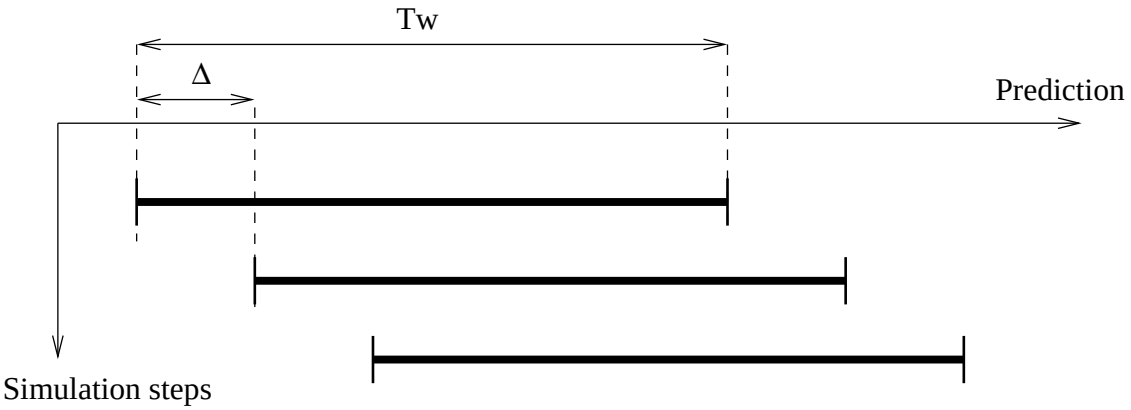


FIG. 9 – Fenêtres glissantes

– Lorsqu'un avion doit attendre, le point et l'heure de fin d'attente sont supposés être respectés, ce qui permet de réduire les positions d'incertitudes de l'avion.

3 Simulation

La simulation s'effectue par fenêtres glissantes sur toute une journée de trafic : à chaque pas de simulation t (toutes les Δ minutes), la prédiction de trafic est effectuée sur un horizon temporel T_w ($T_w > \Delta$). La situation est résolue sur cet horizon, et cette résolution donne la nouvelle situation Δ minutes plus tard (cf. figure 9).

L'horizon de prédiction (et de résolution) étant limité, des «effets d'horizon» néfastes peuvent apparaître :

– Deux avions peuvent être amenés l'un en face de l'autre, créant un blocage définitif pour les situations futures.
– Un avion peut se retrouver bloqué dans l'aire de piste alors qu'une arrivée (non prévue par les résolution précédentes) se présente...

Une analyse des positions en fin d'horizon ($t + T_w$) est donc nécessaire pour assurer la viabilité d'une résolution : toute paire d'avion en position de face à face en fin d'horizon sera considérée comme conflictuelle, et aucun avion ne sera admis sur la piste à l'heure $t + \Delta$ s'il ne l'a pas libérée avant la fin de l'horizon.

Ces nouvelles règles de séparation viennent s'ajouter aux précédentes et seront en vigueur pour toutes les méthodes de résolution présentées dans les parties suivantes.

3.1 BB : Méthode 1 contre n

Dans cette stratégie, les mouvements prévus sur l'horizon sont classés et résolus l'un après l'autre : le problème est donc réduit à celui d'un seul avion devant éviter des avions dont la trajectoire a été précédemment choisie. Ce problème peut être résolu par *Branch & Bound* [HT95] sur le graphe présenté ci-dessous.

3.1.1 Définition du graphe

Sur chacun des chemins possibles de l'avion, le problème peut être résolu rapidement par une recherche meilleure en premier dans l'arbre suivant :

- Un nœud de l'arbre est une position de l'avion à une heure donnée.
- La racine de l'arbre est la position initiale de l'avion au début de l'horizon de prédiction.
- Les feuilles (nœuds terminaux) sont constituées de feuilles solution : les positions non conflictuelles de l'avion en fin d'horizon ou à la fin du chemin ; et de feuilles non solution : toute position conflictuelle de l'avion.
- Chaque nœud (non terminal) a deux fils : le premier correspond au cas où l'avion avance sur son chemin, le deuxième au cas où l'avion attend. Si le premier aboutit à une feuille solution, c'est la meilleure solution depuis ce nœud.

On peut obtenir la meilleure solution pour l'avion en itérant cette recherche meilleur en premier sur chacun de ses chemins.

Cependant, il est possible de connaître pour chaque nœud de l'arbre relatif à l'un des chemins, le retard pris par l'avion jusqu'à ce nœud. Ceci permet de borner la recherche par le minimum des retards pris par l'avion sur les chemins déjà explorés : si le retard dépasse cette borne, la recherche sur ce chemin sera abondonnée.

L'algorithme devient ainsi un *Branch & Bound* avec stratégie d'exploration «meilleur en premier».

3.1.2 Classement des mouvements

Le classement des mouvements est bien sur un point crucial puisque les derniers avions sont extrêmement pénalisés, la plupart des classements pouvant même s'avérer sans solution.

L'analyse du problème permet cependant de dégager deux principales contraintes que doit respecter le classement :

- Les avions prévus à l'atterrissage ne peuvent pas être retardés avant d'être sortis de la piste, et doivent donc disposer de la piste libre : ils doivent être classés avant les décollages sur cette piste.

– Les avions au décollage faisant la queue pour la piste ne peuvent être classés que dans l'ordre donné par la queue, c'est-à-dire en fonction de la distance restante avant la piste.

En dehors de ces contraintes, un classement «réaliste»des mouvements peut être obtenu par comparaison des heures d'activation du plan de vol.

3.2 GA et GA+BB : algorithmes génétiques

Deux stratégies de résolution sont développées et utilisent des algorithmes génétiques classiques [Gol95, Mic92].

La première stratégie recherche un chemin et éventuellement une attente par mouvement. La deuxième recherche un chemin et un ordre de classement par mouvement, en utilisant un algorithme BB (cf. 3.1) pour développer et évaluer les trajectoires correspondantes.

3.2.1 Codage des données

Dans la première stratégie, la trajectoire d'un avion est décrite par 3 paramètres : le numéro n du chemin suivi, la position p d'attente, et l'heure t de fin d'attente (si p est atteinte après l'heure t, l'avion n'attend pas). Les éléments de la population (ou chromosomes) sont donc constitués de $3N$ variables pour un problème à N avions.

Dans la deuxième stratégie, la trajectoire d'un avion est donnée par 2 paramètres : le numéro n du chemin suivi et l'ordre de classement o de l'avion. Les chromosomes sont donc constitués de $2N$ variables.

3.2.2 Fonction d'adaptation

Dans les deux stratégies, les trajectoires décrites par un chromosome peuvent ne pas être admissibles. La fonction d'adaptation (ou fitness) F à maximiser, comprise entre 0 et 1, doit assurer qu'un chromosome représentant une solution admissible est toujours mieux évalué qu'un chromosome décrivant une situation avec conflits.

Pour cela, la fitness d'une solution admissible sera toujours supérieure à $\frac{1}{2}$ et la fitness des éléments contenant des conflits inférieure à $\frac{1}{2}$.

Si n_c est le nombre de conflits et f_c la fonction de coût décrite au 2.2, la fitness sera donnée par :

$$\begin{aligned} &\text{si } n_c > 0, \quad F = \tfrac{1}{1+n_c} \\ &\text{si } n_c = 0, \quad F = \tfrac{1}{2} + \tfrac{1}{1+f_c} \end{aligned}$$

3.2.3 Croisement et Mutation

Le caractère partiellement séparable du problème de résolution de conflits est à nouveau utilisé.

Une fitness locale F_i est calculée pour chaque avion i, en fonction du nombre de conflits n_{c_i} impliquant cet avion et de sa participation f_{c_i} à la fonction de coût (cf. 2.2) :

$$\text{si } n_{c_i} > 0,\ F_i = K n_{c_i} \quad \text{sinon, } F_i = f_{c_i}$$

(K étant une constante telle que $K \gg f_c$)

3.2.4 Sharing

Le problème est combinatoire et présente de nombreux optima locaux. Le sharing décrit dans Yin et Germay [YG93] permet d'empêcher la population de s'homogénéiser trop rapidement sur les optima locaux.

Il nécessite l'introduction d'une distance entre deux éléments A et B de la population, cette distance étant utilisée pour dissocier différents groupes (clusters) dans la population. La distance définie est la suivante :

$$D(A,B) \quad = \quad \frac{\sum_{i=1}^{N} |l_{A_i} l_{B_i}|}{N}$$

l_{A_i} (resp l_{B_i}) étant la longueur du chemin de l'avion i dans le chromosome A (resp B).

3.2.5 Critère d'arrêt

Le critère d'arrêt de l'algorithme génétique est défini par un nombre maximal absolu de 50 générations, et un nombre maximal de 20 générations sans conflit (i.e. à meilleur élément décrivant une solution admissible).

3.2.6 Résolutions par Clusters

Pour diminuer autant que possible la complexité du problème, une fermeture transitive est appliquée aux paires d'avions en conflit dans la situation initiale (sans attente ni déviation). Les ensembles (ou clusters) d'avions ainsi formés sont résolus séparément.

A chaque fois que la résolution séparée de deux clusters provoque des conflits entre leurs avions, les deux clusters sont réunis et une nouvelle optimisation est effectuée.

4 Résultats

4.1 Simulations

Les simulations sont effectuées avec les plans de vol réels d'une journée de forte activité (18/06/1999) de Roissy et d'Orly. Les résultats obtenus pour Roissy et Orly étant similaires, seuls ceux de Roissy sont présentés et commentés dans cet article.

Les trois stratégies (BB, GA et GA+BB) sont comparées avec les paramètres suivants :

- Nombre de chemins par avion : $k = 30$
- Sens uniques : Oui
- Horizon : $T_w = 5mn$
- Pas de résolution : $\Delta = 2mn$
- Incertitude sur les vitesses : $\delta = 10\%$

4.2 Comparaison des stratégies

La figure 10 donne la corrélation entre le nombre de mouvements actifs et le délai généré pour chacune des stratégies de résolution.

Pour les situations à faible trafic (moins de 10 mouvements actifs), la meilleure méthode est GA : elle ne classe pas les avions et peut ainsi trouver des solutions plus proches de l'optimum global.

Lorsque le nombre de mouvements augmente, c'est la méthode GA+BB qui génére le moins de retards : ceci signifie que le classement des avions est beaucoup moins pénalisant à forte densité de trafic, et qu'à partir d'un certain nombre de mouvements, le problème devient trop complexe pour que la méthode GA s'approche suffisamment de l'optimum global.

La méthode BB obtient de moins bons résultats que GA+BB, car elle ne remet pas suffisamment en cause le classement initial des mouvements.

Le classement s'avère donc une méthode efficace pour la résolution de conflits impliquant un grand nombre de mouvements, à condition de revoir régulièrement ce classement pour "l'adapter" aux nouvelles situations.

La figure 11 donne le nombre de mouvements actifs à chaque pas de simulation. La méthode GA+BB parvient souvent à des situations moins denses que celles des autres méthodes, et ce surtout pendant les périodes de pointe.

Ceci fait apparaître un phénomène important concernant les aéroports : une bonne résolution de la situation réduit d'une part les délais dans l'immédiat, et amène d'autre part des meilleures situations futures (impliquant moins de mouvements).

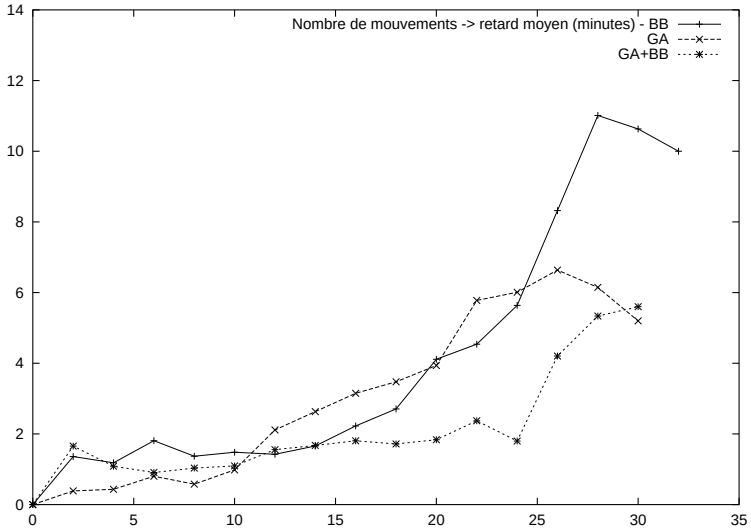


FIG. 10 – Retard moyen en fonction du nombre de mouvements

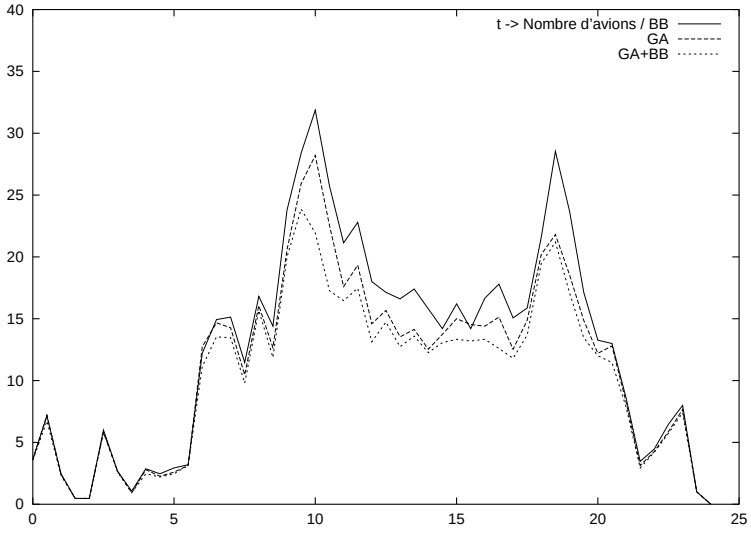


FIG. 11 – Nombre de mouvements actifs sur la journée

4.3 Conclusion

Ces premiers résultats font apparaître que le retard moyen du trafic au roulage sur des aéroports comme Roissy ou Orly peut-être réduit de plusieurs minutes en fonction de la stratégie d'optimisation employée. Ceci montre l'intérêt potentiel du développement d'outils d'aide au contrôle au sol (affectation de chemins et de points d'attente).

Les algorithmes génétiques semblent bien adaptés pour traiter ce problème combinatoire car leurs solutions s'approchent plus facilement de l'optimum global, tandis que des algorithmes déterministes (méthode 1 contre n) se bornent à des optima locaux créant des retards plus importants.

On peut noter également que le modèle à pu être amélioré facilement (nouvelles pistes à Roissy, incertitudes sur les vitesses, prise en compte des sens uniques...) sans modifier pour autant les algorithmes d'optimisation. Les simulations peuvent ainsi servir à évaluer la mise en place de nouvelles structures ou de nouvelles procédures sur un aéroport.

Beaucoup d'améliorations restent à apporter : par exemple, le critère d'optimisation devra prendre en compte les mouvements ayant un créneau de départ impératif (suite à la régulation européenne du trafic), et les contraintes des secteurs d'approche.

Références

[AMO93] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network Flows, Theory, Algorithms and Applications*. Prentice Hall, 1993.

[DAN96] N. Durand, J.M. Alliot, and J. Noailles. 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*, 1996.

[DF98] V.N. Duong and P. Faure. On the applicability of the free-flight mode in european airspace. In *Procceedings of the 2nd USA/Europe Seminar*, 1998.

[Dur96] N. Durand. *Optimisation de trajectoires pour la résolution de conflits aériens en route*. PhD thesis, Institut National Polytechnique de Toulouse, Mai 1996.

[FMF99] E. Frazzoli, Z.H. Mao, and E. Feron. Aircraft conflict resolution via semidefinite programming. *AIAA Journal of Guidance, Control and Dynamics*, 1999.

[FMT93] X. Fron, B. Maudry, and J.C. Tumelin. Arc 2000 : Automatic radar control. Technical report, Eurocontrol, 1993.

[Gol95] D. E. Goldberg. *Algorithmes génétiques, exploration, optimisation et apprentissage automatique.* Paris. Addison Wesley, 1995.

[Gra02] G. Granger. *Détection et résolution de conflits aériens : modélisations et analyse*. PhD thesis, Ecole Polytechnique, 2002.

[GT00] R. Gosh and C. Tomlin. Maneuver design for multiple aircraft conflict resolution. In *American Control Conference*, 2000.

[HT95] R. Horst and H. Tuy. *Global Optimization, Deterministic Approaches*. Springler, 1995.

[JM96] V.M. Jimenez and A. Marzal. Computing the k shortest paths : A new algorithm and an experimental comparison. 1996.

[Méd98] F. Médioni. *Méthodes d'optimisation pour l'évitement aérien : systèmes centralisés, systèmes embarqués*. PhD thesis, Ecole Polytechnique, 1998.

[Mic92] Z Michalewicz. *Genetic algorithms + Data Structures = Evolution Programs*. Springer-verlag, 1992.

[Nie89] W.P. Niedringhaus. A mathematical formulation for planning automated aircraft separation for AERA3. Technical report, FAA, 1989. DOT/FAA/DS-89/20.

[YG93] Xiaodong Yin and Noel Germay. A fast genetic algorithm with sharing scheme using cluster analysis methods in multimodal function optimization. In C.R. Reeves R.F.Albrecht and N.C. Steele, editors, *In proceedings of the Artificial Neural Nets and Genetic Algorithm International Conference, Insbruck Austria*. Springer-Verlag, 1993.

[Zeg94] K. Zeghal. *Vers une théorie de la coordination d'actions, application à la navigation aérienne*. PhD thesis, Universite Paris VI, 1994.