

UNIVERSITE SAAD DAHLAB DE BLIDA

Faculté des Sciences de l'Ingénieur

Département d'Aéronautique

MEMOIRE DE MAGISTER

En Aéronautique

Spécialité : Aéronautique

DETECTION ET RESOLUTION DE CONFLITS AERIENS :

Modélisations et Analyse

Présenté Par

BENAISSA Rachid

Devant le Jury composé de:

M. S. BOUKRAA	Professeur, Université de Blida	Président
M. L. AMTOUT	Chargé de Cours, Université de Blida	Examineur
M. T. REZOUG	Maître de Conférences, Université de Blida	Examineur
M. R. HAOUI	Maître de Conférences, USTH Bab-Ezzouar	Examineur
M. B. BENAOUA-Z.	Maître de Conférences, Université de Blida	Rapporteur

Blida, Septembre 2004

RESUME

L'objet de notre travail consiste à étudier le problème de conflits aériens, considéré de tout temps comme étant un sujet sensible dans le domaine de la sécurité aérienne.

Pour cela, un code de calcul utilisant l'algorithme A* a été élaboré pour la détection et la résolution de conflits aériens, pour un système embarqué.

Le programme est capable de détecter tous les conflits, en prenant comme référence chacun des avions individuellement, sur un rayon fixé, dans une première phase, pour ensuite résoudre les conflits dans une deuxième phase, aussi bien dans le plan horizontal que dans le plan vertical.

Pour la résolution de conflits, le programme génère, pour chaque avion, toutes les trajectoires libres de tout conflit, et en choisit l'optimale.

La simulation de plusieurs cas de figures de conflits aériens a générée des résultats très satisfaisants, vu les temps de réponses et la capacité de résoudre des cas de conflits complexes.

ABSTRACT

The purpose of this work is to study fly conflicts, considered all times to be a sensible subject in the field of fly security.

A software has been developed using A* algorithm in order to detect and resolve fly conflicts.

This software is capable to detect all the conflicts taking as reference each of the airplanes individually on a limited fixed space at first step, and further on resolve the conflict at second step, also in both horizontal and vertical plane.

For conflict resolution, the computational program generate for each airplane all routes free of conflicts, and select th optimal one.

Many cases of airplane conflicts has been tested in simulation and has generated satisfied results, taking into account the duration of time calculation and the capability to resolve more complicated cases of conflicts.

ملخص

الهدف من هذا العمل هو دراسة مشكلة تصادم الطائرات، الذي لا يزال يعتبر كموضوع حساس في مجال أمن الطائرات.

لهذا طورنا برنامج حاسوب لجهاز محمول يستعمل الطريقة *Algorithm A قادر على كشف و حل مشكل تصادم الطائرات.

هذا البرنامج قادر على كشف كل أخطار التصادم قبل وقوعها، أخذاً بعين الإعتبار كل طائرة على حدى في مجال محدد في المرحلة الأولى و حلها في المرحلة الثانية في كل من المستوى الأفقي و العمودي.

لحل مشكل التصادم، البرنامج يبحث لكل طائرة المسارات الخالية من وقوع حالة التصادم و يختار الحل الأفضل.

تجارب رقمية لعدة حالات تصادم أعطت نتائج جد مرضية، أخذاً بعين الإعتبار المدة الزمنية لإستجابة الحاسوب و القدرة على إيجاد الحل لحالات أكثر تعقيداً.

REMERCIEMENTS

Je remercie en premier lieu Dr. BENAOUADA-ZOUAOUI Braham d'avoir accepté d'être mon directeur de thèse, et de m'avoir apporté toute l'aide dont j'ai eue besoin.

Je remercie tout particulièrement M. AMTOUT Lounis pour son soutien et ses conseils qui m'ont permis d'achever ce travail.

Je remercie également MM M. BOUNEDER et S. BERGHEUL pour leur aide précieuse.

Je tiens à remercier vivement MM S. BOUKRAA, T. REZOUG et R. HAOUI d'avoir accepté de faire partie du jury.

Je voudrais enfin remercier toute personne qui a contribué de près ou de loin à la réalisation de ce travail.

LISTE DES ILLUSTRATIONS, GRAPHIQUES ET TABLEAUX

Figure 1.1	Exemple de cluster	11
Figure 1.2	Routes et waypoints	12
Figure 2.1	Modélisation d'une trajectoire d'évitement par point tournant	27
Figure 2.2	schéma de la trajectoire à la troisième étape	30
Figure 2.3	Différentes formes possibles pour l'ensemble P(t)	32
Figure 2.4	Modélisation d'une trajectoire d'évitement par offset	32
Figure 2.5	Trajectoire d'évitement avec un point tournant avec retour direct	40
Figure 2.6	Schéma de l'arbre de recherche de trajectoires	43
Figure 2.7	Principe général des algorithmes génétiques	46
Figure 2.8	Modèles de manœuvre	46
Figure 2.9	La surface occupée par la manœuvre de résolution	48
Figure 2.10	exemple de slicing crossover	49
figure 2.11	exemple de slicing crossover à deux points	50
Figure 3.1	Représentation de la trajectoire d'évitement par point tournant	53
Figure 3.2	Représentation de la trajectoire d'évitement par offset	54
Figure 3.3	Schéma de représentation des champs de détection et de résolution	57
Figure 3.4	Schéma de trajectoire d'évitement d'un angle de 10°	57
Figure 3.5	Schéma de l'organigramme du code de calcul	60
Figure 4.1	représentation graphique du conflit de 2 avions en route directe	62
Figure 4. 2	Représentation graphique pour le cas de 3 avions (<u>A >> B >> C</u>)	64
Figure 4. 3	Représentation graphique pour le cas de 3 avions (<u>A >> C >> B</u>)	65
Figure 4. 4	Représentation graphique pour le cas de 3 avions (<u>B >> A >> C</u>)	66
Figure 4.5	Représentation graphique pour le cas de 3avions ayant des vitesses Différentes et un même niveau de vol	68
Figure 4.6	Représentation graphique pour le cas de 4avions (<u>A>>B>>C>>D</u>)	70
Figure 4.7	Représentation graphique pour le cas de 4avions (<u>D>>B>>C>>A</u>)	71
Figure 4.8	Représentation graphique pour le cas de 8 avions	73
Figure 4.9	Représentation graphique pour le cas de 8 avions avec des vitesses différentes	75
Figure 4.10	Représentation graphique pour le cas de 14 avions	79
Figure 4.11	Représentation graphique pour le cas de 32 avions (pas= 1sec)	83

Figure 4.12 Représentation graphique pour le cas de 32 avions (pas= 10sec)	85
Tableau 2.1 Codage d'un individu à n avions	47
Tableau 4.1 Les données en entrée des deux avions	61
Tableau 4.2 Résultats numériques du 1 ^{er} cas	62
Tableau 4.3 Les données en entrée des trois avions	63
Tableau 4.4 Résultats numériques pour le cas de 3avions (<u>A >> B >> C</u>)	63
Tableau 4.5 Résultats numériques pour le cas de 3avions (<u>A >> C >> B</u>)	65
Tableau 4.6 Résultats numériques pour le cas de 3avions (<u>B >> A >> C</u>)	66
Tableau 4.7 Récapitulatif des trois cas de figures	67
Tableau 4.8. Les données en entrée pour le cas de 3avions avec des vitesses différentes	67
Tableau 4.9 Résultats numériques pour le cas de 3avions avec des vitesses différentes	68
Tableau 4.10 Les données en entrée pour le cas de 4 avions	69
Tableau 4.11 Résultats numériques pour le cas de 4avions (<u>A >> B >> C >> D</u>)	69
Tableau 4.12 Résultats numériques pour le cas de 4avions(<u>D >> B >> C >> A</u>)	71
Tableau 4.13 Récapitulatif des résultats pour le cas de 4 avions	72
Tableau 4.14 Les données en entrée pour le cas de 8 avions	72
Tableau 4.15 Résultats numériques pour le cas de 8 avions	73
Tableau 4.16 Les données en entrée pour le cas de 8 avions avec des vitesses différentes	74
Tableau 4.17 Résultats numériques pour le cas de 8 avions avec des vitesses différentes	74
Tableau 4.18 Les données en entrée pour le cas de 14 avions	76
Tableau 4.19 Résultats numériques pour le cas de 14 avions (1 ^{er} simulation)	77
Tableau 4.20 Résultats numériques pour le cas de 14 avions (2 ^{ème} simulation)	78
Tableau 4.21 Résultats numériques pour le cas de 14avions (pas=10sec)	80
Tableau 4.22 Les données en entrée pour le cas de 32 avions (pas=1sec)	81
Tableau 4.23 Résultats numériques pour le cas de 32 avions (pas=1sec)	82
Tableau 4.24 Résultats numériques pour le cas de 32 avions (pas=10sec)	84

TABLE DES MATIERES

RESUME	1
REMERCIEMENTS	3
TABLE DES MATIERES	4
LISTE DES ILLUSTRATIONS, GRAPHIQUES ET TABLEAUX	4
INTRODUCTION	7
1. GENERALITES ET CONTEXTE DE L'ETUDE	9
1.1 Introduction	9
1.2 Principes généraux	9
1.3 La gestion de l'espace aérien	11
1.4 La gestion des flux	13
1.5 Le contrôle du trafic aérien	14
1.6 Différentes approches pour l'aide à la résolution	17
1.7 Les projets de recherche sur l'automatisation avancée	20
1.8 Impact de l'automatisation sur les performances	24
2. TECHNIQUES D'AUTOMATISATION	25
2.1 Introduction	25
2.2 Génération de trajectoires d'évitement par des méthodes d'intervalles	25
2.3 Génération de trajectoires d'évitement utilisant l'algorithme A*	39
2.4 Génération de trajectoires d'évitement utilisant l'algorithme génétique	44
3. ELABORATION DU CODE DE CALCUL	52
3.1 Environnement de programmation	52
3.2 Hypothèses de travail	52
3.3 Les objectifs	53
3.4 Les manœuvres utilisées	53
3.5 Principe général de fonctionnement	55
3.6 La détection de conflits	56
3.7 Description du code de calcul	59
4. RESULTATS ET COMMENTAIRES	61
CONCLUSION	86
REFERENCES	87

INTRODUCTION

Le trafic aérien connaît une progression très importante depuis plusieurs décennies. Parallèlement, les avions s'équipent de moyens de plus en plus sophistiqués de navigation tels que : FMS¹, GPS² etc. permettant des tenues plus précises de trajectoires. Depuis longtemps, la gestion de toutes les phases d'un vol s'effectue sans toucher aux commandes de l'avion depuis le décollage. En revanche, le contrôle aérien n'a connu que très peu d'évolutions depuis trente ans. Plusieurs éléments permettent d'expliquer la situation actuelle :

Sur le plan économique, l'évolution d'une flotte d'avions est une opération coûteuse. C'est pourquoi de nombreux modèles anciens d'avions ne sont pas équipés de moyens de navigation et de contrôle très évolués et l'hétérogénéité de la flotte contraint les systèmes au sol à devoir continuer à satisfaire les avions les moins « performants », ce qui limite les évolutions.

Le niveau de recherche dans le domaine de la gestion du trafic aérien est encore à un stade primitif. Les premiers projets de recherche dans ce domaine remontent au début des années 80.

La détection et la résolution de conflits aériens sont des problèmes très complexes, que l'on utilise un modèle de résolution centralisé ou un modèle décentralisé, les techniques informatiques classiques n'apportent pas de solutions satisfaisantes.

Le problème de la détection et résolution de conflits aériens a été abordé par certains auteurs utilisant : la programmation semi-définie [3], l'optimisation stochastique et la programmation linéaire [6] et [10], les réseaux de neurones [8], la programmation par contraintes [1], et les algorithmes génétiques [2]et[9]et[11], dans le cadre de stages de DEA ou de thèses de doctorat.

Trois techniques de modélisations pour la détection et la résolution de conflits aériens ont fait l'objet de notre étude, et sont décrites dans le chapitre 2. L'algorithme A* a été utilisé dans la réalisation d'un code de calcul, pour la détection et la résolution de conflis aériens dans le cadre des systèmes embarqués. Le chapitre 3 fait une description

¹ FMS : Flight Management System

² GPS : Global Positionning System

détaillée des différents modules constituant le code de calcul. Le chapitre 4 est consacré à la présentation et aux commentaires des résultats obtenus.

CHAPITRE 1

GENERALITES ET CONTEXTE DE L'ETUDE

1.1 Introduction :

Au cours des dernières années, le nombre de vols contrôlés n'a cessé de croître dans des proportions très importantes. Même si la crise actuelle a entraîné un ralentissement de cette évolution, les prévisions à long terme montrent que ce ralentissement ne devrait être que provisoire [7].

Dans ce contexte la gestion du trafic aérien s'est lentement organisée, pour se diviser aujourd'hui en trois grands modules : la gestion de l'espace aérien, la gestion des flux et le contrôle du trafic [2].

1.2 Principes généraux :

Le but premier du contrôle du trafic aérien est d'assurer la sécurité des aéronefs. Le but second, d'assurer un écoulement aussi optimal que possible des flux de trafic, en particulier en terme de retards.

Nous allons tout d'abord poser quelques définitions indispensables pour la suite de notre exposé :

1.2.1 Route aérienne :

le cheminement d'un avion dans l'espace est une série de segments de droite, reliant des points de report appelés balises. Historiquement, ces balises étaient bien souvent des points équipés de moyens de radionavigation.

1.2.2 Plan de vol :

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

1.2.3 Contrôle en route :

il s'agit du contrôle à l'extérieur des zones entourant les aéroports (dans ces dernières, on parle de *contrôle d'approche*).

1.2.4 Secteurs de contrôle :

l'espace aérien est divisé en secteurs de contrôle. Chaque secteur est confié à un, ou plus souvent deux, contrôleurs, qui ont la charge d'assurer la séparation des aéronefs dans cette portion de l'espace. Le transfert d'un avion d'un secteur à un autre secteur fait l'objet d'une *coordination* entre les contrôleurs en charge de chacun des deux secteurs.

1.2.5 Séparations :

on définit une distance horizontale exprimée en milles nautiques (NM), la *séparation horizontale*, et une distance verticale exprimée en pieds (ft) : la *séparation verticale*. 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 différence de leurs altitudes est supérieure à la séparation verticale.

1.2.6 Conflit élémentaire :

deux avions sont dits en *conflit* lorsqu'ils 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.

1.2.7 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 $t < T$ (où T est l'horizon temporel exploré) et B est en conflit avec C à l'instant $t + \Delta t < T$ alors A , B et C appartiennent au même cluster. Ainsi dans la figure 1.1, 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 .

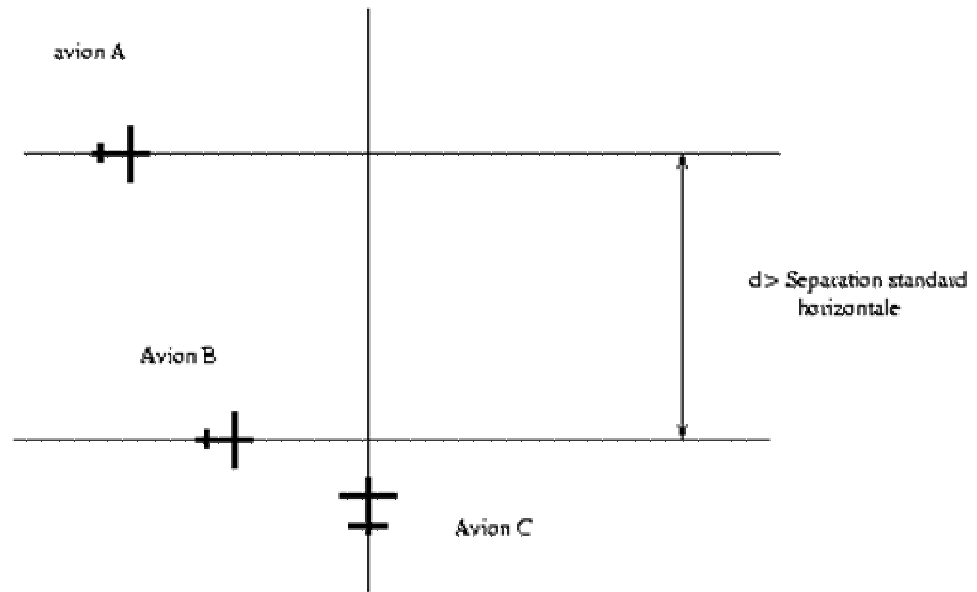


Figure 1.1: Exemple de cluster.

1.3 La gestion de l'espace aérien :

Les avions qui souhaitent traverser l'espace aérien d'un pays doivent obligatoirement respecter un des deux modes de vol suivants :

***Vol VFR** : VFR signifie Visual Flight Rules. Les avions de la catégorie VFR doivent assurer eux mêmes leur séparation les uns par rapport aux autres par les moyens visuels. Il doivent être équipés d'un moyen de radiocommunication pour entrer dans la plupart des zones entourant les aéroports.

***Vol IFR** : IFR signifie Instrument Flight Rules. Les avions de la catégorie IFR doivent être équipés d'un matériel de radionavigation et de vol sans visibilité extérieure. Ils doivent également déposer un plan de vol et doivent obtenir une autorisation de décollage ou de survol.

L'espace a été ensuite divisé volumiquement par l'OACI (Organisation de l'Aviation Civile Internationale) en six catégories, que voici :

- **Classe A** : espaces réservés aux vols IFR, ils se trouvent au dessus du niveau de vol 180 (FL 180) et dans certaines zones denses (aéroports parisiens).

- **Classe B,C et D** : ils se trouvent autour des aéroports. On peut y trouver des aéronefs IFR et VFR. La classe B entoure les aéroports les plus importants, la classe D les moins importants.

Les services rendus dans ces espaces sont différents selon la classe. Dans ces espaces, les vols sont tous connus et en contact radio avec les services de contrôle.

- **Classe E** : il s'agit de l'espace contrôlé situé en dessous du niveau 180 et loin des aéroports. Dans ces espaces les vols ne sont pas tous connus.

- **Classe G** : espace non contrôlé. Les vols ne sont pas connus des services de contrôle.

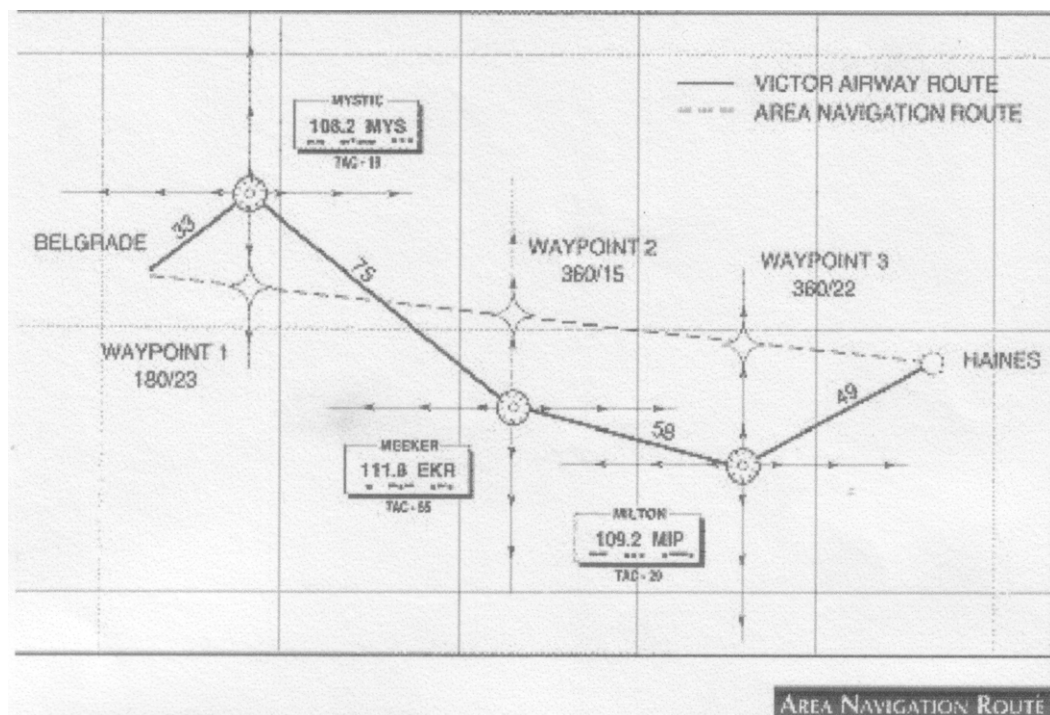


Figure 1.2 Routes et waypoints

Pour aller d'un aéroport à un autre, un aéronef volant dans un espace contrôlé doit suivre une route aérienne. Il s'agit de segments de droites reliant des points de report (waypoints) : la figure 1.2 montre deux exemples de route, les points de report (petits cercles) sont des balises de radionavigation qui existent physiquement au sol et sur lequel les avions se repèrent grâce à leurs moyens de radionavigation (VOR, TACAN, etc). Pour des raisons souvent topographiques, il était impossible de positionner ces balises n'importe où, d'où l'aspect parfois tourmenté des routes.

Le second exemple (appelée Area Navigation Route, ou RNAV) correspond aux nouveaux modèles de route rendu possibles par l'évolution des moyens de radionavigation des avions. Les points de report sont représentés par les losanges et sont purement fictifs, mais les nouveaux systèmes de gestion de vol des avions (Flight Management System, ou FMS) sont aujourd'hui parfaitement capables de les suivre. Ils utilisent en effet les relèvements de plusieurs balises de radionavigation, et peuvent également les combiner avec d'autres moyens de navigation comme les plates-formes inertielle ou la navigation par satellite (GPS). Ces nouveaux modes ont ainsi permis de rendre le réseau de route aussi souple que l'on souhaite ouvrant ainsi des perspectives très importantes pour la redéfinition de l'espace aérien. Il faut cependant souligner qu'une route aérienne ne peut pas passer n'importe où. Il faut d'une part respecter les contraintes d'environnement, mais aussi éviter les zones de l'espace réservées aux militaires.

L'espace aérien contrôlé est en effet découpé en tranches d'espace volumiques, appelées secteurs de contrôle. Chacun de ces secteurs est laissé à la charge d'une équipe de contrôleurs (deux en général) qui sont chargés d'assurer la sécurité du trafic dans l'espace qui leur est confié.

1.4 La gestion des flux :

L'option généralement choisie aujourd'hui est d'imposer aux avions de décoller à une heure donnée. Il est obligatoire de respecter l'heure imposée. On parle de créneaux de décollage car une tolérance de 5 minutes avant l'heure et de 10 minutes après l'heure est admise. Si un avion manque son créneau, il ne peut décoller avant d'en avoir obtenu un autre[4].

1. 5 Le contrôle du trafic aérien :

La définition officielle du contrôle du trafic aérien est la suivante :

Le but du contrôle du trafic aérien est d'assurer la sécurité du trafic et donc d'éviter les abordages entre aéronefs opérant dans le système, puis d'optimiser les flux de trafic[4].

La mission première reste donc la sécurité, la notion de capacité n'intervenant que par la suite.

Le contrôle aérien peut être divisé en trois catégories :

1. 5.1 Le contrôle au sol : Il s'agit du contrôle des aéronefs sur les plates-formes aéroportuaires (roulage, etc.).

1. 5.2 Le contrôle d'approche : Il s'agit du contrôle autour des plates-formes aéroportuaires. Le contrôle d'approche fait appel à des techniques spécifiques, essentiellement orientées vers le séquençage des avions. Les contrôleurs d'approche peuvent mettre les avions en attente dans des « stacks » : une stack est une zone de l'espace aérien où on empile les avions en les faisant tourner. L'avion en bas de la stack est dirigé vers l'aéroport lorsque la piste est prête à le recevoir, libérant ainsi un niveau pour l'avion immédiatement au dessus, etc. Ce type de méthode n'est pas utilisé pour le contrôle en route.

1. 5.3 Le contrôle en route : Il s'agit du contrôle hors zone d'approche. C'est celui qui nous intéresse directement.

Les contrôleurs en route sont regroupés dans des centres spécialisés. Chacun de ces centres prend en charge plusieurs secteurs, et dispose donc de plusieurs positions de contrôle. Sur chacune de ces positions, on trouve deux contrôleurs qui sont en charge du trafic dans ce secteur. Le travail des contrôleurs est différencié. Le contrôleur « organique » s'occupe principalement de la gestion du trafic à moyen terme. Pour ce faire, il dispose de petites bandes de papier (appelées « strip ») posées sur des réglettes rouges ou vertes en fonction du sens de déplacement des vols et qui contiennent toutes les informations fondamentales pour chaque avion appelé à traverser son secteur. Ces informations sont : l'heure d'entrée dans le secteur, la liste des balises de la route de l'avion, le niveau d'entrée, le niveau de vol demandé, etc.

Les informations véhiculées par les strips sont données au contrôleur une dizaine de minutes avant l'arrivée de l'avion dans son secteur. Le contrôleur organique doit également assurer l'interface avec le contrôleur du secteur précédent et celui du secteur suivant, pour assurer une bonne transmission des avions. Cette procédure appelée « coordination » est une des tâches majeures du contrôle. Il assure aussi la gestion globale du flux dans le secteur, et doit s'assurer que la tâche que devra effectuer le second contrôleur, appelé contrôleur radar ou contrôleur tactique, ne sera pas trop lourde.

Le contrôleur « tactique », quant à lui, doit assurer la gestion à court terme du trafic, et en particulier la séparation des aéronefs. Il doit maintenir en permanence une séparation minimale entre les avions. Cette séparation est d'environ 5 NM (1NM =1819m) dans le plan horizontal ou 1000ft (1ft=30cm) dans le plan vertical. Lorsque ces deux normes sont simultanément violées, on dit qu'il y'a perte de séparation, ou conflit. Pour faciliter le travail du contrôleur, les avions volent, lorsqu'ils sont stables, à un niveau de vol « entier » : ainsi, un avion volera par exemple à 31000ft (niveau de vol FL310) mais pas à 31700ft. Les niveaux de vol standards sont ainsi séparés de 1000ft (FL290,FL300,FL310).

Le contrôleur dispose d'une image radar qui lui montre l'ensemble du trafic présent dans son secteur. Aujourd'hui, cette image est une visualisation générée par ordinateur à partir des informations radars collectés puis traités par un système informatique, le STR (Système de Traitement Radar). Chaque avion est représenté par un plot et une étiquette qui contient l'ensemble des informations utiles : cap, indicatif, altitude(montée, stable ou descente) et balise suivante. Il existe encore très peu d'aide à la détection et à la résolution de conflit. Le seul système actif aujourd'hui est le « filet de sauvegarde », un dispositif qui se déclenche lorsque le risque de collision est imminent.

Pour assurer la séparation, le contrôleur tactique dispose de deux catégories de manœuvre : les manœuvres en niveau et les manœuvres en cap. Les manœuvres en niveau se divisent elles-mêmes en trois catégories[5] :

- on peut stabiliser les avions en montée avant de les autoriser à poursuivre vers leur niveau
- on peut anticiper la descente d'avions proches de leur destination
- on s'autorise parfois à monter ou descendre d'un niveau de vol les avions stables en croisières

Cette dernière manœuvre n'est généralement pas très appréciée des pilotes, car toute modification de niveau de vol entraîne la mise en route d'une check-list, et souvent une modification des régimes réacteurs.

Les manœuvres en cap consistent simplement à modifier le cap d'avion à gauche ou à droite, puis à lui faire reprendre sa trajectoire[5].

La charge de travail d'une équipe de contrôleurs travaillant sur un secteur en route peut donc se décomposer de la façon suivante :

La coordination qui consiste à gérer l'entrée et la sortie du secteur des avions ;

La surveillance qui consiste à surveiller le trafic ;

La résolution qui consiste à donner les ordres adaptés pour résoudre les conflits.

1. 6 Différentes approches pour l'aide à la résolution

On peut grossièrement classer les différentes approches de la façon suivante :

1. 6.1 Approche anti-calculatoire :

Cette 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 méthode a l'immense avantage d'être facilement intégrable au système de contrôle actuel, car elle s'emploie à changer le moins possible les modes opératoires existants. Elle est cependant sous les feux de deux critiques : d'une part, les limitations maintenant bien connues propres aux approches dites ``expertes" ou ``cognitives" : 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. Il s'agit donc d'une approche essentiellement court ou moyen terme.

1. 6.2 Automatisation centralisée complète :

Dans un tel cas, un système central gère l'ensemble des avions présents dans l'espace, et leur donne les ordres de contrôle nécessaires à la résolution des conflits. Cette approche a été abordée pour la première fois par le projet AERA-III (nous détaillerons cela dans la suite) aux Etats-Unis, ou plus récemment le projet ARC-2000 du Centre Expérimental Eurocontrol. Le grand avantage de cette méthode est qu'elle permet d'augmenter de façon considérable la capacité de l'espace, la fluidité de l'écoulement du trafic et permet de garantir l'optimalité globale des solutions obtenues. Le principal problème est la transition vers un tel système depuis le système actuel. Il ne peut donc s'agir que d'une approche long ou très long terme.

1. 6.3 Délégation de tâches :

Dans ce cas, on tente de réaliser un partage dynamique des tâches entre l'homme et la machine au sein d'un même secteur de contrôle ; l'ordinateur prendrait en charge certains conflits et laisserait l'homme en charge du trafic restant. Cette méthode pourrait être intéressante à moyen terme : elle devrait permettre de soulager l'opérateur en cas de pointe de trafic tout en maintenant sa vigilance et ses qualifications. L'inconvénient majeur est qu'elle suppose une totale confiance de l'homme envers le calculateur, et envers la répartition de trafic et les résolutions qu'il exécute. Trop peu d'expérimentations ont été menées jusqu'ici pour pouvoir conclure.

1. 6.4 Approche autonome :

Dans ce type d'approche, on suppose que les avions se trouvant dans une certaine zone de l'espace (par exemple au dessus de 32000 ft) utilisent des senseurs et des algorithmes embarqués pour réaliser eux-mêmes la détection et la résolution de conflits. déjà examinée dans le cadre du projet ATLAS, le Centre Expérimental d'Eurocontrol travaille dans le cadre du projet Free-R à la définition d'algorithmes pouvant permettre la réalisation de ce type de système ; les documents définissant les diverses stratégies R&D font d'ailleurs une place importante à ce type de méthodes. Son principal avantage est sa relative compatibilité avec le système actuel : sa mise en place pourrait se faire progressivement par une tactique d'encerclement (en commençant par les espaces trans-océaniques, et les espace supérieurs). D'autre part, elle permettrait de donner aux avions les trajectoires directes origine-destination qu'ils demandent à l'intérieur de ces zones ``libres''.

Notons enfin que ce type d'approche est parfois confondu avec l'ACAS (Airborne Collision Avoidance System) ; ceci mérite une petite précision. L'ACAS, (et particulièrement son implantation actuelle à bord des avions : le TCAS), est un système embarqué conçue pour faire de l'évitement à très court terme (de l'ordre de la minute). Il s'agit avant tout d'un système de dernier secours pour éviter des collisions, et non d'un moyen de contrôle ou de maintien de séparation.

En fait, le dilemme dans lequel nous nous trouvons enfermés est un peu le suivant : les outils capacitifs sont difficilement intégrables dans le système actuel, et les outils facilement intégrables risquent fort d'être peu capacitifs[2].

1.7 Les projets de recherche sur l'automatisation avancée

Des projets de recherche ont eu pour objet des systèmes entièrement automatisés. Nous en présentons ici quelques uns parmi les principaux.

1.7.1 Des systèmes centralisés au sol :

- 1- le premier projet d'automatisation « poussée » est le projet AERA (Automated En Route Air Traffic Control) financé par la FAA développé au début des années 80 à la MITRE Corporation, aux USA [2]. Ce projet devrait se réaliser en trois phases :
 - la première phase AERA1 consistait à prévoir les trajectoires des avions en fonction des intentions des pilotes et à détecter d'éventuelles violations des minima de séparation ou de restriction de flux pour les vols existants ou désignés par un contrôleur. C'était donc avant tout un outil d'information qui ne proposait pas de solution aux opérateurs.
 - la seconde phase AERA2 proposait de nouvelles aides comme une liste de «résolutions recommandées». Néanmoins, la responsabilité de séparer les avions relevait en dernier ressort du contrôleur, AERA2 était un outil d'aide à la résolution.
 - AERA3 devait implanter un système de résolution automatique. La structure d'AERA3 était hiérarchique. On comptait 3 composants :
 - L'ASF (Automated Separation Function) séparait les paires d'avions,
 - Le MOM (Maneuver Option Manager) assurait le respect du contexte global pour l'ASF (qui ne savait gérer que des paires d'avions séparées du reste du trafic) et
 - L'A MPF (Airspace Manager Planning Function) s'assurer que MOM pouvait opérer avec succès en prévenant les trop fortes densités de trafic. AERA3 ne résolvait les conflits que dans le plan horizontal. AERA3 n'a pas abouti et n'a apporté aucune réponse viable aux problèmes techniques de l'automatisation
- 2- Les projet ARC2000 [2] a été développé au centre Expérimental d'Eurocontrol à la fin des années 80. Il s'agit du premier concept qui soit allé, avec un certain succès, jusqu'au stade de l'expérimentation en simulation arithmétique.

Le projet ARC2000, l'automatisation complète du contrôle en route européen, a connu de profonds changements depuis son origine. Néanmoins la modélisation en « tubes 4D » des trajectoires d'avions est restée et a permis de définir des outils d'aide à la décision innovants et efficaces pour le contrôleur.

La trajectoire d'un avion est représentée par une courbe dans \mathbb{R}^4 , les trois premières variables représentant la position dans l'espace de l'avion au temps t , quatrième variable. Les imprécisions de mesures de positions et de tenues de trajectoires 4D transforment, dans un souci de réalisme, cette courbe en « tube » en lui donnant une certaine section. Pour séparer les trajectoires d'avions, il faut donc fabriquer un « tube » pour chaque avion de sorte que les différents tubes soient d'intersection vide. Pour cela, ARC2000 utilise un algorithme d'optimisation locale (de type « gradient ») qui permet, étant donné n tubes d'intersection vide, de construire un tube $n+1$ d'intersection vide avec tous les précédents et minimisant le retard de l'avion $n+1$.

Les hypothèses de départ d'ARC2000 le projetaient bien au-delà de l'an 2000. en effet, la négociation des tubes 4D supposait que les avions soient équipés de FMS-4D, ce qui n'était pas réaliste à court terme. En effet, même aujourd'hui, les meilleurs FMS ne peuvent garantir que la tenue d'un horaire sur un point précis situé (relativement) loin dans le futur, mais en aucun cas une tenue de trajectoire 4D sur l'ensemble du vol.

Le plan stratégique d'ARC2000 consistait à garantir des trajectoires sans conflit pour les 20 minutes à venir. Il fallait en effet trouver un juste milieu entre des trajectoires sans conflit très longtemps à l'avance, et ne pas tenir compte des conflits à l'avance et réajuster continuellement les trajectoires. Le compromis dans ARC2000 consistait à prévoir la trajectoire dans sa totalité avec le moins possible de conflits tout en surveillant les conflits potentiels. 20 ou 30 minutes à l'avance, les conflits étaient éliminés. Avant cette période, seuls les conflits qui étaient fermement diagnostiqués pouvaient entraîner une réorganisation des trajectoires.

ARC2000 s'est ensuite tourné vers des hypothèses plus réalistes. La modélisation de la trajectoire totale sans conflit (de l'origine à la destination) semble avoir été abandonnée pour gestion à 20 ou 30 minutes des groupes de conflits ou clusters. La recherche de solutions optimales a également évolué

puisqu'au principe « dernier arrivé dernier servi », a succédé une série de règles permettant de classer les avions du plus prioritaire au moins prioritaire et les manœuvres de la plus opportune à la moins opportune.

Le projet ARC2000 a été arrêté au milieu des années 90.

- 3- Le projet CATS [2] a été développé au CENA dans les années 90 sur l'idée d'une résolution globale centralisée.

Ce projet a défini la notion de cluster, ou ensemble d'avions interférant dont les trajectoires doivent être calculées simultanément.

Les manœuvres mises en œuvre pour résoudre les conflits restent simples, et les algorithmes prennent en compte les incertitudes sur les vitesses horizontales et verticales des avions. Ces algorithmes tentent de trouver des manœuvres optimales à l'aide d'algorithmes d'optimisation globale.

1.7.2 Des systèmes distribués : avions autonomes et séparations embarquées :

Le projet ATLAS (**A**ir **T**raffic **L**and and **A**irborne **S**ystem), étude de la commission Européenne DGX III, est parmi les premiers projets à avoir envisagé l'hypothèse d'avions autonomes ou hybrides en 1993. Plusieurs méthodes se sont appuyées sur cette hypothèse [2].

- 1- La plus ancienne est celle étudiée par Karim Zeghal à l'ONERA, il introduisait la notion de coordination d'actions grâce à différentes forces qui s'exercent sur les agents, dans notre cas, les avions. Dans la mesure où le processus individuel ne nécessite que des informations locales à l'appareil, il peut fonctionner de façon autonome à partir d'une perception de son environnement. Cependant, ces méthodes échouent relativement souvent lorsque le nombre d'avions est trop important, ou encore savoir donner des manœuvres simples au pilote.
- 2- Le projet FREER (**F**ree-**R**oute **E**xperimental **E**ncounter **R**esolution) est né en 1995 au centre expérimental d'Eurocontrol. L'idée est de généraliser les règles de résolution du vol à vue par des EFR (**E**xtended **F**lying **R**ules)[2].

Les concepteurs de FREER ont complété les règles de l'air de façon à pouvoir d'une part prendre en compte toutes les configurations de conflit à deux avions, et d'autre part définir un ordre total sur l'ensemble des avions dès lors qu'on s'intéresse à trois avions ou plus.

- 3- Le projet FACES [2], développé au CENA , résout le problème de la coordination des avions par un mécanisme de distribution de jetons. Cet algorithme permet de construire un ordre total pour l'ensemble des avions, même si ceux-ci ne sont pas en visibilité directe les uns des autres. Une fois cet ordre défini, on utilise un algorithme classique en robotique (A*) pour résoudre les conflits. FACES a été testé en simulation sur le trafic réel et permet de résoudre l'ensemble des conflits sur la France au dessus du niveau 300.

Tous ces projets ont permis de tirer un bon nombre d'enseignement, et de mieux évaluer de façon précise les bénéfices que l'on peut attendre, en terme de capacité et d'efficacité, d'une évolution conséquente des méthodes de contrôle, et de valider certains algorithmes élémentaires de détection/résolution de conflits d'avions.

Ces études ont également montré que les systèmes centralisés sont plus capacitifs que les systèmes autonomes, mais surtout, ces projets ont permis de dégager plus nettement les deux directions possibles d'évolution de l'ATM (**A**ir **T**raffic **M**anagement) :

- la navigation 4D et
- les systèmes autonomes (qui fera l'objet de notre application dans la suite de ce document)

Pour le futur [12], deux grandes voies semblent devoir s'imposer :

- l'une vers les avions plus autonomes assurant eux-mêmes leur séparation
- l'autre vers des avions gérés de porte à porte le long d'une trajectoire 4D.

1.8 Impact de l'automatisation sur les performances :

L'automatisation de l'ATM peut améliorer les performances dans trois domaines[12] :

- *La sécurité* en éliminant les erreurs humaines ou en diminuant leur impact, par une meilleure information, des filets de sauvegarde et une certaine supervision de l'homme par la machine .
- *La capacité* en utilisant la machine pour assister l'homme .
- *L'efficacité* de la gestion du trafic, en réduisant les pénalités imposées aux avions (routes directes, niveaux optimaux, flexibilité en horaires et ponctualité).

Le futur du système ATM est celui de son automatisation. La gestion de trajectoires 4D et avions autonomes, voire l'hybridation des deux, sont le futur de l'ATM. L'évolution vers les avions autonomes dans les espaces à forte densité est peu probable à cause de la complexité des algorithmes distribués à mettre en œuvre[12].

CHAPITRE 2

TECHNIQUES D'AUTOMATISATION

2.1 Introduction :

Des études réalisées dans ce domaine sont peu nombreuses, celles-ci ont abordés le problème utilisant différentes techniques et méthodes de résolution de conflits aériens, ces techniques sont : les algorithmes stochastiques[10], les réseaux de neurones[8], l'algorithme A* [2]et[4] et les algorithmes génétiques[2],[9],[11]. Les résultats de la simulation obtenus par certains auteurs ont permis de dégager certaines conclusions, quelques indications sur les performances des différentes techniques nous ont conduit à présenter les techniques suivantes :

- Méthodes d'intervalles
- Algorithme A*
- Algorithmes génétiques

2.2 Génération de trajectoires d'évitement par des méthodes d'intervalles :

2.2.1 Introduction :

Le problème de l'évitement aérien est un problème d'optimisation (minimiser les retards) sous contraintes (les avions devant rester séparer). Parmi les méthodes d'optimisation globale existe l'algorithme de type Branch and Bound utilisant l'analyse d'intervalles.

L'analyse d'intervalles a été originalement développée en 1966 par MOORE comme un outil permettant de prendre en compte et contrôler les erreurs d'arrondis dans les calculs numériques effectués sur ordinateurs.

Elle a été ensuite utilisée dans le cadre d'algorithmes d'optimisation global.

L'intérêt d'utiliser les méthodes d'intervalles s'appuie sur les deux caractéristiques suivantes :

- Les méthodes d'intervalles permettent d'utiliser des méthodes d'optimisation globales spécifiques.
- Elles permettent aussi la prise en compte des incertitudes inhérentes au problème à traiter. Ainsi, les méthodes d'intervalles permettront de considérer que les virages des avions dans le cadre de leurs trajectoires d'évitement

n'auront pas lieu à des instants précis, mais prendront place dans des intervalles de temps.

Le fait de travailler avec des intervalles, et non plus avec des points, fait que certaines opérations arithmétiques sont définies sur les intervalles, correspondants à celles qui existent sur \mathbb{R}^n , et des fonctions traduisant en termes d'intervalles les fonctions de \mathbb{R}^n dans \mathbb{R}^n (pour plus de détails voir [4]).

2.2.2 Le principe :

Le principe général d'un algorithme de type Branch and Bound basé sur des méthodes d'intervalles pour la résolution d'un problème d'optimisation sous contraintes est le suivant :

- les trajectoires d'évitement possibles pour un avion éventuellement impliqué dans un conflit sont définies par certaines manœuvres, réalisées à des instants donnés. Dans le cadre de l'utilisation des méthodes d'intervalles, ces instants ne seront pas définis comme des réels, mais comme des intervalles, représentant les intervalles de temps dans lesquels prendront place les manœuvres d'évitement.
- On pourra, au vu des intervalles dans lesquels prendront place les manœuvres d'évitement, savoir si les avions seront ou non séparés durant la durée de l'évitement. Cela sera possible en déterminant, pour un instant t donné, l'ensemble des positions que pourra occuper chaque avion en fonction des intervalles de temps dans lesquels prendront place ses manœuvres, puis en discrétisant le temps sur lequel sera conduit l'évitement, et en vérifiant, pour chaque pas de temps, si les positions possibles pour les avions susceptibles d'être impliqués dans un conflit sont telles que leur séparation sera, ou non, assurée.

La première étape vers l'utilisation de telles méthodes d'intervalles est de calculer les positions susceptibles d'être occupées par un avion à un instant donné quand ses manœuvres ne prennent pas place à des instants ponctuels, mais à l'intérieur d'intervalle de temps.

Deux modélisations de trajectoires d'évitement seront présentées, modélisation par point tournant (section 2.2.3), et modélisation par offset (section 2.2.4).

2.2.3 Modélisation des trajectoires d'évitement par point tournant :

La trajectoire d'évitement par point tournant est une trajectoire d'évitement normalisée, qui comporte quatre étapes :

1. jusqu'à un temps t_0 , l'avion reste sur sa trajectoire d'origine.
2. au temps t_0 , l'avion quitte sa trajectoire, vers la droite ou vers la gauche, mais avec un angle α fixé. Il poursuit ensuite sa trajectoire déviée avec un cap constant jusqu'à un temps t_1 .
3. au temps t_1 , l'avion prend un cap de retour tel que sa trajectoire de retour fasse avec sa trajectoire d'origine un angle égal à l'angle de déviation α pris par l'avion au temps t_0 .

L'utilisation de cette modélisation de l'évitement reste valable si cet angle de retour a une valeur différente de celle de l'angle de déviation.

4. il rejoint sa trajectoire d'origine à un temps t_2 , reprend son cap d'origine et poursuit ensuite sa route sur sa trajectoire d'origine.

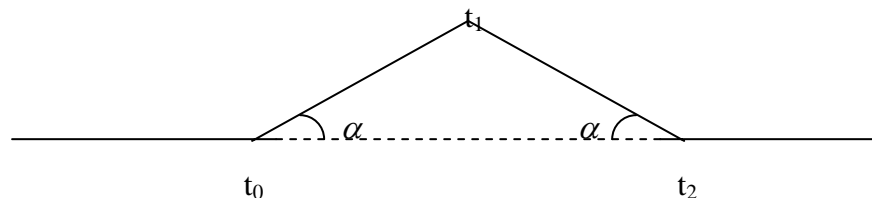


Figure 2.1 Modélisation d'une trajectoire d'évitement par point tournant

Soit $p(t) = (x(t), y(t))$ la position de l'avion à l'instant t , on distingue quatre cas, correspondant aux quatre étapes de la trajectoire d'évitement :

1. $t \leq t_0$. on a alors :

$$\begin{cases} x(t) = v \cdot t \\ y(t) = 0 \end{cases} \quad \text{v étant la vitesse de l'avion} \quad (2.1)$$

l'avion n'a pas encore été dévié

2. $t_0 \leq t \leq t_1$. on a alors :

$$\left\{ \right.$$

$$x(t)=v.(t_0(1-\cos \alpha)+t.\cos \alpha) \quad (2.2)$$

$$y(t)=(t-t_0)\sin \alpha$$

l'avion a été dévié et s'écarte de sa trajectoire d'origine, avec un angle α

3. $t_1 \leq t \leq t_2$. on a alors :

$$\begin{cases} x(t)=v.(t_0(1-\cos \alpha)+t.\cos \alpha) \\ y(t)=(2t_1-t_0-t)\sin \alpha \end{cases} \quad (2.3)$$

l'avion revient sur sa trajectoire d'origine, avec un angle de retour égal à α

4. $t \geq t_2$. on a alors :

$$\begin{cases} x(t)=v.(t-2(t_1-t_0) (1-\cos \alpha)) \\ y(t)=0 \end{cases} \quad (2.4)$$

l'avion est revenu sur sa trajectoire d'origine

Une fois l'avion revenu sur sa trajectoire, le retard entraîné par la trajectoire d'évitement peut se calculer de la manière suivante :

$$\boxed{r(t_0,t_1)=2(t_1-t_0)(1-\cos \alpha)} \quad (2.5)$$

En introduisant des intervalles, on considère que l'on a $t_0 \in T_0$ et $t_1 \in T_1$ où T_0 et T_1 sont deux intervalles de $I(\mathbb{R})$ ($T_0 = [t_{0\min}, t_{0\max}]$ et $T_1 = [t_{1\min}, t_{1\max}]$), les équations 1,2,3 et 4 deviennent :

1^{ère} étape : $t \leq t_{0\max}$

$$\begin{cases} x(t)=v.t \\ y(t)=0 \end{cases}$$

ce point sera noté B1(t). on a donc :

$$\begin{cases} P_1(t) = B1(t) & \text{si } t \leq t_{0\max} \\ P_1(t)=0 & \text{sinon} \end{cases}$$

2^{ème} étape : $t_{0\min} \leq t \leq t_{1\max}$

$$\begin{cases} x(t)= v.(t_0(1-\cos \alpha)+t.\cos \alpha) \\ y(t)= v.(t-t_0).\sin \alpha \end{cases} \quad \text{avec} \quad \begin{cases} t_{0\min} \leq t_0 \leq t_{1\max} \\ t_{0\min} \leq t_0 \leq t_{0\max} \\ t_0 \leq t \end{cases}$$

On note $t'_{0\max} = \min(t, t_{0\max})$, l'ensemble des positions possibles de l'avion correspondant à la deuxième étape de l'évitement (noté $P_2(t)$) est alors un segment de droite dont les extrémités sont :

$$\begin{aligned}
 A2(t) & \begin{cases} x_{A2}(t) = v \cdot (t_{0\min}(1 - \cos \alpha) + t \cdot \cos \alpha) \\ y_{A2}(t) = v \cdot (t - t_{0\min}) \cdot \sin \alpha \end{cases} \\
 B2(t) & \begin{cases} x_{B2}(t) = v \cdot (t'_{0\max}(1 - \cos \alpha) + t \cdot \cos \alpha) \\ y_{B2}(t) = v \cdot (t - t'_{0\max}) \cdot \sin \alpha \end{cases} \\
 \rightarrow P_2(t) & \begin{cases} P_2(t) = [A2(t), B2(t)] & \text{si } t_{0\min} \leq t \leq t_{1\max} \\ P_2(t) = 0 & \text{sinon} \end{cases}
 \end{aligned}$$

3^{ème} étape :

$$\begin{cases} x(t) = v \cdot (t_0(1 - \cos \alpha) + t \cdot \cos \alpha) \\ y(t) = v \cdot (2t_1 - t_0 - t) \cdot \sin \alpha \end{cases} \begin{cases} t_{0\min} \leq t_0 \leq t_{0\max} \\ t_{1\min} \leq t_1 \leq t_{1\max} \\ t_0 \leq t_1 \leq t \\ t \leq 2t_1 - t_0 \end{cases}$$

On note $t'_{1\max} = \min(t_{1\max}, t)$, on obtient :

$$\begin{cases} t_{0\min} \leq t_0 \leq t'_{0\max} \\ t_{1\min} \leq t_1 \leq t'_{1\max} \\ t \leq 2t_1 - t_0 \end{cases}$$

L'ensemble des positions possibles de l'avion correspondant à la troisième étape est donc un parallélogramme tronqué, plus précisément l'intersection du parallélogramme et du demi plan d'équation $y=0$.

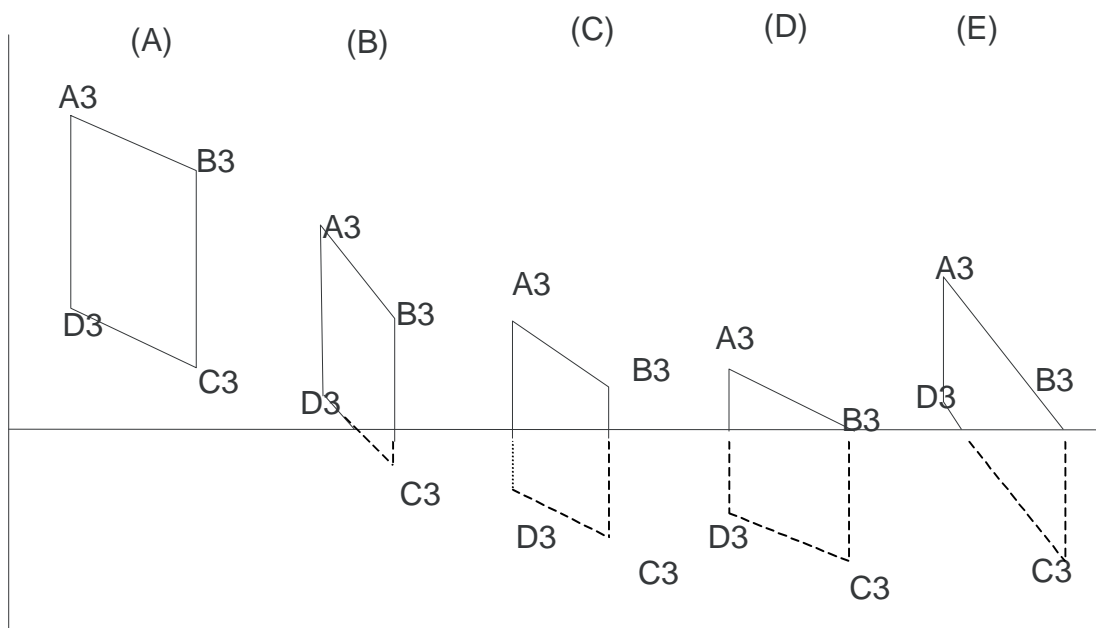
Les coordonnées des sommets du parallélogramme sont :

$$A3(t) \begin{cases} x_{A3}(t) = v \cdot (t_{0\min}(1 - \cos \alpha) + t \cdot \cos \alpha) \\ y_{A3}(t) = v \cdot (2t'_{1\max} - t_{0\min} - t) \cdot \sin \alpha \end{cases}$$

$$B3(t) \begin{cases} x_{B3}(t) = v \cdot (t'_{0max}(1 - \cos \alpha) + t \cdot \cos \alpha) \\ y_{B3}(t) = v \cdot (2t'_{1max} - t'_{0max} - t) \cdot \sin \alpha \end{cases}$$

$$C3(t) \begin{cases} x_{C3}(t) = v \cdot (t'_{0max}(1 - \cos \alpha) + t \cdot \cos \alpha) \\ y_{C3}(t) = v \cdot (2t'_{1min} - t'_{0max} - t) \cdot \sin \alpha \end{cases}$$

$$D3(t) \begin{cases} x_{D3}(t) = v \cdot (t_{0min}(1 - \cos \alpha) + t \cdot \cos \alpha) \\ y_{D3}(t) = v \cdot (2t'_{1min} - t_{0min} - t) \cdot \sin \alpha \end{cases}$$



- Le point A3 correspond au cas où

L'avion a quitté sa trajectoire le plus tôt possible au temps $t_{0\min}$ et a amorcé son retour le plus tard $t_{1\max}$

- Le point B3 correspond au cas où

L'avion a quitté sa trajectoire le plus tard possible au temps $t_{0\max}$ et a amorcé son retour le plus tard $t_{1\max}$

- Le point C3 correspond au cas où

L'avion a quitté sa trajectoire le plus tard possible au temps $t_{0\max}$ et a amorcé son retour le plus tôt $t_{1\min}$

- Le point D3 correspond au cas où

L'avion a quitté sa trajectoire le plus tôt possible au temps $t_{0\min}$ et a amorcé son retour le plus tôt $t_{1\min}$

4^{ème} et dernière étape :

$$\left\{ \begin{array}{l} x(t) = v \cdot (t - 2(t_1 - t_0)(1 - \cos \alpha)) \\ y(t) = 0 \end{array} \right. \quad \begin{array}{l} t_{0\min} \leq t_0 \leq t_{0\max} \\ t_{1\min} \leq t_1 \leq t_{1\max} \\ t_0 \leq t_1 \leq t \\ t \geq 2t_1 - t_0 \end{array}$$

Il s'agit d'un segment de droite noté [EF] dont les coordonnées sont :

$$E(t) \quad \left\{ \begin{array}{l} x_E(t) = v(t - 2)(t_{1\max} - t_{0\min})(1 - \cos \alpha) \\ y_E(t) = 0 \end{array} \right.$$

$$F(t) \quad \left\{ \begin{array}{l} x_F(t) = v \cdot t \\ y_F(t) = 0 \end{array} \right.$$

L'ensemble $P(t)$ peut se schématiser comme suit :

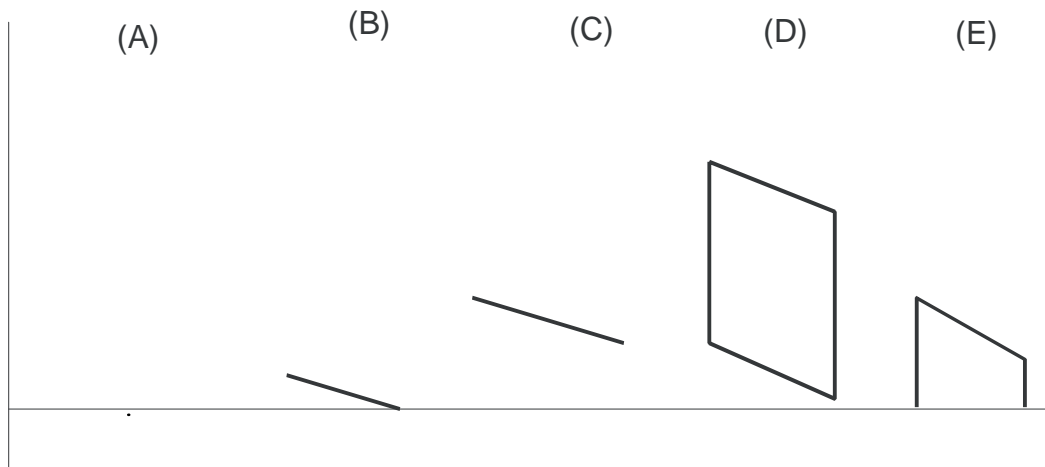


Figure 2.3 Différentes formes possibles pour l'ensemble $P(t)$

2.2.4 Modélisation des trajectoires d'évitement par offset :

La trajectoire d'évitement par offset comporte les étapes suivantes :

1. jusqu'à un temps t_0 , l'avion reste sur sa trajectoire d'origine
2. au temps t_0 , l'avion quitte sa trajectoire d'origine en effectuant un virage vers la gauche ou vers la droite, la valeur absolue de son changement de cap étant toujours égale à une même valeur, fixée, notée α . Il continue de s'éloigner de sa trajectoire jusqu'au temps t_1 .
3. au temps t_1 , il reprend son cap d'origine, et poursuit sa route sur une nouvelle trajectoire, parallèle à sa trajectoire d'origine.

Les deux premières étapes de l'évitement par offset sont en fait identiques aux deux premières étapes de l'évitement par point tournant. La trajectoire d'évitement par offset est représentée par la figure ci-dessous.

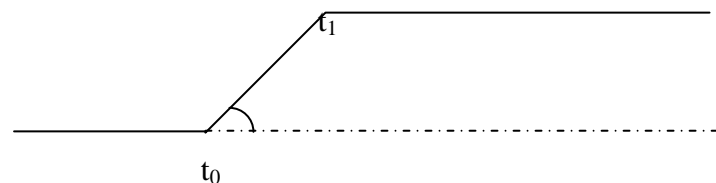


Figure 2.4 Modélisation d'une trajectoire d'évitement par offset

Si on note $P_1(t)$ l'ensemble des positions possibles de l'avion à l'étape i , les ensembles $P_1(t)$ et $P_2(t)$ sont les mêmes que ceux qui ont été définis précédemment (équations (2.1) et (2.2)).

L'ensemble $P_3(t)$ de la 3^{ème} étape est l'ensemble des points $p(t)$ dont les coordonnées $(x(t), y(t))$ sont données par les équations :

$$\begin{cases} x(t) = v.(t-(t_1-t_0))(1-\cos \alpha) & t_{0\min} \leq t_0 \leq t_{0\max} \\ y(t) = v.(t_1-t_0).\sin \alpha & t_{1\min} \leq t_1 \leq t_{1\max} \\ & t_0 \leq t_1 \leq t \end{cases}$$

On définit $t'_{1\max} = \min(t_{1\max}, t)$ et $t'_{0,1} = \max(0, t_{1\min} - t_{0\max})$. Si on note $\tau = t_1 - t_0$, les équations précédentes deviennent :

$$\begin{cases} x(t) = v.(t - \tau (1 - \cos \alpha)) \\ y(t) = v. \tau .\sin \alpha \end{cases}$$

avec la condition

$$t'_{0,1} \leq \tau \leq t'_{1\max} - t_{0\min}$$

Les équations et contraintes ci-dessus impliquent que $P_3(t)$ est un segment de droite dont les extrémités sont notées A3 et B3.

Les coordonnées des extrémités sont :

$$A3(t) \begin{cases} x_{A3}(t) = v.(t - (t'_{1\max} - t_{0\min}))(1 - \cos \alpha) \\ y_{A3}(t) = v.(t'_{1\max} - t_{0\min})\sin \alpha \end{cases}$$

$$B3(t) \begin{cases} x_{B3}(t) = v.(t - t'_{0,1})(1 - \cos \alpha) \\ y_{B3}(t) = v. t'_{0,1} \sin \alpha \end{cases}$$

On a donc :

$$\begin{cases} P_3(t) = [A3(t), B3(t)] & \text{si } t_{0\min} \leq t \leq t_{1\max} \\ P_3(t) = 0 & \text{sinon} \end{cases}$$

2.2.5 Distances entre avions :

Le premier but d'un système d'évitement, qu'il soit ou non automatique, est d'assurer la séparation des avions, c'est à dire d'assurer que, pendant toute la durée de l'évitement, les avions susceptibles d'être impliqués dans un conflit soient séparés deux à deux.

La distance entre deux points p et q du plan sera notée : d(p,q).

Soient a^i et a^j deux avions, les positions des avions au temps $t \in [0, t_f]$ sont $p^i(t)$ et $p^j(t)$ alors :

$$\delta^{ij}(t, t_0^i, t_1^i, t_0^j, t_1^j) = d(p^i(t), p^j(t))$$

est la distance entre elles à l'instant t

La condition de séparation est :

$$\delta^{ij}(t, t_0^i, t_1^i, t_0^j, t_1^j) \geq n_h \quad \forall t \in [0, t_f]$$

2.2.6 Retards :

Le retard subi par l'avion a^i s'il suit une trajectoire d'évitement définie par les temps de manœuvres t_0^i et t_1^i :

- pour l'évitement par point tournant le retard est :

$$r_{pt}^i(t_0^i, t_1^i) = 2(t_1^i - t_0^i)(1 - \cos \alpha) \quad (2.6)$$

- pour l'évitement par offset le retard est :

$$r_{off}^i(t_0^i, t_1^i) = (t_1^i - t_0^i)(1 - \cos \alpha) \quad (2.7)$$

On remarque que le retard d'un avion, pour les mêmes valeurs de t_0^i et t_1^i , est le même dans les deux cas, à un facteur 2 près.

Quand on remplace les temps de manœuvres t_0^i et t_1^i par les intervalles de manœuvres T_0^i et T_1^i , on cherche l'ensemble des retards possibles pour un avion dont la trajectoire d'évitement est définie par les intervalles T_0^i et T_1^i .

On ne peut pas remplacer, dans les équations (2.6) et (2.7), $t_1^i - t_0^i$ par $T_1^i - T_0^i$, au sens de la soustraction sur l'ensemble $I(\mathbb{R})$ des intervalles de \mathbb{R} (plus de détail[4]). En effet, il faut tenir compte des contraintes supplémentaires sur t_1^i et t_0^i :

$$t_1^i \leq t_0^i$$

On doit donc remplacer, $t_1^i - t_0^i$ par l'intervalle $T_{0,1}^i$ défini de la manière suivante :

$$T_{0,1}^i = \{y = t_1^i - t_0^i ; t_1^i \in T_1^i, t_0^i \in T_0^i \text{ et } t_1^i \leq t_0^i\}$$

Ce qui donne

$$T_{0,1}^i = [\max(0, t_{0\min}^i - t_{0\max}^i), t_{1\max}^i - t_{0\min}^i]$$

Donc pour des intervalles de manœuvres T_0^i et T_1^i , avec les deux modélisations envisagées, l'ensemble des retards possibles est un intervalle.

On note

$$r_{\min}^i(T_0^i, T_1^i) = \max(0, (t_{1\min}^i - t_{0\max}^i)(1 - \cos \alpha))$$

et

$$r_{\max}^i(T_0^i, T_1^i) = (t_{1\max}^i - t_{0\min}^i)(1 - \cos \alpha)$$

L'ensemble des valeurs possibles de $r^i(t_0^i, t_1^i)$ est donc donné par l'intervalle $R^i(T_0^i, T_1^i)$:

$$\boxed{R^i(T_0^i, T_1^i) = [r_{\min}^i(T_0^i, T_1^i), r_{\max}^i(T_0^i, T_1^i)]} \quad (2.8)$$

2.2.7 Application de l'algorithme d'optimisation :

Dans le cadre de l'évitement par point tournant comme dans celui de l'évitement par offset, l'optimisation doit être menée sur les temps de manœuvres et les sens de déviation. Les temps de manœuvres sont tels que l'on a $(t_0, t_1) \in [0, t_f]^2$.

Pour un problème à n avions, traités dans le cadre de l'approche globale, l'espace sur lequel sera mené l'optimisation sera donc :

$$E_{\text{opt}}^n = ([0, t_f] \times [0, t_f] \times \{-\alpha, \alpha\})^n$$

On cherchera donc à minimiser une fonction f_{ot}^n définie sur un sous ensemble D_{opt}^n de l'ensemble E_{opt}^n . L'ensemble D étant l'ensemble des points de E_{opt}^n permettant de définir une trajectoire d'évitement tel que l'on a $t_0 \leq t_1$:

$$D_{\text{opt}}^n = \{((t_0^1, t_1^1), \dots, (t_0^n, t_1^n)) \mid t_0^i \leq t_1^i, i=1, n\}$$

Un point de D_{opt}^n définit une trajectoire d'évitement pour chacun des n avions impliqués dans le conflit.

En considérant $t_r = ((t_0^1, t_1^1, \alpha_1), \dots, (t_0^n, t_1^n, \alpha_n)) \in D_{\text{opt}}^n$

$$f_{\text{opt}}^n = \sum_{i=1}^n t_1^i - t_0^i$$

La forme générale de l'algorithme sera :

1. la file F est initialisée. Elle contient 2^n éléments, qui sont les éléments de Γ_{opt}^n de la forme $(([0, t_f], [0, t_f], \alpha^1), \dots, ([0, t_f], [0, t_f], \alpha^n))$, correspondant chacun à une des 2^n combinaisons possibles pour des angles de déviation α^i des n avions.
2. on extrait le premier élément X de cette file
3. calcul de l'intervalle image et d'un estimateur $f_{\text{est}}(X)$ de cet intervalle.
4. $f_{\text{est}}^* \leftarrow f_{\text{est}}(X)$
5. Tant que $F \neq \emptyset$ on répète les étapes suivantes :
 - a. On extrait X le premier élément de la file F
 - b. Si X satisfait le critère C_{sat} on insère X dans $f_{\text{res}}^{\text{sat}}$
 - c. Si $f_{\text{res}}^{\text{sat}}$ est vide et si X satisfait le critère C_{ind} , insère X dans $f_{\text{res}}^{\text{ind}}$
 - d. Si X n'a été inséré ni dans $f_{\text{res}}^{\text{sat}}$ ni dans $f_{\text{res}}^{\text{ind}}$, si la taille de X est supérieure à $\varepsilon_{\text{rés}}^{\text{sat}}$ ou si $f_{\text{res}}^{\text{sat}}$ est vide, on coupe X en deux intervalles X_1 et X_2 ,

Les cinq étapes suivantes seront menées pour $j=1$ et $j=2$

- i- calcul d'un estimateur $f_{\text{est}}(X_j)$ et l'intervalle image de X_j
- ii- si X_j est non admissible, il est supprimé
- iii- si X_j est admissible, réduction de l'intervalle X_j et de son intervalle image

iv- si $f_{\text{est}}(X_j) < f_{\text{est}}^*$, $f_{\text{est}}^* \leftarrow f_{\text{est}}(X_j)$

v- si $f_{\text{min}}(X_j) \leq f_{\text{est}}^*$, on insère X_j dans la file F

6. quand $F = \emptyset$, si $f_{\text{res}}^{\text{sat}}$ n'est pas vide, le résultat de l'algorithme est le premier élément de $f_{\text{res}}^{\text{sat}}$ (une bonne solution). Si $f_{\text{res}}^{\text{sat}}$ est vide et $f_{\text{res}}^{\text{ind}}$ ne l'est pas le résultat de l'algorithme est le premier élément de $f_{\text{res}}^{\text{ind}}$ (une mauvaise solution). Si $f_{\text{res}}^{\text{sat}}$ et $f_{\text{res}}^{\text{ind}}$ sont vides, il n'y a pas de solution.

L'approche séquentielle procédera de la manière suivante :

1. on numérote les avions selon l'ordre de priorité
2. l'avion a_1 conserve sa trajectoire initiale
3. pour $i=2$ à n , on suit les étapes suivantes :
 - a. on applique pour générer une trajectoire pour l'avion a_i l'algorithme ci dessus, les contraintes étant imposées par les trajectoires générées par les avions a_0 à a_{i-1}
 - b. si une trajectoire d'évitement est trouvée pour l'avion a_i (l'algorithme fournit une solution, bonne ou mauvaise), les temps de manœuvres et le sens de déviation la décrivant sont mémorisés, et la trajectoire ainsi prévue pour l'avion a_i imposera des contraintes sur les trajectoires des avions a_j , $j > i$
 - c. si aucune trajectoire d'évitement n'est trouvée, fin de l'optimisation, pas de solution
4. si une solution a été trouvée pour chaque avion, l'algorithme propose les trajectoires prévues pour les n avions, en précisant les retards entraînés par ces trajectoires, et la séparation assurée dans les cas où c'est une mauvaise solution qui a été trouvée à l'étape (3b)
5. si une solution n'a pas été trouvée pour chacun des n avions, l'algorithme donne le numéro de l'avion à partir duquel il n'y a plus eu de solution.

Dans une étude réalisée par F.Médioni [4], il a été démontré que dans les deux approches, approche globale et approche séquentiel ou embarqué, l'algorithme des intervalles était sensiblement moins efficace en utilisant des trajectoires d'évitement par offset que quand il utilise des trajectoires d'évitement par point tournant. Les temps de

calcul sont meilleurs en utilisant les trajectoires par point tournant que celles en utilisant les trajectoires d'évitement par offset.

Dans son exemple de trois avions, $t_f=800s$, $\varepsilon_{res}^{sat}=10s$ le nombre d'intervalles $n_{int}=80$ de taille ε_{res}^{sat} dans $[0,t_f]$. Pour chaque avion, pour un sens de déviation donné, compte tenu du fait que les temps de manœuvres t_{i0} et t_{i1} doivent respecter l'inégalité $t_{i0} \leq t_{i1}$, on obtient $n_{int}(n_{int}+1)/2$ possibilités pour les intervalles de manœuvres. Les avions sont répartis sur un cercle de rayon de 28NM, convergent vers les centre à une vitesse de 420kt

- la taille maximale des intervalles insérés dans f_{satres} est $\varepsilon_{res}^{sat}=10s$
- la taille maximale des intervalles insérés dans f_{indres} est $\varepsilon_{res}^{sat}=5s$
- la norme de séparation est $n_h=5NM$
- le pas de temps utilisé pour le calcul des distances entre avions au cours de l'évitement est de 10s, ce qui donne comme valeur pour la norme de séparation majorée à $n_h=5,52NM$

les résultats pour l'approche globale sont très satisfaisant en terme de qualité, mais a nécessité un temps de calcul très important : 25 100s (7heures) de temps cpu sur un pentium II de 300Mhz, le retard moyen était de 16s.

Les résultats pour l'approche séquentiel, pour le cas test de 5 avions, sont très différentes en terme de qualités en fonction de l'ordre des avions choisie, le retard moyen sur les 5 avions impliqués dans le conflit varie ainsi, selon l'ordre de priorité des avions, de 20,8s à 26,1s.

2.3 Génération de trajectoires d'évitement utilisant l'algorithme A*:

L'algorithme A* est un algorithme permettant de rechercher dans un arbre le plus court chemin menant d'un état initial à un état terminal. Il s'agit d'un algorithme de recherche de type meilleur en premier, c'est-à-dire d'un algorithme dans lequel on développe à chaque étape le meilleur nœud généré non encore développé.

L'utilisation d'un algorithme de parcours d'arbre, comme l'algorithme A*, pour la recherche d'une trajectoire d'évitement, nécessite de pouvoir construire un arbre possédant les caractéristiques suivantes :

- la racine de l'arbre correspond à l'état initial de l'évitement : au temps 0, les avions ont leurs positions, vitesses et caps initiaux.
- Les feuilles terminales de l'arbre correspondent aux états finaux de l'évitement : tous les avions ont atteint leurs positions finales.
- Toutes les trajectoires possibles dans le cadre de la modélisation adoptée peuvent être représentées par un chemin entre la racine de l'arbre et une des feuilles.
- Le coût d'un chemin sur l'arbre doit correspondre à la fonction que l'on cherche à minimiser (dans notre cas : minimiser l'allongement des trajectoires).

2.3.1 Modélisation des trajectoires d'évitement :

La trajectoire d'évitement que nous allons décrire ici est du type « point tournant avec retour direct vers la destination » (voir figure 2.5), c'est-à-dire que les avions suivront des trajectoires normalisées, constituées de 4 étapes définies de la manière suivante :

- 1- Première étape (E0) : jusqu'à un temps $t_0 \geq 0$, l'avion suit sa trajectoire d'origine.
- 2- Deuxième étape (E1) : au temps t_0 , l'avion est dévié de sa trajectoire d'origine, vers la droite ou vers la gauche, mais avec un changement de cap dont la valeur absolue est déterminée, et garde son nouveau cap jusqu'au temps t_1 .
- 3- Troisième étape (E2) : au temps t_1 , l'avion prend le cap qui lui permet de rejoindre directement sa position finale.

- 4- Quatrième étape (Efin) : l'avion a atteint sa position finale, et poursuit sa route sur sa trajectoire initialement prévue.

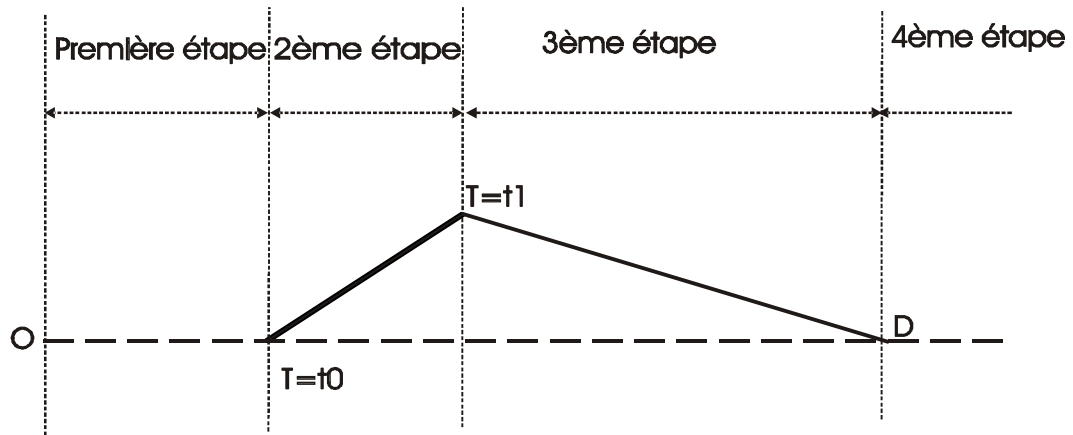


Figure 2.5 Trajectoire d'évitement avec un point tournant avec retour direct

Une telle trajectoire d'évitement normalisée est donc entièrement déterminée par les valeurs de t_0 et de t_1 et par la direction de la déviation subie par l'avion au temps t_0 (à droite ou à gauche).

Considérons les possibilités de choix offertes à un avion au cours du déroulement de sa trajectoire d'évitement :

Lorsqu'un avion est encore à la première étape de l'évitement (E0), il a à tout moment le choix entre trois actions :

1-poursuivre sa route en conservant son cap d'origine, en restant dans la première étape de l'évitement

2-modifier sa trajectoire en opérant un virage normalisé à gauche et passer ainsi à la deuxième étape de l'évitement (E1)

3-opérer un virage normalisé à droite et passer à l'étape de l'évitement (E1).

Si l'avion opère un virage, à droite ou à gauche, et passe ainsi à seconde étape de l'évitement, il n'a plus le choix, alors, qu'entre deux actions possibles :

1-poursuivre sa route en conservant son cap modifié, en restant dans la deuxième étape de l'évitement (E1).

2-opérer un virage pour suivre un cap direct vers sa position finale, il passe alors à la troisième étape de l'évitement (E2).

S'il opère ce virage et passe à la troisième étape de sa trajectoire d'évitement, l'avion n'a plus alors aucun choix : il garde le même cap jusqu'à ce qu'il atteigne sa position finale et reprend à ce moment sa route originelle.

2.3.2 L'algorithme A* :

L'utilisation de l'algorithme A* nécessite la connaissance des données suivantes :

- u_0 : l'état initial
- T : l'ensemble des états terminaux
- p_1, p_2, \dots, p_n : l'ensemble des règles de production, permettant de générer les fils d'un nœud en fonction de l'état correspondant à ce nœud.
- $k(u,v)$: si u et v sont des nœuds de l'arbre, k est une fonction donnant le coût du passage par l'arc (u,v).
- $h(u)$: u est un nœud de l'arbre, h est une fonction heuristique qui estime le coût de passage de u à un état terminal, c'est-à-dire le minimum sur tous les chemins $(u,v_1,v_2,\dots,v_n,u_t)$, où u_t est un état terminal, de la somme :

$$k(u,v_1)+k(v_1,v_2)+\dots+k(v_n,u_t)$$

ce minimum sera noté $h^*(u)$

Pour décrire le fonctionnement d'un algorithme A*, nous aurons besoins des variables suivantes :

- G : la liste des états générés mais non encore développés
- Premier(G) : une fonction qui renvoie le premier élément de la liste G
- D : la liste des états générés et développés
- $g(v)$: une fonction de l'état (v) donnant le coût total du trajet pour atteindre l'état (v) à partir de l'état initial

- $f(v)$: une fonction de l'état (v) donnant le coût total estimé de l'état (v), c'est-à-dire que $f(v)=g(v)+h(v)$, où $h(v)$ est l'heuristique définie ci-dessus.
- Insérer_selon_f(G,v) : une fonction qui insère v dans l'ordre f croissant puis g décroissant
- père(v) : un tableau indexé par les états, donnant pour un état (v), l'état qui l'a généré.

Une fois définies ces variables, l'algorithme A^* , dans sa forme la plus générale, peut se décrire de la manière suivante :

- 1- $G \leftarrow (u_0)$; $D \leftarrow 0$; $g(u_0) \leftarrow 0$; $f(u_0) \leftarrow 0$;
 - 2- Tant que $G \neq 0$ faire
 - a. $U \leftarrow \text{premier}(G)$; $G \leftarrow G/(u)$; $D \leftarrow DU(u)$
 - b. Si u appartient à T alors donner-solution ; fin-pgme ; fin-si
 - c. Pour i allant 1 jusqu'à n faire
 - i. $v \leftarrow u$, utilisant la règle π_i
 - ii. Si $v \notin DUG$ ou $[g(v) > g(u) + k(u,v)]$ alors
 - $g(v) \leftarrow g(u) + k(u,v)$
 - $f(v) = g(v) + h(v)$
 - père(v) $\leftarrow u$
 - insérer-selon-f(G,v)
 - fin-si
- fin faire
- fin faire

Dans le cadre de l'approche séquentielle, on procède selon les étapes suivantes :

- 1- on établit entre les avions un ordre de priorité. On supposera ensuite que les avions sont numérotés de 1 à n selon cet ordre.
- 2- La trajectoire de l'avion a_1 est inchangée : l'avion a_1 conserve sa trajectoire initialement prévue, c'est-à-dire, ici, la trajectoire qui va de sa position initiale à sa position finale en suivant une ligne droite.
- 3- On génère ensuite successivement les trajectoires des avions qui sont numérotés de 2 à n . Pour $1 < i < n$, on considère que les trajectoires des avions j , pour $j < i$ sont fixés et on génère la trajectoire de l'avion i de manière à ce que :

- l'avion a_i soit à tout moment séparé de chaque avion a_j , pour $j < i$
- l'allongement de la trajectoire de l'avion a_i soit minimale, compte tenu des contraintes de séparation.

2.3.3 Recherche dans l'arbre :

La construction et le parcours de l'arbre pour la recherche d'une trajectoire optimisée, nécessite un arbre possédant les caractéristiques suivantes :

- La racine de l'arbre correspond à l'état initial de l'évitement au temps $t=0$, elle contient la description des trajectoires des avions contraintes ainsi que le cap, la position et la destination de l'avion central (ici l'avion qui doit établir sa nouvelle trajectoire).
- Les feuilles terminales de l'arbre correspond aux états dans lequel on a terminer la simulation ou aux états dans lequel l'avion a atteint sa destination.
- Chaque branche de l'arbre représente une trajectoire de l'avion central.
- Le coût d'un chemin sur l'arbre doit correspondre à la fonction que l'on recherche à minimiser (c'est la longueur de la trajectoire).

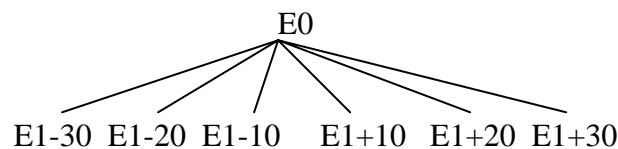


Figure 2.6 Schéma de l'arbre de recherche de trajectoires

Dans le premier état (E0), l'avion suit sa trajectoire prévue jusqu'à t_0 . Pendant le deuxième état (E1), à partir de t_0 , l'avion est dévié de sa trajectoire d'origine, vers la droite ou vers la gauche d'un cap de plus ou moins 10, 20 ou 30°. Ce cap est conservé jusqu'au temps t_1 . Lors du troisième état (E2), au temps t_1 , l'avion prend le cap qui lui permet de rejoindre directement sa position finale. Enfin, dans le dernier état (E3), l'avion a atteint sa position finale.

La nouvelle trajectoire de l'avion est donc entièrement déterminée par t_0 , t_1 et le cap pris en déviation.

2.4 Génération de trajectoires d'évitement utilisant l'algorithme génétique:

2.4.1 Principe :

Les algorithmes génétiques forment une branche d'une famille d'algorithmes d'optimisation que l'on peut appeler les algorithmes évolutionnaires [6].

Le principe de base des algorithmes génétiques s'inspire, très schématiquement, de l'évolution naturelle. Le vocabulaire employé révèle clairement cette source d'inspiration : on parle de population, de générations, de sélection, de reproduction, de croisement, de mutation, de chromosomes.

Les origines de ces algorithmes remontent à 1962, aux travaux de Holland. Mais ce n'est qu'à partir de 1989 qu'ils ont connu une véritable popularité.

L'algorithme génétique suit les étapes suivantes :

- 1- on génère d'abord les éléments de la population initiale, ces éléments sont appelés individus. Ils codent les points de l'espace à optimiser qu'on note E_{opt} . La partie d'un individu qui porte le codage d'un point de l'espace d'états E_{opt} est souvent appelée chromosome. Ce codage est aussi désigné par le terme de génotype, tandis que le point de E_{opt} qu'il code est désigné par le terme phénotype.
- 2- on calcul la fonction d'adaptation $f_a(e)$ pour chacun des éléments e de la population. La manière dont est calculée cette fonction dépend bien entendu du problème.
- 3- On opère la reproduction des éléments de la population, les éléments les plus adaptés se reproduisant mieux que les éléments moins adaptés. Le produit de cette étape de reproduction est une nouvelle population, contenant le même nombre d'éléments que la population de départ.
- 4- On applique deux opérateurs dont l'effet est de diversifier la population : l'opérateur de croisement et l'opérateur de mutation.

Croisement : on choisit aléatoirement, avec une probabilité P_c fixée, des couples d'éléments de la population.

L'opérateur de croisement, à partir de chaque couple de chromosomes, génère deux nouveaux chromosomes qui en sont issus et qui les remplacent dans la population, on appellera les éléments choisis dans la population, et qui à partir desquels on opère le croisement, les parents, tandis que les produits de ce croisement seront appelés les enfants.

Mutation : on choisit aléatoirement des éléments de la population, avec la probabilité P_m . Chacun de ces éléments est remplacé dans la population par un nouvel élément, obtenu à partir de lui par une perturbation aléatoire.

5- on revient à la deuxième étape, et on parcourt, à chaque itération, les 5 dernières étapes. On parle, par analogie avec les mécanismes de l'évolution naturelle dont s'inspire les algorithmes génétiques, de génération. Une nouvelle population apparaît à chaque itération de ces étapes, c-à-d à chaque génération. Cette population est issue, à travers la reproduction, le croisement et la mutation, de la population présente à la génération précédente. Certains éléments de la population peuvent demeurer inchangés d'une génération à l'autre s'ils n'ont subi ni croisement ni mutation. Le nombre d'éléments de la population reste constant à travers les générations.

D'après [4] le plus souvent, il n'est pas possible de comparer le meilleur élément obtenu à l'optimum cherché. Il faut alors choisir un critère d'arrêt arbitraire.

Les critères d'arrêt les plus communément utilisés sont les suivants :

- 1- un nombre maximal de générations peut-être fixé à priori
- 2- le processus peut-être arrêté quand la population s'est stabilisée, c'est-à-dire n'évolue plus, ou quand la valeur de la fonction d'adaptation du meilleur élément a cessé d'augmenter ou n'augmente plus suffisamment rapidement.

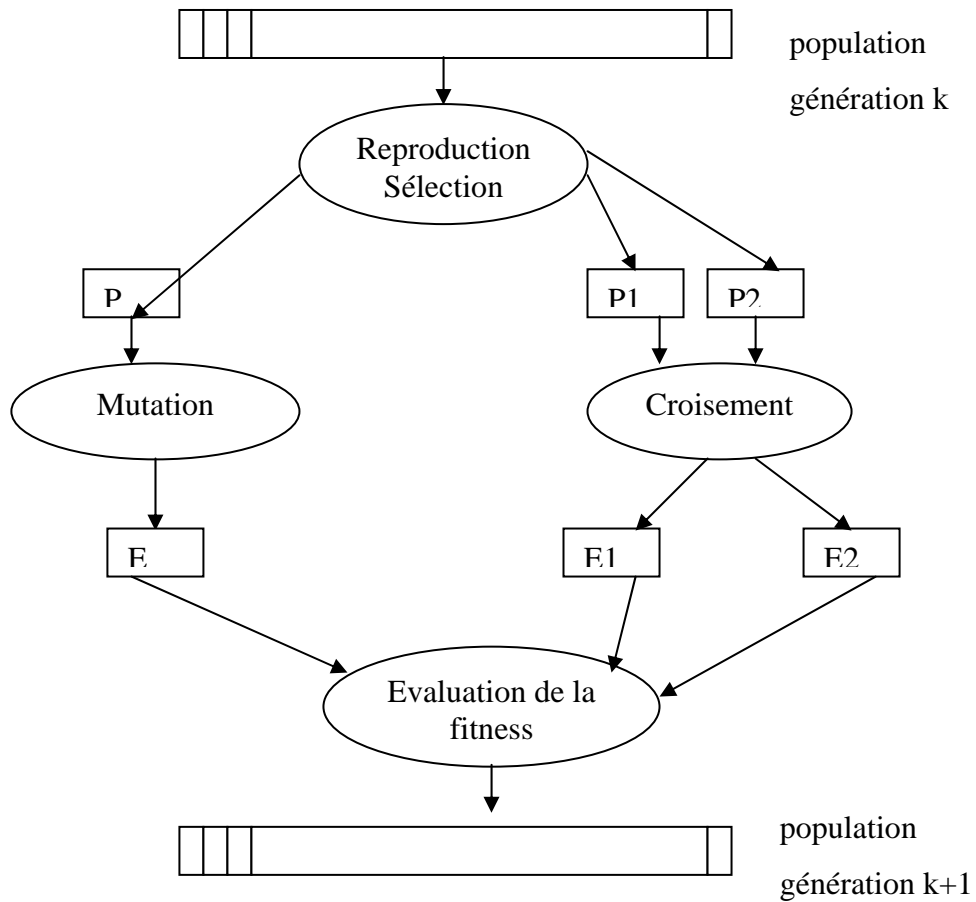


Figure 2.7 Principe général des algorithmes génétiques

2.4.2 Codage :

Si on suppose que les modèles de manœuvres utilisés pour l'évitement sont soit le modèle de point tournant ou le modèle d'offset. Le schéma suivant illustre les instants de manœuvres pour chacun des deux modèles.

Modèle d'offset



Modèle de point tournant

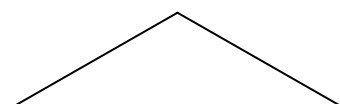


Figure 2.8 Modèles de manœuvre

Nous obtenons donc le modèle où une manœuvre est déterminée par les 4 paramètres suivants :

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

Pour le problème de résolution de conflits aériens, les variables t_0 , t_1 , t_2 et α de chaque avion sont disposées dans une matrice qui constitue un individu de la population.

On appellera gènes les variables t_0 , t_1 , t_2 et α de chaque avion. Un élément de population contient donc $4n$ gènes.

Tableau 2.1 Codage d'un individu à n avions

t_0^1	t_1^1	t_2^1	α_1
t_0^2	t_1^2	t_2^2	α_2
t_0^3	t_1^3	t_2^3	α_3
t_0^4	t_1^4	t_2^4	α_4
t_0^5	t_1^5	t_2^5	α_5
.	.	.	.
.	.	.	.
t_0^n	t_1^n	t_2^n	α_n

gène t_2 de l'avion 3

Lorsqu'un conflit à n avions est fourni à l'algorithme génétique, il se peut que certains gènes correspondant à des manœuvres déjà entamées ou qui doivent être entamées avant Δ minutes soient imposés à l'algorithme génétique. Ainsi, tous les gènes t_0 , t_1 , t_2 dont la valeur est inférieure à Δ sont imposés dans l'algorithme et ne

pourront plus être modifiés par la suite. De même si $t_0 < \Delta$ alors l'angle α correspondant est fixé et ne pourra plus être modifié. Toutes ces variables ne seront donc pas initialisées de façon aléatoire.

Les contraintes liées à la séparation des avions sont directement introduites dans la fitness et ne sont pas gérées à priori mais à posteriori.

2.4.3 Fitness :

La fonction fitness doit prendre en compte à la fois les contraintes de séparation des avions et la qualité de la résolution lorsque celle-ci est réalisée.

La fonction fitness est représentée par une matrice triangulaire inférieure F . Si le cluster à résoudre comprend n avions alors « n » sera la taille de la matrice.

Si $i \neq j$, alors F_{ij} mesure la gravité du conflit entre l'avion i et l'avion j

S'il n'y a pas de conflit $F_{ij}=0$

F_{ij} croît avec la gravité du conflit

F_{ii} mesure la pénalisation de la trajectoire de l'avion i provoquée par ses manœuvres.

Le calcul des éléments de la matrice F se fait de la manière suivante :

1- les éléments de la diagonale sont calculés par la formule :

$$F_{i,i} = S_i + kN_i$$

S_i est la surface occupée par la résolution (figure 2.9)

N_i est le nombre de manœuvres (égale à 2 pour modèle point tournant et égale à 3 pour modèle offset)



Figure 2.9 la surface occupée par la manœuvre de résolution

2- Evaluation 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 les temps t pour donner $F_{i,j}$ mesure du conflit entre i et j .

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

La fitness scalaire peut se calculer de la façon suivante :

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

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

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).

2.4.4 Opérateurs de croisement et de mutation :

Le croisement a pour but d'enrichir la diversité de la population en manipulant la structure des chromosomes.

Pour effectuer le croisement, on tire aléatoirement une position dans chacun des parents P1 et P2. On échange ensuite les deux sous chaînes terminales de chacun des deux chromosomes, ce qui produit deux enfants E1 et E2. Ce type de croisement est représenté par la figure suivante :

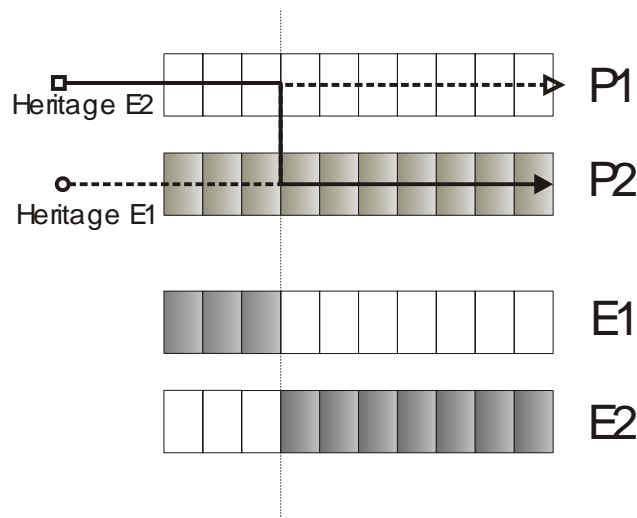


Figure 2.10 exemple de slicing crossover

On peut étendre ce principe en découpant le chromosome non pas en 2 sous chaînes mais en 3, 4, etc... (figure 2.11)

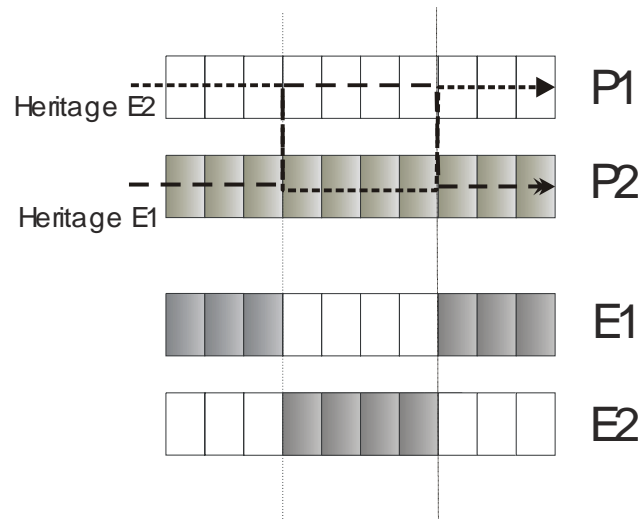


figure 2.11 exemple de slicing crossover à deux points

Si on considère $P1(i)$ et $P2(i)$ respectivement le i -ème gène du père P1 et le i -ème gène du père P2, alors on obtient :

$$C1(i) = \alpha P1(i) + (1 - \alpha) P2(i) \text{ le } i\text{-ème gène du fils1}$$

$$C2(i) = (1 - \alpha) P1(i) + \alpha P2(i) \text{ le } i\text{-ème gène du fils2}$$

$$\text{Où } \alpha \in [-0.5, 1.5]$$

L'opérateur de mutation a pour but d'apporter aux algorithmes génétiques la propriété d'ergodicité de parcours d'espace. [4]

Ergodicité signifie que l'algorithme génétique sera susceptible d'atteindre tous les points de l'espace d'état sans pour autant les parcourir tous dans le processus de résolution.

L'opérateur de mutation consiste généralement à tirer aléatoirement un gène dans le chromosome et à le remplacer par une valeur aléatoire.

Ces opérateurs (croisement et mutation) ont l'avantage d'être assez déterministes en début de convergence de sorte qu'une solution sans conflit (dont la fitness est supérieure à 0.5) peut être rapidement dégagée. Quand les solutions sans conflits deviennent suffisamment nombreuses, ces opérateurs deviennent moins déterministes et la recherche dans l'espace d'état devient plus large. Ceci conduit à l'utilisation de certaines méthodes telle que : le sharing, pour atténuer l'effet du déterminisme introduit.

CHAPITRE 3

ELABORATION DU CODE DE CALCUL

3.1 Environnement de programmation :

Un code de calcul a été développé pour la détection et la résolution de conflits aériens, que nous avons nommé « Solveur embarqué », utilisant l'algorithme A*, dans un environnement windows XP disposant de l'outil de programmation « Fortran Power Station V4.0 ».

Le Fortran étant le langage de programmation utilisé pour l'écriture de ce « solveur embarqué », offrant plusieurs possibilités et avantages : la programmation modulaire, outils graphiques disponibles, simplicité du langage et son universalité dans le monde académique.

Les caractéristiques du matériel informatique utilisé pour la simulation de plusieurs cas de conflits aériens, sont les suivants :

- micro-ordinateur compatible IBM Pentium IV, 2.0 GHz
- 256 MO de mémoire
- 20 GO de Disque Dur

Du fait de la complexité du problème traité, un certain nombre d'hypothèses sont fixées, tenant compte des paramètres pratiques réelles, afin de rendre la modélisation du problème moins complexe et réalisable.

3.2 Hypothèses du travail :

- Un avion est capable de communiquer à tout autre avion, donc détecter un conflit, dans un rayon de 90 Nautiques(5NM équivalant à 9,3 kilomètres).
- La norme de séparation horizontale est égale à 5NM
- La norme de séparation verticale est égale à 1000ft (1000ft → 304,8mètres)
- Les avions sont supposés voler en route directe.
- Les vitesses de vol sont supposées constantes durant la période de la simulation.
- Le temps de simulation ou d'anticipation est de 20min
- La trajectoire d'évitement est du type point tournant avec retour directe

- Pour résoudre un conflits, la trajectoire d'évitement est recherchée dans le plan horizontal en premier lieu, dans le cas échéant, dans le plan vertical.
- Les angles de déviation sont limités aux valeurs suivantes : 10° , 20° et 30° à gauche ou à droite.

3.3 Les objectifs :

- Chaque avion doit pouvoir détecter tout éventuel conflit dans un rayon de 90NM
- En cas d'un conflit potentiel, trouver les trajectoires libres de tout conflit
- Minimiser simultanément, les critères :
 - *la durée de résolution du conflit
 - *les retards dues aux manœuvres

3.4 Les manœuvres utilisées :

Les manœuvres de résolution proposées aux avions sont simples. Elles comportent les manœuvres sur plan horizontal et vertical et sont respectivement de type point tournant et de type offset.

La trajectoire que nous utilisons est la trajectoire d'évitement par point tournant(voir figure 3.1). Donc, la trajectoire de l'aéronef est décrite par les éléments suivants :

- l'instant t_0 de début de manœuvre d'évitement
- l'angle α alpha de déviation de la trajectoire d'origine
- l'instant t_1 de retour vers le point de destination final

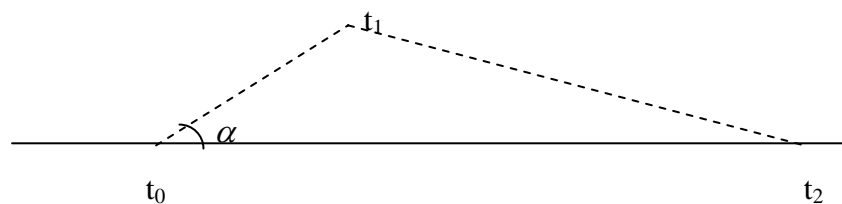


Figure 3.1 Représentation de la trajectoire d'évitement par point tournant.

Ce type de manœuvre permet de rester proche de la trajectoire optimale, est simple à réaliser par un pilote humain, et facile à comprendre par les pilotes des avions voisins concernés.

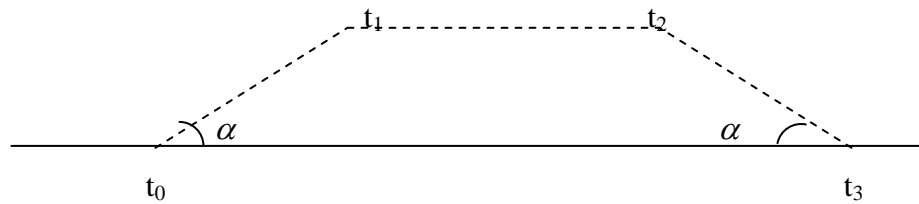


Figure 3.2 Représentation de la trajectoire d'évitement par offset.

La manœuvre de type offset permet de modifier le niveau de vol de l'aeronef, soit d'un niveau plus haut ou d'un niveau plus bas (+ ou - 1000 ft), de maintenir ce niveau jusqu'à l'instant t_2 puis revenir à son niveau d'origine, et c'est ce qui se passe dans la pratique.

Soient (x_d, y_d, z_d) les coordonnées d'un avion à l'instant $t=0$, V , C_i , C_d et C_r sont respectivement la vitesse, le cap initial, le nouveau cap de déviation et le nouveau cap de retour sur la trajectoire initiale, la position de l'avion à un instant T est calculée comme suit :

Cas de trajectoire d'évitement par point tournant :

$T \leq t_0$

$$\left\{ \begin{array}{l} X = x_d + V \cdot \cos C_i \cdot T \\ Y = y_d + V \cdot \sin C_i \cdot T \\ Z = z_d \end{array} \right.$$

$t_0 < T \leq t_1$

$$\left\{ \begin{array}{l} X = x_{t_0} + V \cdot \cos C_d \cdot (T - t_0) \\ Y = y_{t_0} + V \cdot \sin C_d \cdot (T - t_0) \\ Z = z_d \end{array} \right. \quad \text{avec } x_{t_0}, y_{t_0} : \text{coordonnées à l'instant } t_0$$

$t_1 < T \leq t_2$

$$\left\{ \begin{array}{l} X = x_{t_1} + V \cdot \cos C_d \cdot (T - t_1) \\ Y = y_{t_1} + V \cdot \sin C_d \cdot (T - t_1) \\ Z = z_d \end{array} \right. \quad \text{avec } x_{t_1}, y_{t_1} : \text{coordonnées à l'instant } t_1$$

Cas de trajectoire d'évitement par offset :

T ≤ t0

$$\left\{ \begin{array}{l} X = x_d + V \cdot \cos C_i \cdot T \\ Y = y_d + V \cdot \sin C_i \cdot T \\ Z = z_d \end{array} \right.$$

t0 < T ≤ t1

$$\left\{ \begin{array}{l} X = x_{t0} + V \cdot \cos inc \cdot \cos C_d \cdot (T - t_0) \quad \text{avec } x_{t0}, y_{t0} : \text{coordonnées à l'instant } t_0 \\ Y = y_{t0} + V \cdot \cos inc \cdot \sin C_d \cdot (T - t_0) \quad inc : \text{angle de descente ou de montée} \\ Z = z_d + V \cdot \sin inc \cdot (T - t_0) \end{array} \right.$$

t1 < T ≤ t2

$$\left\{ \begin{array}{l} X = x_{t1} + V \cdot \cos C_d \cdot (T - t_1) \quad \text{avec } x_{t1}, y_{t1}, z_{t1} : \text{coordonnées à l'instant } t_1 \\ Y = y_{t1} + V \cdot \sin C_d \cdot (T - t_1) \\ Z = z_{t1} \end{array} \right.$$

t2 < T ≤ t3

$$\left\{ \begin{array}{l} X = x_{t2} + V \cdot \cos C_d \cdot (T - t_2) \quad \text{avec } x_{t2}, y_{t2}, z_{t2} : \text{coordonnées à l'instant } t_2 \\ Y = y_{t2} + V \cdot \sin C_d \cdot (T - t_2) \\ Z = z_{t2} - V \cdot \sin inc \cdot (T - t_2) \end{array} \right.$$

3.5 Principe général de fonctionnement :

Le principe général de fonctionnement du code de calcul que nous avons développé, suit les étapes suivantes :

1^{ère} étape : Lecture des données d'avions, et des paramètres de calcul

2^{ème} étape : pour chaque avion, on effectue :

- 1- Détection de conflits
- 2- Génération de toutes les trajectoires d'évitement libres de tout conflit, en considérant les trajectoires des avions les plus prioritaires comme contraintes, et calcul des retards induits
- 3- Choix de la trajectoire optimale

3^{ème} étape : affichage des résultats numériques, et des représentations graphiques

Donc, le code de calcul dispose de deux fichiers de données, le premier est celui qui contient les paramètres de calcul, et qui sont : le temps de la fenêtre d'anticipation ou de la simulation, la norme de séparation horizontale, la norme de séparation verticale et le rayon de détection. Le deuxième fichier est organisé en lignes de données, où chaque ligne contient les informations d'un avion donné, et qui sont : les coordonnées de l'avion (x,y,z), la vitesse de vol, le cap et l'angle de montée ou de descente.

Ce deuxième fichier de données est organisé de façon que l'avion dont les données se trouvent sur la première ligne est plus prioritaire que celui dont les données sont sur la deuxième ligne, et ainsi de suite, jusqu'à la dernière ligne du fichier qui contient les informations de l'avion le moins prioritaire. Donc, c'est ainsi que l'ordonnancement des avions est établi.

3.6 La Détection de conflits :

A chaque avion est associée une première sphère dont les rayons définis par : Rx(rayon dans la 1^{ère} dimension), Ry(rayon dans la 2^{ème} dimension) et Rz(rayon dans la 3^{ème} dimension), vérifient l'équation suivante :

$$R_x=R_y=R_z=\text{Rayon de détection fixé à } 90\text{NM.}$$

Soient (x_a,y_a,z_a) et (x_b,y_b,z_b) les coordonnées de deux avions, respectivement l'avion A de référence et l'avion B, on dira que l'avion A a détecté un conflit avec l'avion B si la condition suivante est vérifiée:

$$\frac{((x_b-x_a)^2/R_x^2)+((y_b-y_a)^2/R_y^2)+((z_b-z_a)^2/R_z^2)-1}{2} \leq 0$$

Cette condition si elle est vérifiée, se traduit par : l'avion B est localisé à l'intérieur de la sphère de l'avion A (qui est l'avion de référence).

Une fois le conflit détecté, la procédure de résolution du conflit est déclenchée lorsque l'avion B franchisse une deuxième sphère interne, associée toujours à l'avion A.

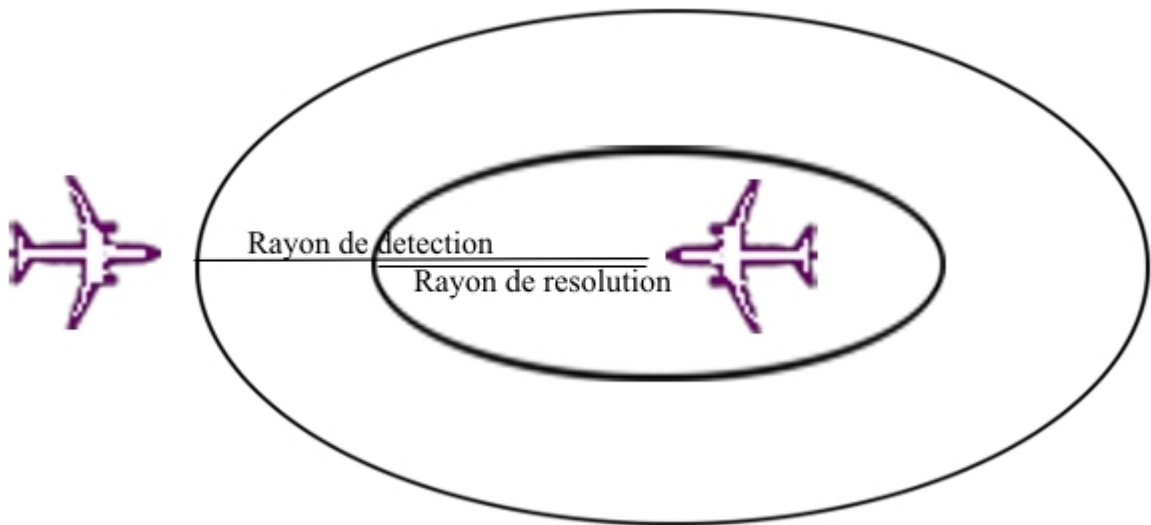


Figure 3.3 Schéma de représentation des champs de détection et de résolution

La question que nous nous sommes posées pour pouvoir calculer les rayons de cette deuxième sphère :

A quelle distance, deux avions en conflit volant face face à la même vitesse, pour qu'un des avions soit dévié de sa trajectoire d'un angle de 10° (à gauche ou à droite) pour éviter la collision ?

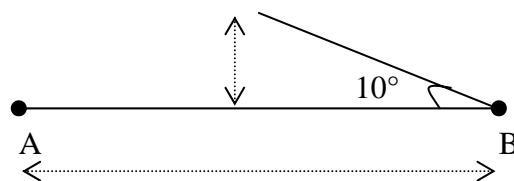


Figure 3.4 Schéma de trajectoire d'évitement d'un angle de 10°

Donc, nous avons supposés que l'avion B en effectuant une déviation de 10° va atteindre un point distant de l'axe de la trajectoire originelle d'une valeur est égale à la norme de séparation, et cette distance multiplié par deux sera la distance qu'on cherche à déterminer.

Soit Nh la norme de séparation horizontale, la distance parcourue par l'avion B jusqu'au point distant de la trajectoire originelle de Nh est notée « d » alors :

$$\frac{Nh}{d} = \sin \frac{\pi}{18}$$

ce qui donne,

$$d = \frac{1}{\sin \frac{\pi}{18}} \cdot Nh = 6 * Nh$$

D'où la distance entre les deux avions doit être au moins égale à $12 * Nh$ pour pouvoir garantir l'évitement.

Les rayons de la deuxième sphère définis par : RRx (rayon de la 1^{ère} dimension), RRy (rayon de la 2^{ème} dimension) et RRz (rayon de la 3^{ème} dimension), vérifient l'équation suivante :

$$RRx = RRy = 12 * Nh$$

Pour des raisons pratiques et opérationnelles le rayon de la sphere de résolution en Z aura la valeur suivante :

$$RRz = 6 * Nv \quad \text{avec } Nv : \text{ la norme de séparation verticale}$$

Le « pas de temps » pour suivre l'évolution des avions en vol, est un paramètre qui affecte sensiblement le temps d'exécution, et aussi les solutions de résolution de conflits. C'est pourquoi, deux pas de temps le premier est égale à 1 seconde et le deuxième à 10 secondes, ont été choisie pour effectuer des tests et de faire le constat.

3.7 Description du code de calcul :

Le code de calcul se compose des modules suivants :

- module d'initialisation « INIT »: Le module de lecture des paramètres de calcul, et des données des avions.
- module de détection de conflits « CONF-DETECT »
- module de recherche des trajectoires libre de tout conflit pour un avion sur le plan horizontal « H-TRAJ »
- module de recherche des trajectoires libre de tout conflit pour un avion sur le plan vertical « V-TRAJ »
- module de calcul des retards « CAL-RETARD »
- module de choix de trajectoire optimale « OPT-TRAJ »
- module graphique de tracer des la trajectoire originelle de chaque avion « ORIG-TRAJ »
- module graphique de tracer de la trajectoire d'évitement « SOL-TRAJ »
- module d'affichage des résultats numériques « NUM-RES »

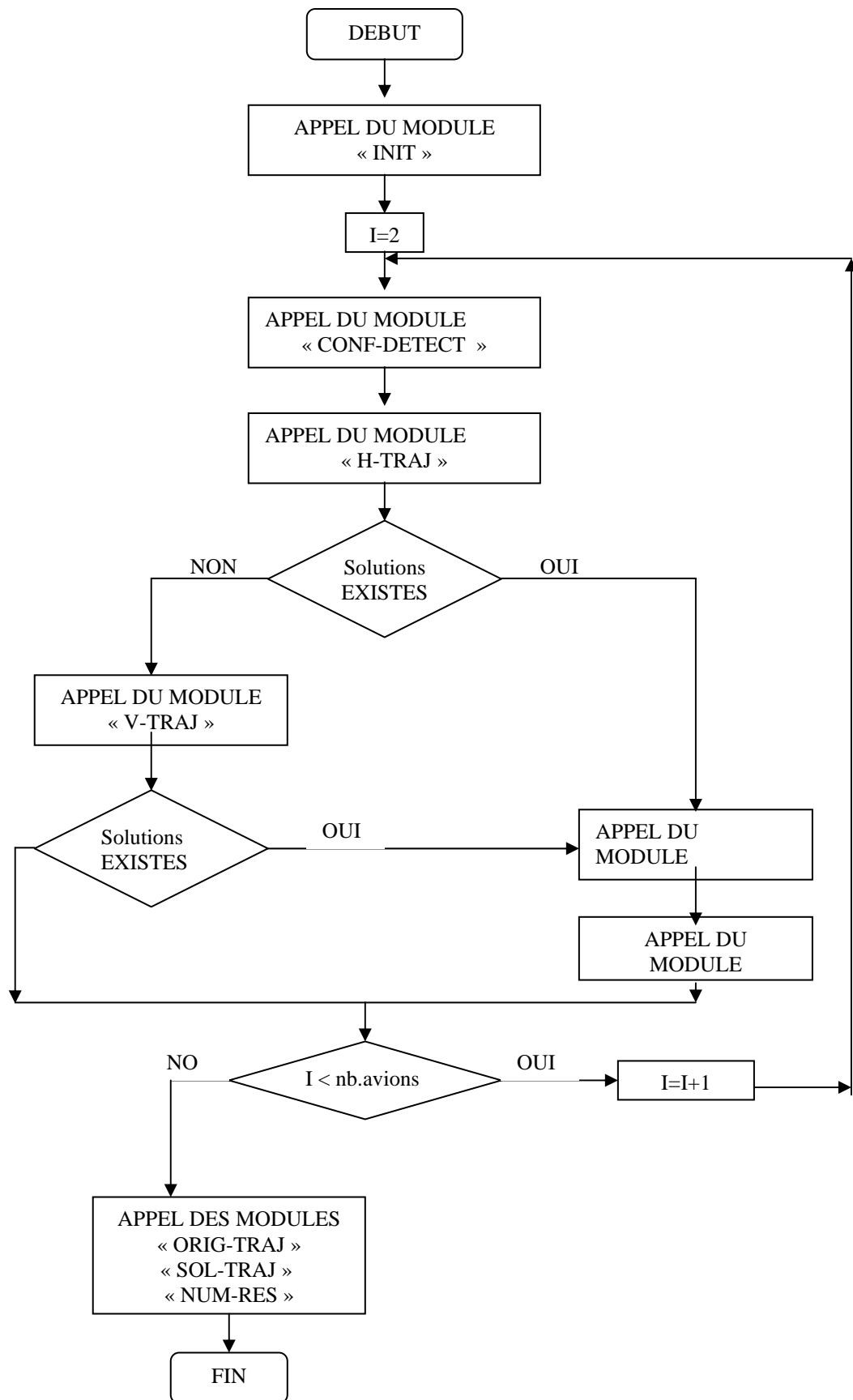


Figure 3.5 Schéma de l'organigramme du code de calcul

CHAPITRE 4

RESULTATS ET COMMENTAIRES

Des résultats de plusieurs simulations de cas de conflits aériens sont présentés. Il s'agit des cas de conflits aériens simples, mais aussi des cas complexes, l'influence de l'ordonnancement des avions sur la solution optimale locale pour l'avion (qui se traduit par le retard de chacun des avions induit par la trajectoire d'évitement) et sur la solution optimale globale de tous les avions (qui se traduit par la moyenne des retards) est démontrée. Le temps d'exécution (qui se traduit par le temps CPU de chacun des cas présentés) est un des paramètres décisif sur l'efficacité, la faisabilité et la rapidité de réponse exigée d'un tel code s'il est embarqué sur un avion dans le cas pratique.

Le choix de deux pas de temps pour l'exécution ont été choisie, un premier pas de temps d'une seconde (1sec) est appliqué, qui pénalise les résultats en terme de temps de calcul, mais, permet d'avoir des résultats beaucoup plus précises. Un deuxième pas de 10 secondes permet de réduire le temps calcul, et de voir si les réponses du système sont aussi précises.

Dans tous les cas de figures traités, un ordonnancement des avions par priorité a été appliqué, l'avion ayant le numéro le plus faible est le plus prioritaire et ne modifie pas sa trajectoire originelle.

1^{er} CAS : Conflit de deux avions en vol face à face, ayant la même vitesse et le même niveau de vol (FL320) :

Tableau 4.1 les données en entrée des deux avions

Avions	X(NM)	Y(NM)	Altitude(ft)	Vitesse(m/s)	Cap	Incidence
A	80.	0.	32000.	260	180.°	0.°
B	-40.	0.	32000.	260	0.°	0.°

Tableau 4.2 Résultats numériques pour le cas de deux avions

Avion	Début de déviation (sec)	Instant de retour(sec)	déviatiion horizontal	déviatiion vertical	Retard	Dist-Suppl (kms)		
A	Trajectoire originelle maintenue							
B	216	646	10°L	0°	11s	2.9858	0.96%	
<i>Moyenne des retards</i>		<i>6s</i>	<i>Nbre de conflits</i>		<i>1</i>	<i>Temps CPU</i>		<i>Moins de 1s</i>

L'avion B étant moins prioritaire que l'avion A, effectue une déviation de 10° à gauche, et accuse un retard de 11 seconde, ce qui représente un allongement de 0.96% de la longueur de la trajectoire originelle, qui est très faible, donc la solution est très acceptable.

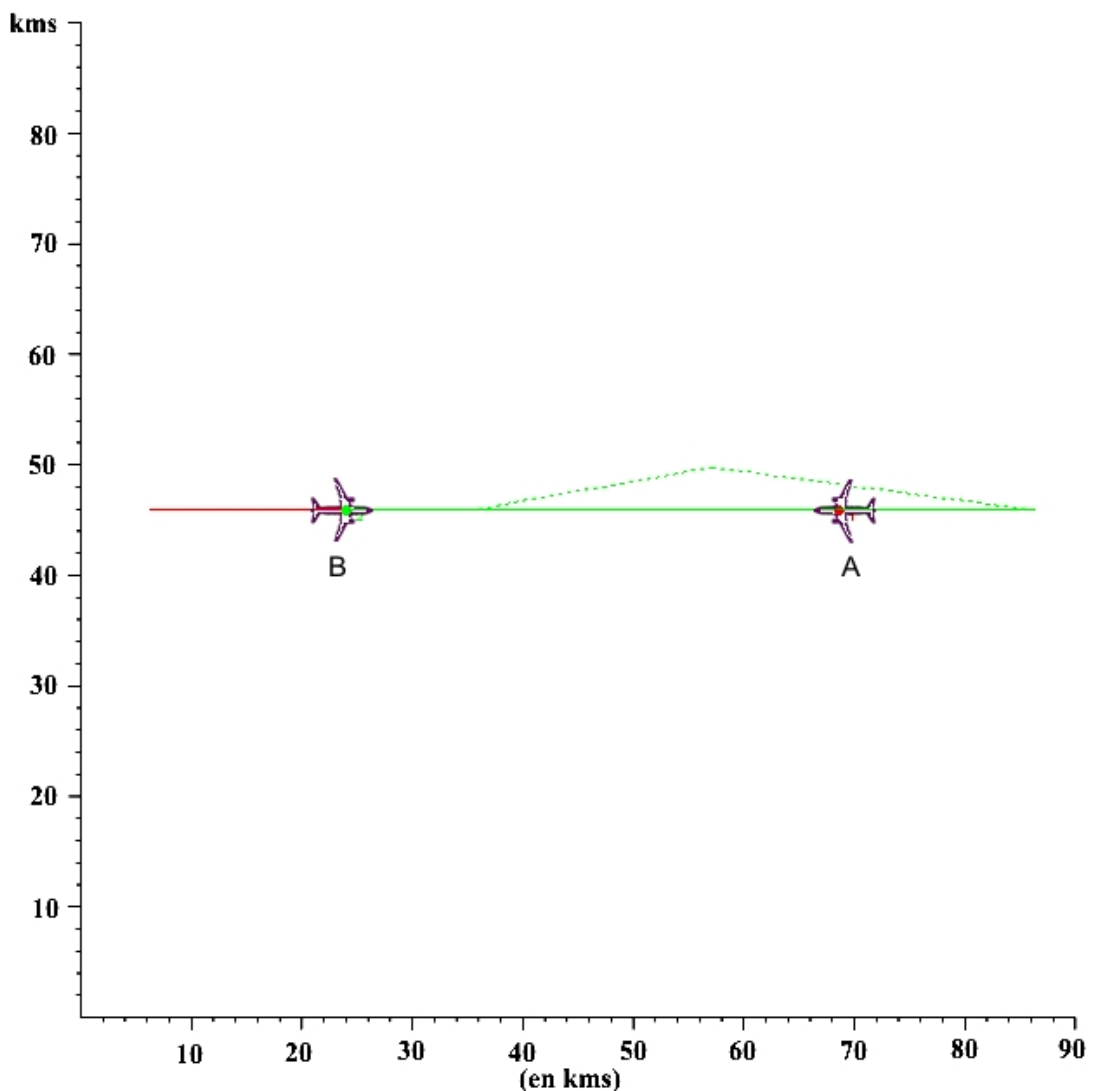


Figure 4.1 représentation graphique du conflit de 2 avions en route directe

2^{ème} CAS : Conflit de trois avions ayant la même vitesse et le même niveau de vol(FL320):

Trois cas de figures sont présentés et qui permettent de constater l'importance quant au choix de l'ordre des avions (du prioritaire au moins prioritaire) et de son influence sur les résultats obtenus.

Tableau 4.3 les données en entrée des trois avions

Avions	X(NM)	Y(NM)	Altitude(ft)	Vitesse(m/s)	Cap	Incidence
A	80.	0.	32000.	260	180.°	0.°
B	-80.	0.	32000.	260	0.°	0.°
C	0.	-80.	32000	260	90.°	0.°

Cas où A >> B >> C : ce qui signifie que A est plus prioritaire que B et C, et que B est plus prioritaire que C.

Tableau 4.4 Résultats numériques pour le cas de 3avions(ordre A >> B >> C)

Avion	Début de déviation (sec)	Instant de retour(sec)	déviation horizontal	déviation vertical	Retard	Dist-Suppl (kms)	
A	Trajectoire originelle maintenue						
B	359	890	10°L	0°	13s	3.4367	1.10%
C	270	905	10°L	0°	29s	7.5370	2.42%
Moyenne des retards		14s	Nbre de conflits	3	Temps CPU	Moins de 1s	

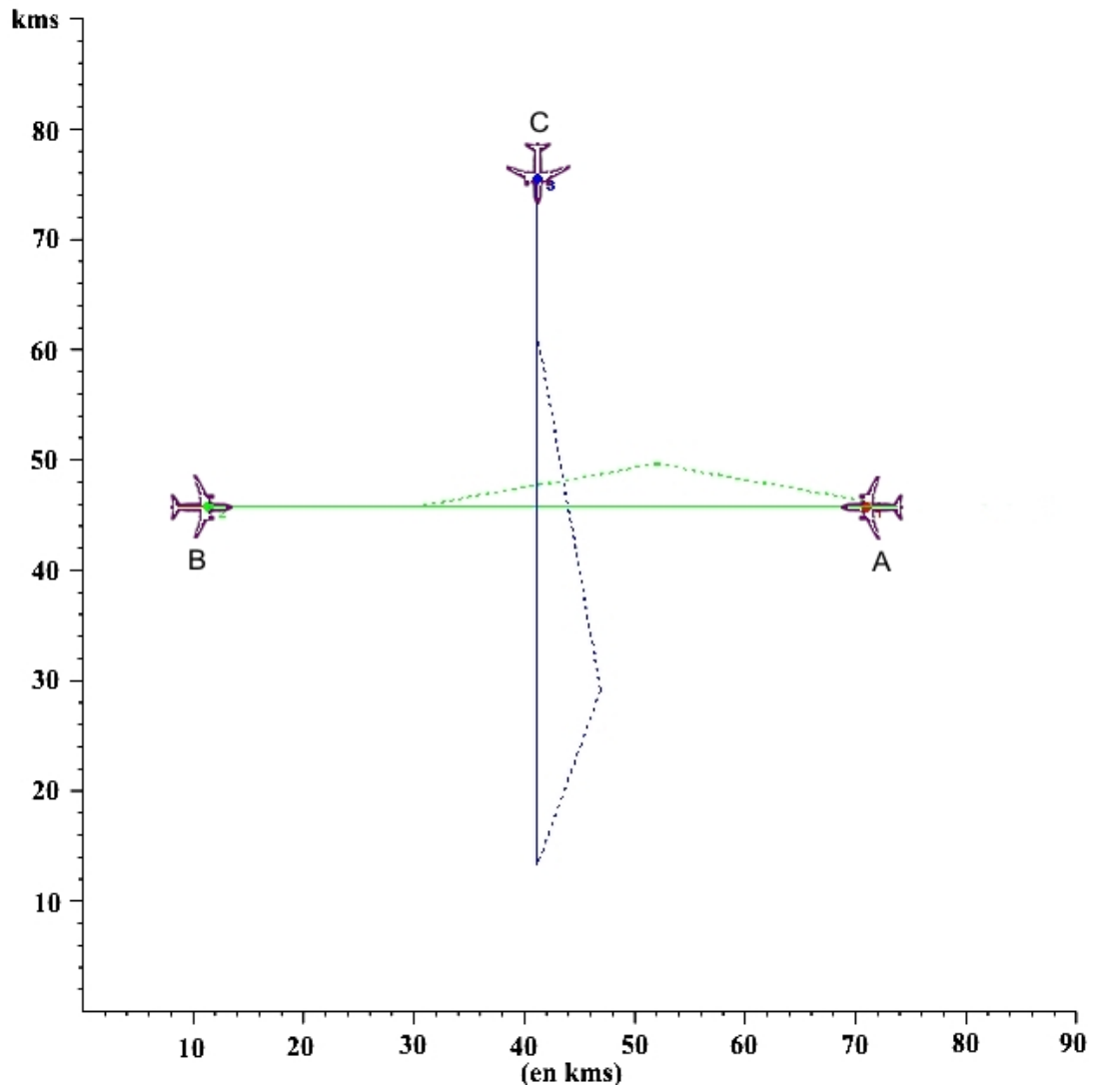


Figure 4. 2 Représentation graphique pour le cas de 3 avions (ordre A>>B>>C)

Cas où $A \gg C \gg B$: ce qui signifie que A est plus prioritaire que C et B, et que C est plus prioritaire que B.

Tableau 4.5 Résultats numériques pour le cas de 3 avions (ordre $A \gg C \gg B$)

Avion	Début de déviation (sec)	Instant de retour(sec)	déviatiion horizontal	déviatiion vertical	Retard	Dist-Suppl (kms)	
A	Trajectoire originelle maintenue						
B	270	866	10°L	0°	24s	6.3247	2.03%
C	270	830	10°L	0°	21s	5.4078	1.73%
<i>Moyenne des retards</i>		<i>15s</i>	<i>Nbre de conflits</i>		<i>3</i>	<i>Temps CPU</i>	<i>Moins de 1s</i>

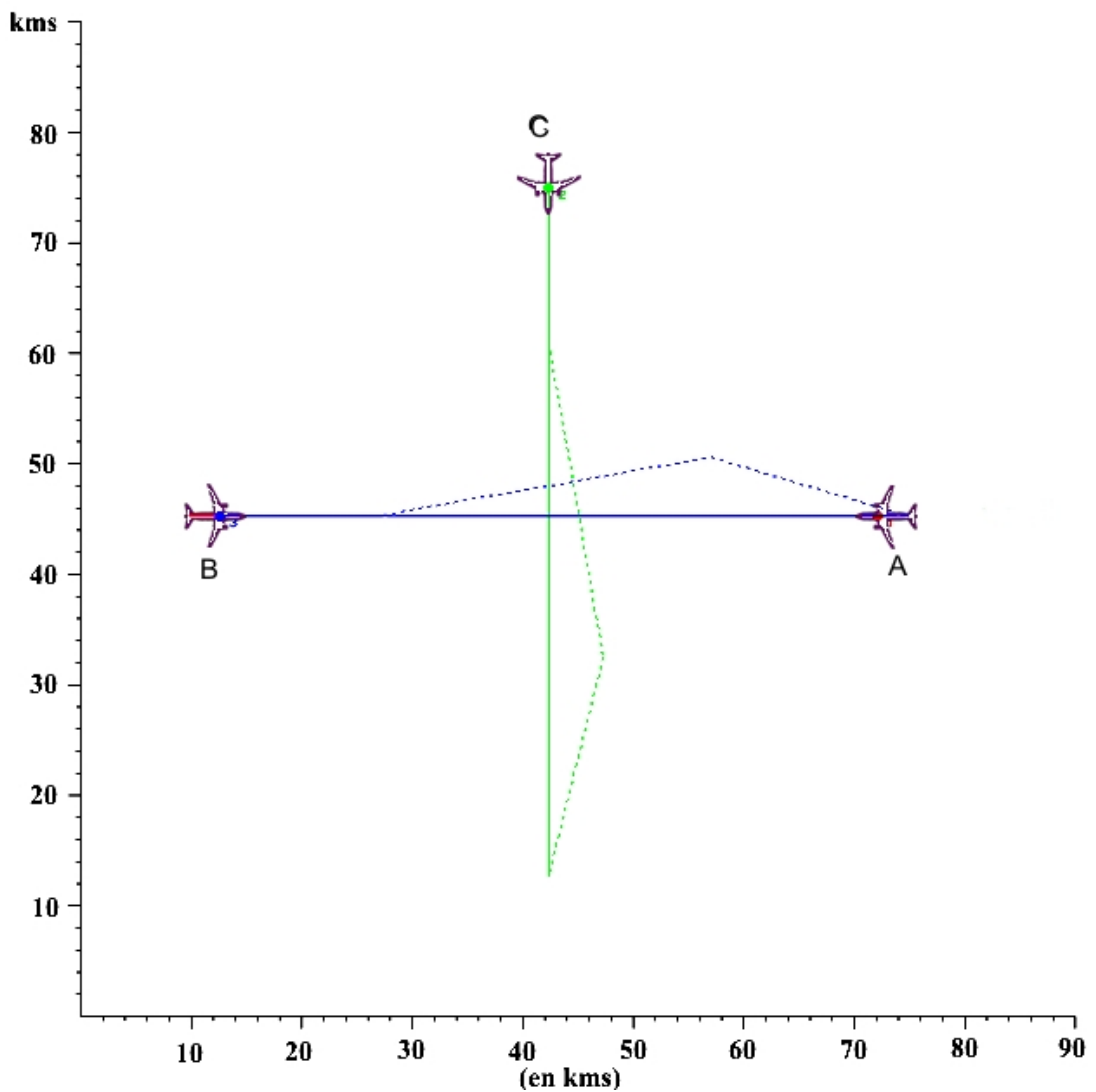


Figure 4. 3 Représentation graphique pour le cas de 3 avions (ordre $A \gg C \gg B$)

Cas où $B \gg A \gg C$: ce qui signifie que B est plus prioritaire que A et C, et que A est plus prioritaire que C.

Tableau 4.6 Résultats numériques pour le cas de 3avions (ordre $B \gg A \gg C$)

Avion	Début de déviation (sec)	Instant de retour(sec)	déviatiion horizontal	déviatiion vertical	Retard	Dist-Suppl (kms)		
A	359	431	10°R	0°	13s	3.4369	1.10%	
B	Trajectoire originelle maintenue							
C	270	587	10°R	0°	23s	6.0794	1.95%	
<i>Moyenne des retards</i>		<i>12s</i>	<i>Nbre de conflits</i>		<i>3</i>	<i>Temps CPU</i>		<i>Moins de 1s</i>

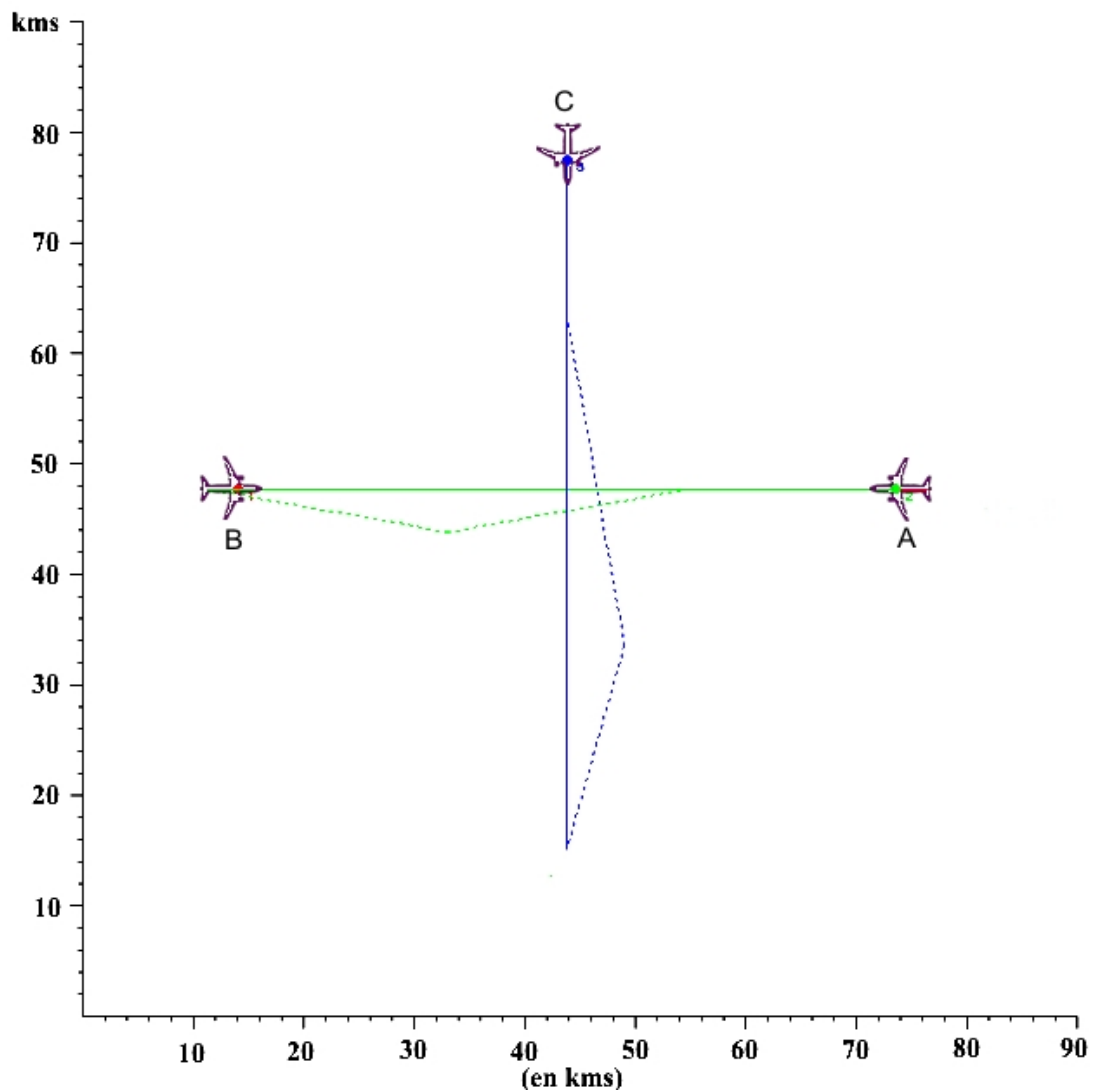


Figure 4. 4 Représentation graphique pour le cas de 3avions (ordre $B \gg A \gg C$)

Tableau 4.7 Récapitulatif des trois cas de figures

Cas de figures		Avion A	Avion B	Avion C	Moyenne des retards des avions déviés
<u>A >> B >> C</u>	Déviations	0°	10° Right	10° Right	
	Retards	0s	13 s	29 s	21s
<u>A >> C >> B</u>	Déviations	0°	10° Left	10° Left	
	Retards	0s	24s	21s	22s
<u>B >> A >> C</u>	Déviations	10° Right	0°	10° Right	
	Retards	13 s	0s	23 s	18s

En termes de moyennes des retards les cas de figures où **A >> B >> C** et où **B >> A >> C** sont relativement meilleure que celle où **A >> C >> B**. Dans ce cas de conflits de trois avions l'influence de l'ordonnancement sur la qualité de la solution reste faible.

3^{ème} CAS : Conflit de trois avions ayant des vitesses différentes et le même niveau de vol(FL320):

Tableau 4.8. Les données en entrée pour le cas de trois avions avec des vitesses différentes

Avions	X(NM)	Y(NM)	Altitude(ft)	Vitesse(m/s)	Cap	Incidence
A	80.	-90.	32000.	260	120.°	0.°
B	10.	-70.	32000.	240	60.°	0.°
C	-50.	-50.	32000	220	20.°	0.°

Tableau 4.9 Résultats numériques pour le cas de 3 avions ayant des vitesses différentes

Avion	Début de déviation (sec)	Instant de retour(sec)	déviatiion horizontal	déviatiion vertical	Retard	Dist-Suppl (kms)		
A	Trajectoire originelle maintenue							
B	98	865	10°L	0°	36s	8.7260	3.03%	
C	50	276	30°R	0°	37s	8.1280	3.08%	
<i>Moyenne des retards</i>		<i>24s</i>	<i>Nbre de conflits</i>		<i>3</i>	<i>Temps CPU</i>		<i>Moins de 1s</i>

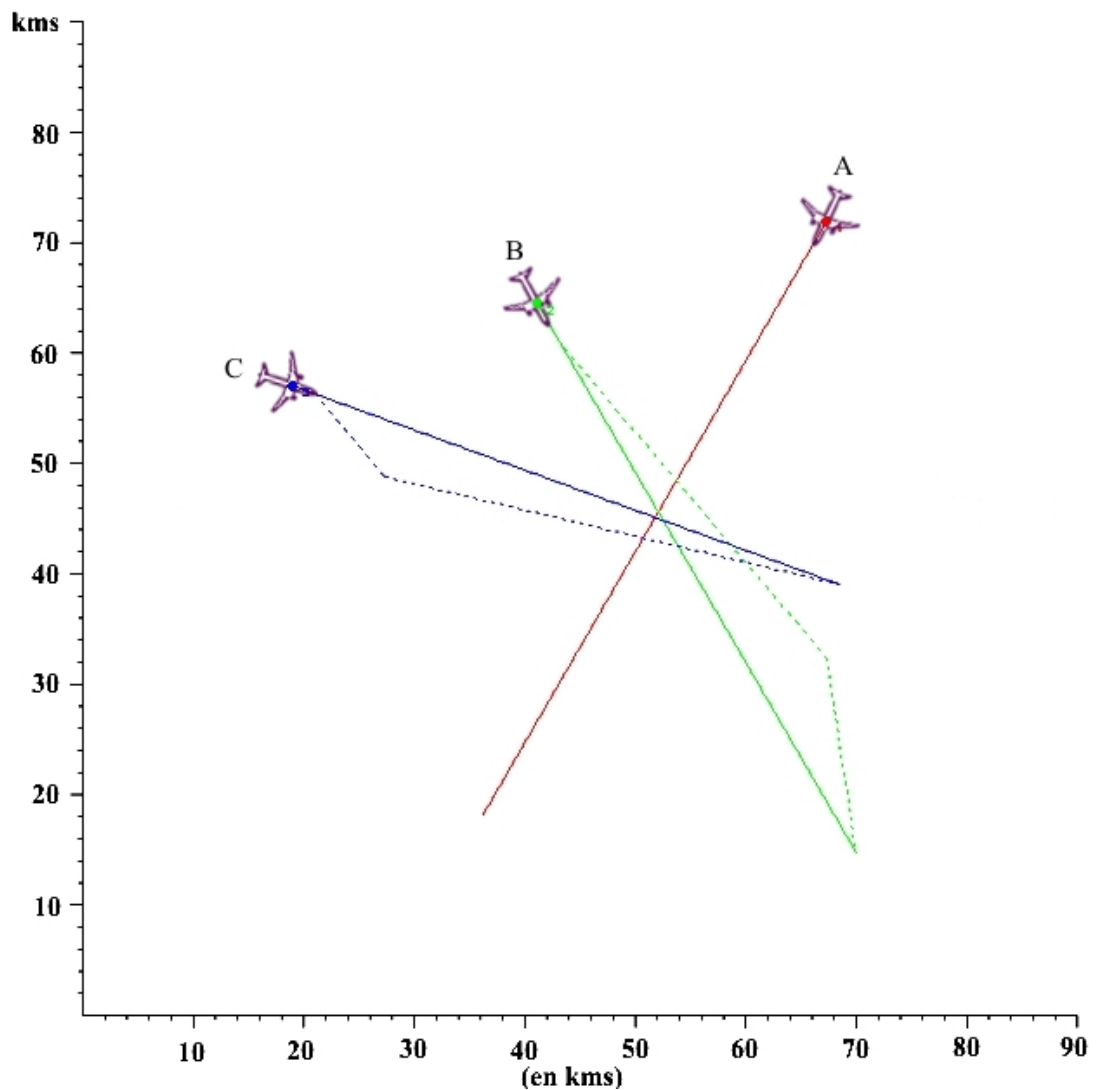


Figure 4.5 Représentation graphique du cas de conflits de 3 avions ayant des vitesses différentes et un même niveau de vol

4^{ème} CAS: Conflit de quatre avions ayant la même vitesse et le même niveau de vol(FL320):

Deux cas de figures sont présentés et qui permettent de constater l'importance quant au choix de l'ordre des avions (du prioritaire au moins prioritaire) et de son influence sur les résultats obtenus.

Tableau 4.10 Les données en entrée de cas de conflits de 4avions

Avions	X(NM)	Y(NM)	Altitude(ft)	Vitesse(m/s)	Cap	Incidence
A	90.	0.	32000.	260	180.°	0.°
B	-90.	0.	32000.	260	0.°	0.°
C	0.	-90.	32000	260	90.°	0.°
D	0.	90.	32000	260	-90°	0°

Cas où A >> B >> C >> D: ce qui signifie que A est plus prioritaire que B, C et D, et que B est plus prioritaire que C et D, et que C est plus prioritaire que D.

Tableau 4.11 Résultats numériques pour le cas de 4avions(ordre A>>B>>C>>D)

Avion	Début de déviation (sec)	Instant de retour(sec)	déviation horizontal	déviation vertical	Retard	Dist-Suppl (kms)		
A	Trajectoire originelle maintenue							
B	431	862	10°L	0°	15s	3.7915	1.22%	
C	342	842	20°R	0°	66s	17.2003	5.51%	
D	342	928	10.°L	0.°	27 s	6.9591	2.23%	
Moyenne des retards		27s	Nbre de conflits		6	Temps CPU	Moins de 1s	

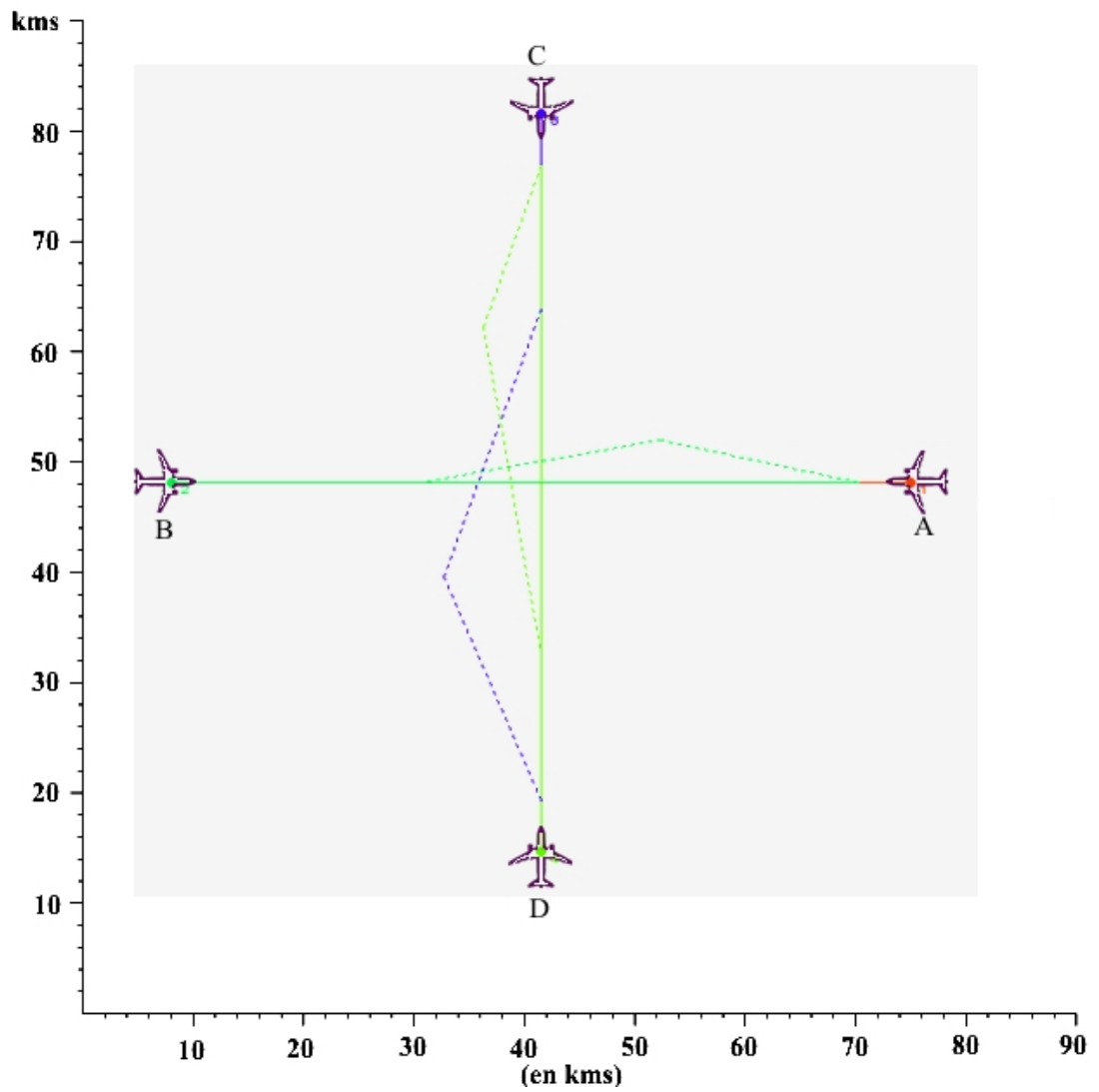


Figure 4.6 Représentation graphique pour le cas de 4 avions (ordre A>>B>>C>>D)

Cas où $D \gg B \gg C \gg A$: ce qui signifie que D est plus prioritaire que B, C et A, et que B est plus prioritaire que C et A, et que C est plus prioritaire que A.

Tableau 4.12 Résultats numériques pour le cas de 4avions (ordre $D \gg B \gg C \gg A$)

Avion	Début de déviation (sec)	Instant de retour(sec)	déviatiion horizontal	déviatiion vertical	Retard	Dist-Suppl (kms)	
A	342	599	10°R		29s	7.4288	2.38%
B	342	560	10°R		24s	6.1265	1.96%
C	342	595	10.°R		28 s	7.2798	2.33%
D	Trajectoire originelle maintenue						
<i>Moyenne des retards</i>		20s	<i>Nbre de conflits</i>		6	<i>Temps CPU</i>	Moins de 1s

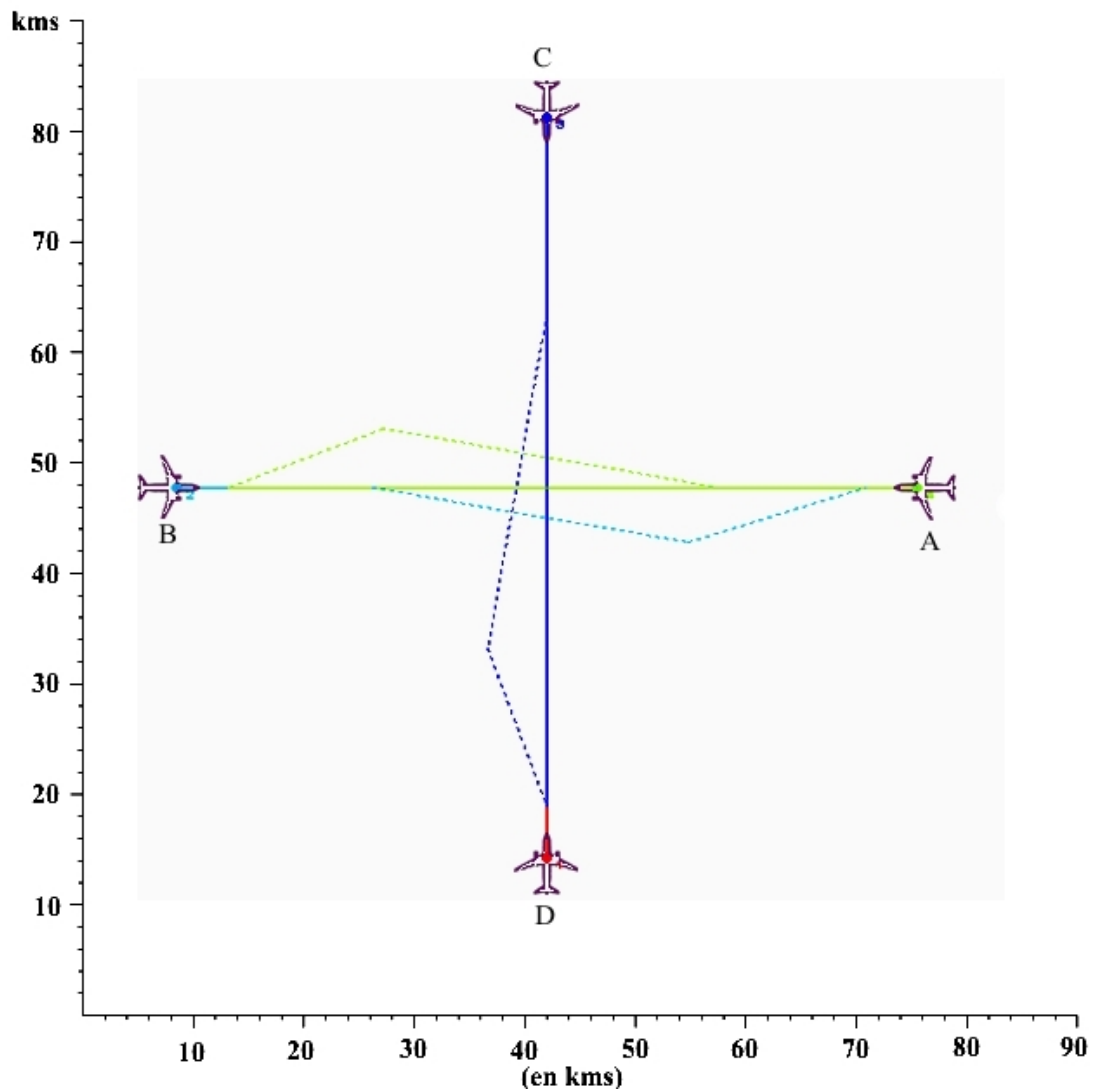


Figure 4.7 Représentation graphique pour le cas de 4avions(ordre D>>B>>C>>A)

Tableau 4.13 Récapitulatif des résultats pour le cas de 4avions

Cas de figures		Avion A	Avion B	Avion C	Avion D	Moyenne des retards des avions déviés
<u>A>>B>>C >>D</u>	Déviations	0°	10° Left	20° Right	10° Left	
	Retards	0s	15 s	66s	27s	36s
<u>D>>B>>C>>A</u>	Déviations	10°Right	10° Right	10° Right	0°	
	Retards	29s	24s	28s	0s	27s

Dans le cas où A>>B>>C >>D l'avion D qui est le moins prioritaire est pénalisé en terme de temps de retard par rapport aux avions plus prioritaire que lui, l'avion D accuse un retard plus que le double de l'avion B, par contre le deuxième cas de figure où D>>B>>C>>A les retards des avions déviés sont équivalents.

Donc l'influence de l'ordonnancement des avions paraît évidente dans ce cas de conflits de quatre avions que celui de 3 avions en conflit que nous avons vu précédemment.

5^{ème} CAS: Conflit de huit avions ayant la même vitesse et le même niveau de vol(FL320):

Tableau 4.14 Les données en entrée pour le cas de 8avions

Avions	X(NM)	Y(NM)	Altitude(ft)	Vitesse(m/s)	Cap	Incidence
A	80	-30	32000.	260	160.°	0.°
B	30.	-70.	32000.	260	100.°	0.°
C	-30.	-90.	32000	260	70.°	0.°
D	-70.	-40.	32000	260	60°	0°
E	-70	40	32000	260	-45°	0°
F	-30	90	32000	260	-60°	0°

G	30	60	32000	260	-120°	0°
H	100	30	32000	260	-150°	0°

Tableau 4.15 Résultats numériques pour le cas de 8 avions

Avion	Début de déviation (sec)	Instant de retour(sec)	déviati horizontal	déviati vertical	Retard	Dist-Suppl (kms)		
A	Trajectoire originelle maintenue							
B	29	779	10°L		31s	7.9513	2.55%	
C	61	121	30°R		8s	2.1976	0.70%	
D	100	160	20.°R		4s	.9933	0.32%	
E	Trajectoire originelle maintenue							
F	115	175	30°R		8s	2.2032	0.71%	
G	Trajectoire originelle maintenue							
H	31	778	10°R		30s	7.8900	2.53%	
Moyenne des retards		10s	Nbre de conflits		28	Temps CPU		5s

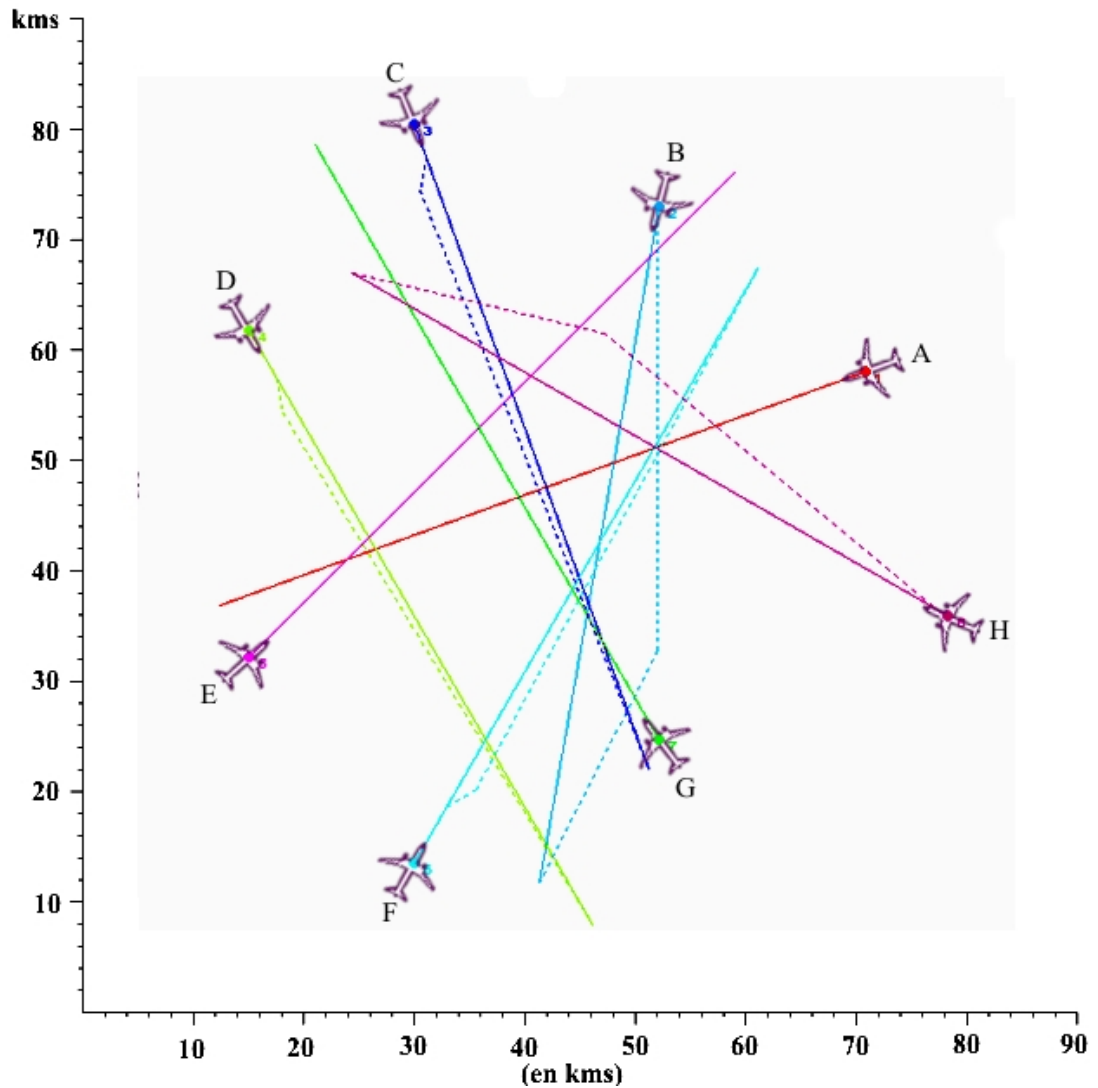


Figure 4.8 Représentation graphique pour le cas de 8 avions

Les résultats montrent que plus le nombre d'avions en conflits augmente, plus le temps d'exécution (temps de réponse croit rapidement), pour ce cas de 8 avions, il est de l'ordre de 5 secondes et qui reste toujours acceptable.

6^{ème} CAS : Conflit de huit avions ayant différents niveaux de vol et de vitesses:

Tableau 4.16 Les données en entrée pour le cas de 8 avions avec des vitesses différentes

Avions	X(NM)	Y(NM)	Altitude(ft)	Vitesse(m/s)	Cap	Incidence
A	80	-30	32000.	260	160.°	0.°
B	30.	-70.	31000.	240	100.°	0.°

C	-30.	-90.	33000	230	70.°	0.°
D	-70.	-40.	32000	220	60°	0°
E	-70	40	32000	260	-45°	0°
F	-30	90	33000	240	-60°	0°
G	30	60	31000	250	-120°	0°
H	100	30	30000	260	-150°	0°

Tableau 4.17 Résultats numériques pour le cas de 8 avions avec des vitesses différentes

Avion	Début de déviation (sec)	Instant T1(sec)	Instant T2(sec)	déviatiion horizontal	déviatiion vertical	Retard	Dist-Suppl (kms)	
A	Trajectoire originelle maintenue							
B	Trajectoire originelle maintenue							
C	57	117		30°R		8s	1.9437	0.70%
D	102	112	1192		30.°C	2s	.4400	0.17%
E	Trajectoire originelle maintenue							
F	130	190		30°R		8s	2.0353	0.71%
G	Trajectoire originelle maintenue							
H	31	778		10°R		30s	7.8900	2.53%
Moyenne des retards		6s	Nbre de conflits		28	Temps CPU		5s

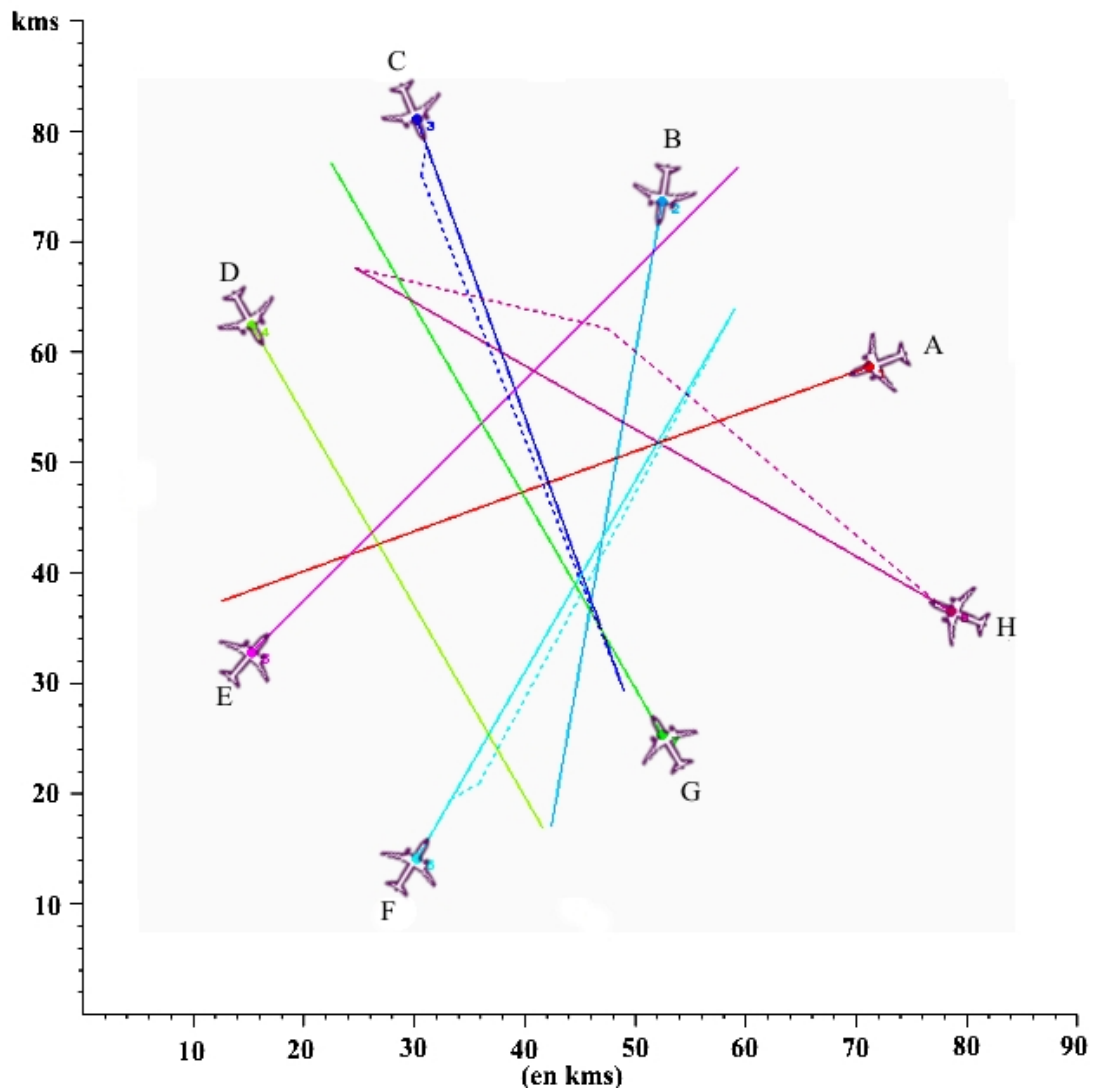


Figure 4.9 Représentation graphique pour le cas de 8 avions avec des vitesses différentes

En comparaison avec le 5^{ème} cas, nous constatons que le nombre de conflits détectés est identique, alors que le nombre d'avions dévié n'est pas le même, c'est le cas de l'avion B qui était dévié de sa trajectoire dans le 5^{ème} cas et pas dans ce cas là. Ceci s'explique par le fait que la trajectoire originelle de l'avion B représente la trajectoire optimale respectant les normes de séparation, et que la déviation de l'avion B n'est pas nécessaire, malgré le conflit détecté.

7^{ème} CAS : Conflit de quatorze avions ayant meme niveau de vol et différentes vitesses:

Tableau 4.18 Les données en entrée pour le cas de 14avions

Avions	X(NM)	Y(NM)	Altitude(ft)	Vitesse(m/s)	Cap	Incidence
A	80	-30	32000.	240	180°	0.°
B	80	-25	32000.	260	180°	0.°
C	60	-30	32000	240	180°	0.°
D	30	-70	32000	220	100°	0°
E	-30	-90	32000	260	70°	0°
F	-70	-40	32000	240	60°	0°
G	-70	40	32000	240	0°	0°
H	-70	35	32000	240	0°	0°
I	-30	90	32000	240	-60°	0.°
J	25	60	32000	260	-120°	0°
K	70	60	32000	240	-120	0°
L	30	55	32000	220	-120°	0°
M	100	30	32000	240	-150°	0°
N	80	50	32000	240	-170°	0°

Pour cet exemple, nous avons lancé deux simulations, dans la première simulation, le solveur effectue une résolution de conflits seulement sur le plan horizontal, où nous avons constatés certains cas de conflits non résolus, dans la deuxième simulation, nous avons pu constaté que tous les conflits ont été résolus. Ce cas là montre que la résolution dans le plan horizontal est insuffisante pour résoudre tous les conflits détectés.

Cas où la résolution sur le plan vertical est inactive

Tableau 4.19 Résultats numériques pour le cas de 14avions (1^{er} simulation)

Avion	Début de déviation (sec)	Instant de retour(sec)	déviation horizontal	déviation vertical	Retard	Dist-Suppl (kms)		
A	Trajectoire originelle maintenue							
B	Trajectoire originelle maintenue							
C	Trajectoire originelle maintenue							
D	2	519	30.°L		113	24.7607	9.38%	
E	Trajectoire originelle maintenue							
F	Trajectoire originelle maintenue							
G	Trajectoire originelle maintenue							
H	Trajectoire originelle maintenue							
I	Trajectoire originelle maintenue							
J	24	920	20°R		173s	45.0609	14.44%	
K	Trajectoire originelle maintenue							
L	Pas de solution							
M	2	779	30°L		232s	55.7261	19.35%	
N	2	802	30°L		246s	59.1285	20.53%	
Moyenne des retards		55s	Nbre de conflits		91	Temps CPU		5s
Conflits non résolus		1	<u>Avion K</u> avec <u>Avion L</u>					

Cas où la résolution sur le plan vertical est active

Tableau 4.20 Résultats numériques pour le cas de 14avions (2^{ème} simulation)

Avion	Début de déviation (sec)	Instant T1(sec)	Instant T2(sec))	déviation horizontal	déviation vertical	Retard	Dist-Suppl (kms)	
A	Trajectoire originelle maintenue							
B	Trajectoire originelle maintenue							
C	Trajectoire originelle maintenue							
D	2	519		30.°L		113	24.7607	9.38%
E	Trajectoire originelle maintenue							
F	Trajectoire originelle maintenue							
G	Trajectoire originelle maintenue							
H	Trajectoire originelle maintenue							
I	Trajectoire originelle maintenue							
J	24	920		20°R		173s	45.0609	14.44%
K	Trajectoire originelle maintenue							
L	2	12	1190		10°D	0s	0	0%
M	2	779		30°L		232s	55.7261	19.35%
N	2	802		30°L		246s	59.1285	20.53%
Moyenne des retards		55s	Nbre de conflits		91	Temps CPU		5s

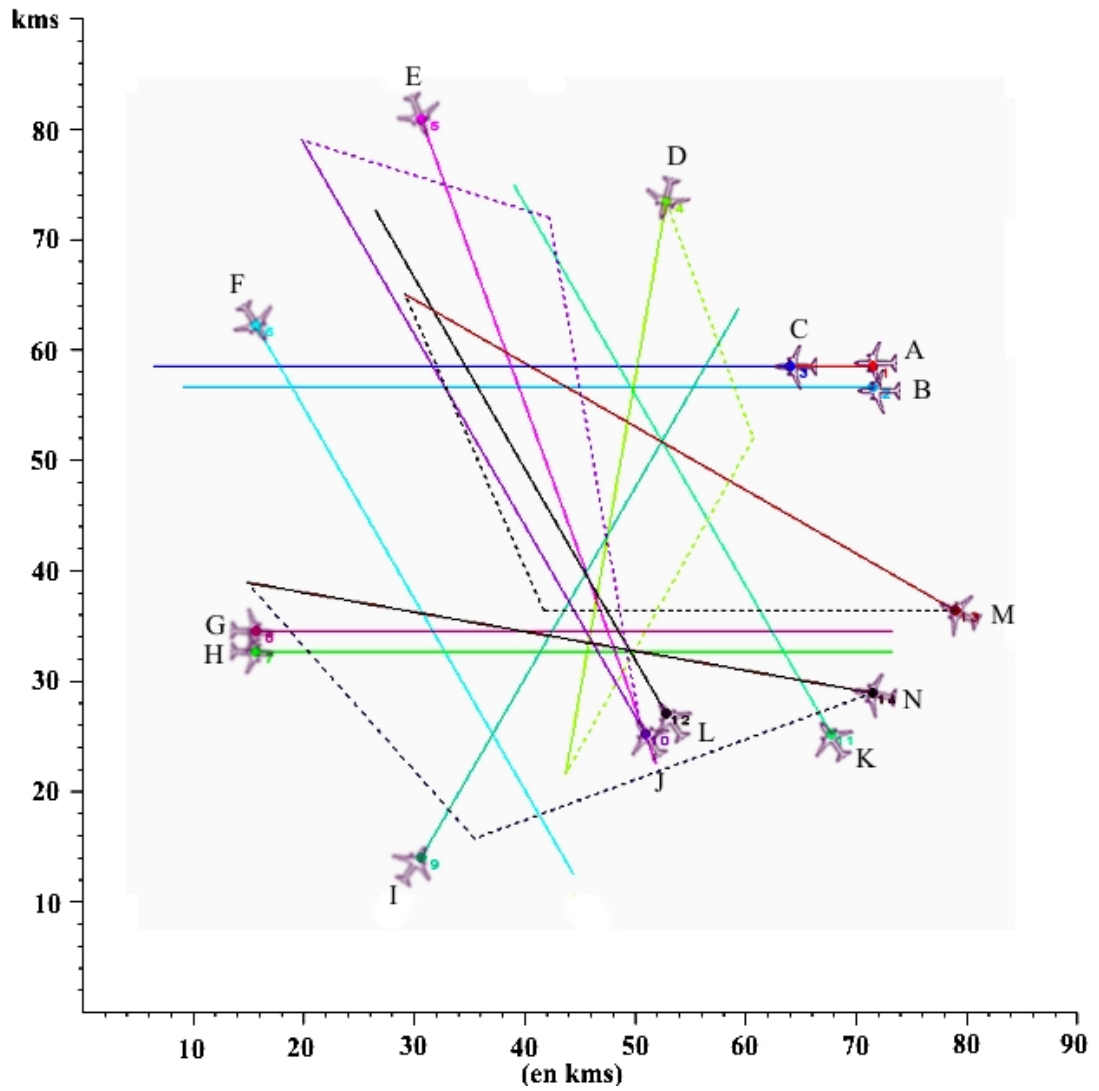


Figure 4.10 Représentation graphique pour le cas de 14 avions

REMARQUE :

Notons bien que tous les cas que nous avons présenté jusque là, sont exécutés avec un *pas de temps d'1 seconde.*

8^{ème} CAS : les même données que « 7^{ème} cas » avec un pas de temps de 10 secondes :

nous avons pris le même exemple que « 7^{ème} cas » que nous avons exécuté avec le pas de temps de 10 secondes, ce qui va nous permettre de vérifier si les réponses sont identiques ou pas avec deux pas de temps différents.

Tableau 4.21 Résultats numériques pour le cas de 14avions (pas de temps de 10s)

Avion	Début de déviation (sec)	Instant T1 (sec)	Instant T2 (sec)	déviation horizontal	déviation vertical	Retard	Dist-Suppl (kms)	
A	Trajectoire originelle maintenue							
B	Trajectoire originelle maintenue							
C	Trajectoire originelle maintenue							
D	20	530		30°L		111s	24.4468	9.26%
E	170	230		10°R		1	0.2515	.08%
F	160	260		20°R		7s	1.5957	.55%
G	Trajectoire originelle maintenue							
H	Trajectoire originelle maintenue							
I	Trajectoire originelle maintenue							
J	30	920		20°R		172	44.6313	14.30%
K	Trajectoire originelle maintenue							
L	20	30	1190		10°D	0s		0%
M	20	790		30°L		232s	55.5667	19.29%
N	20	810		30°L		244s	58.5468	20.33%
Moyenne des retards		55s	Nbre de conflits		91	Temps CPU	1s	

D'après ces résultats, nous constatons clairement que les réponses de résolution de conflits sont quasi équivalentes à celles avec le pas de temps d'1 seconde. Le seul constat que nous puissions faire est que le temps d'exécution diminue d'une manière très nette avec le pas de temps plus grand .

9^{ème} CAS : conflit de 32 avions : (pas = 1 sec)

Tableau 4.22 Les données en entrée pour le cas de 32avions

Avions	X(NM)	Y(NM)	Altitude(ft)	Vitesse(m/s)	Cap	Incidence
A	100	0	32000.	260	180°	0.°
B	70	70	32000.	260	-132°	0.°
C	0	100	32000	260	-90°	0.°
D	-70	70	32000	260	-52°	0°
E	-100	0	32000.	260	0°	0.°
F	-70	-70	32000.	260	52°	0.°
G	0	-100	32000	260	90°	0.°
H	70	-70	32000	260	132°	0°
I	95	20	32000	260	-168°	0°
J	55	85	32000	260	-120°	0°
K	-20	95	32000	260	-88°	0°
L	-85	60	32000	260	-40°	0°
M	-95	-20	32000	260	16°	0.°
N	-55	-85	32000	260	64°	0°
O	20	-95	32000	260	96°	0°
P	85	-60	32000	260	144°	0°
Q	90	40	32000	260	-156°	0°
R	35	90	32000	260	-108°	0°
S	-35	90	32000.	260	-76°	0.°
T	-90	40	32000.	260	-28°	0.°
U	-90	-40	32000	260	28°	0.°
V	-35	-90	32000	260	76°	0°
W	45	-90	32000	260	108°	0°
X	90	-40	32000	260	156°	0°
Y	85	60	32000	260	-144°	0°
Z	20	95	32000	260	-96°	0°
A1	-55	85	32000	260	-64°	0.°
B1	-95	20	32000	260	-16°	0°
C1	-85	-60	32000	260	40°	0°
D1	-20	-95	32000	260	88°	0°
E1	55	-85	32000	260	120°	0°
F1	95	-20	32000	260	168°	0°

Tableau 4.23 Résultats numériques pour le cas de 32avions (pas de temps de 1s)

Avion	Début de déviaton (sec)	Instant T1(sec)	Instant T2(sec)	déviaton horizontal	déviaton vertical	Retard	Dist-Suppl (kms)		
A	Trajectoire originelle maintenue								
B	143	899		20°R		131s	34.1680	10.95%	
C	180	240		30°L		9s	2.2108	0.71%	
D	Trajectoire originelle maintenue								
E	133	861		20°R		118s	30.6124	9.81%	
F	133	867		30°R		233s	60.6683	19.44%	
G	232	421		30°L		9s	2.2386	0.72%	
H	143	801		30°L		189s	49.0995	15.74%	
I	2	640		30°L		160s	41.6006	13.33%	
J	2	224		30°R		36s	9.3207	2.99%	
K	Trajectoire originelle maintenue								
L	Trajectoire originelle maintenue								
M	Trajectoire originelle maintenue								
N	2	337		30°R		60s	15.6529	5.02%	
O	Trajectoire originelle maintenue								
P	2	12	1190		10°D	0s	0	0%	
Q	2	12	1190		10°D	0	0	0	
R	2	207		30°R		33s	8.4759	2.72%	
S	2	759		30°L		220s	57.309	18.37%	
T	Trajectoire originelle maintenue								
U	2	12	1190		10°C	0	0	0%	
V	2	211		30°R		33s	8.6724	2.78%	
W	2	691		30°L		184s	47.8405	15.33%	
X	2	910		30°L		323s	83.9379	26.90%	
Y	Trajectoire originelle maintenue								
Z	Trajectoire originelle maintenue								
A1	2	714		30°L		196s	50.8887	16.31%	
B1	2	12	1191		20°D	1s	.2600	.08%	
C1	2	12	1190		10°D	0s	0	0%	
D1	2	12	1190		10°C	0s	0	0%	
E1	Trajectoire originelle maintenue								
F1	2	12	1190		10°C	0s	0	0%	
Moyenne des retards		60s		Nbre de conflits		493	Temps CPU		45s

Dans ce cas de figure, on constate que pour un nombre aussi important d'avions (32avions) qui se trouvent au même niveau de vol, et qui convergent tous vers le même point central avec une même vitesse (cas extrême de conflits), avec l'ordonnancement d'avions établi, que les résultats obtenues avec le temps CPU de 45 secondes, et une moyenne des retards des avions devies de l'ordre de 1min 28sec, sont très appréciables et très satisfaisants.

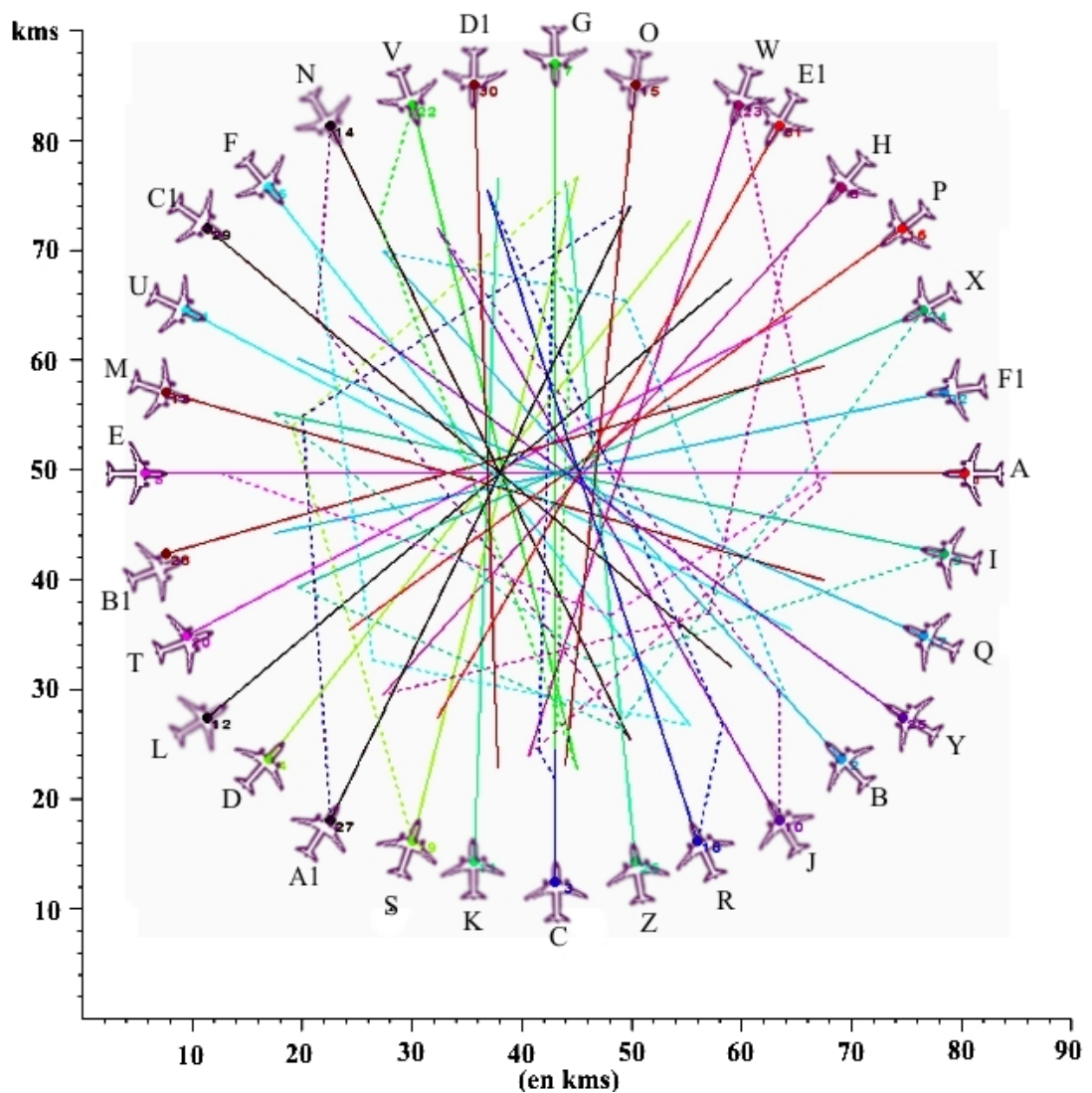


Figure 4.11 Représentation graphique pour le cas de 32avions (pas de temps de 1s)

10^{ème} CAS : conflit de 32 avions : (pas = 10 sec)

Dans ce cas, nous avons pris le même fichier de données d'avions que précédemment, nous constatons que :

- les résultats sont très différentes pour les deux pas de temps fixés(1sec et 10sec).
- le temps CPU qui est de 45s pour le pas=1sec est de 4sec pour le pas=10sec
- les moyennes des retards sont équivalentes à 14sec près.

Tableau 4.24 Résultats numériques pour le cas de 32 avions (pas de temps de 10s)

Avion	Début de déviation (sec)	Instant T1(sec)	Instant T2(sec)	déviati on horizontal	déviati on vertical	Retard	Dist-Suppl (kms)		
A	Trajectoire originelle maintenue								
B	150	910		20°R		135s	35.0160	11.22%	
C	180	240		30°L		9s	2.2108	0.71%	
D	Trajectoire originelle maintenue								
E	140	870		20°R		120s	31.1224	9.98%	
F	140	880		30°R		239s	62.1628	19.92%	
G	340	400		30°L		9s	2.2347	0.72%	
H	150	810		30°L		191s	49.7109	15.93%	
I	20	810		10°L		35s	9.0386	2.90%	
J	20	840		30°L		264s	68.5340	21.97%	
K	Trajectoire originelle maintenue								
L	20	640		30°L		154s	40.0100	12.82%	
M	20	30	1190		10°D	0s	0	0%	
N	20	360		30°R		62s	16.0475	5.14%	
O	Trajectoire originelle maintenue								
P	20	30	1190		10°D	0s	0	0%	
Q	20	110		30°R		13s	3.3738	1.08%	
R	20	950		30°L		347s	90.2500	28.93%	
S	20	80		30R		8s	2.1937	0.70%	
T	Trajectoire originelle maintenue								
U	20	30	1190		10°C	0s	0	0%	
V	20	270		30°R		42s	10.8077	3.46%	
W	20	700		30°L		182s	47.2967	15.16%	
X	20	940		30°L		339s	88.0715	28.23%	
Y	20	80		30°R		8s	2.1937	0.70%	
Z	20	30	1191		20°D	1s	0.2600	0.08%	
A1	20	730		30°L		197s	51.3171	16.45%	
B1	20	30	1191		20°C	1s	0.2600	0.08%	
C1	20	30	1190		10°D	0s	0	0%	
D1	20	30	1190		10°C	0s	0	0%	
E1	Trajectoire originelle maintenue								
F1	20	30	1190		10°C	0s	0	0%	
Moyenne des retards		74s	Nbre de conflits		496	Temps CPU	4s		

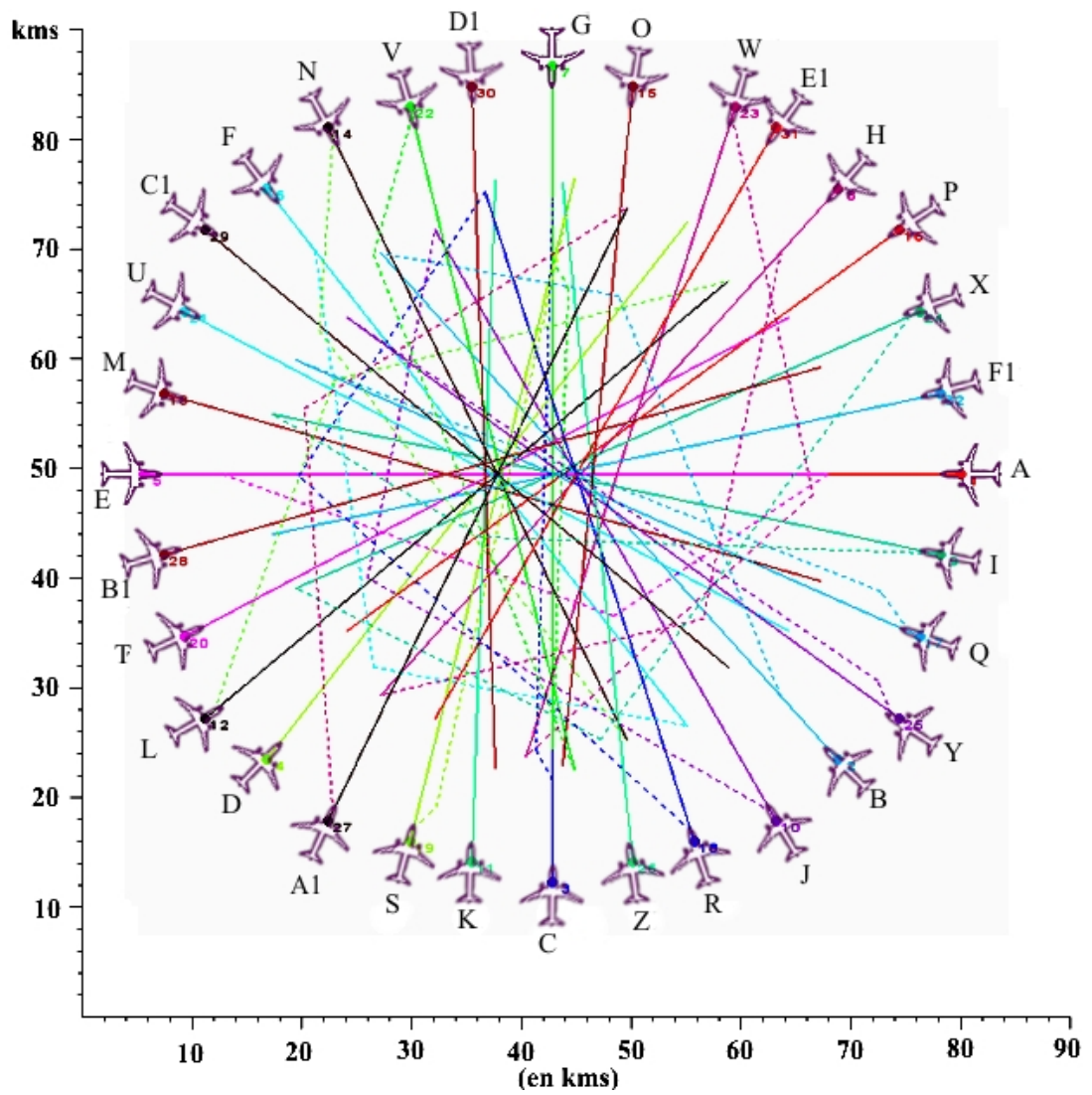


Figure 4.12 Représentation graphique pour le cas de 32 avions (pas de temps de 10s)

CONCLUSION

Dans ce document sont exposées les techniques telles que, les méthodes d'intervalles qui ont montré leurs limites, en termes de temps de calcul, que ce soit dans les systèmes centralisés ou les systèmes embarqués, et les algorithmes génétiques qui semblent être très performants, mais, préconisés plutôt pour les systèmes centralisés, ce qui n'est pas notre cas.

Pour cela, l'algorithme A* a été adopté dans l'étude du problème de conflits aériens.

L'objectif de ce travail consiste donc, à réaliser un code de calcul pour la détection et la résolution de conflits aériens dans le cadre des systèmes embarqués.

L'intégration de ce code dans le système d'avion, rendra ce dernier autonome pour prendre en charge la détection et la résolution de conflits.

Les trajectoires originelles des avions considérées dans ce code de calcul, sont traitées dans l'intervalle de temps de simulation, qui est de l'ordre de 20 minutes, comme étant des segments de droites. Les vitesses des avions pendant ce temps de simulation, sont considérées constantes.

La prise en compte des manœuvres d'évitement aussi bien dans le plan horizontal que dans le plan vertical, nous a permis de trouver des solutions dans les cas de conflits où la résolution dans le plan horizontal n'existe pas.

Le traitement de cas de conflits complexes par ce code de calcul, montre que les solutions trouvées semblent très satisfaisantes, vu la qualité des résultats et la rapidité du calcul.

REFERENCES

1. Nicolas Barnier, « **Application de la programmation par contraintes à des problèmes de gestion du trafic aérien** », Thèse doctorat informatique de l'INP Toulouse (2002/12/06) .
2. Géraud Granger, « **Détection et résolution de conflits aériens : modélisations et analyse** », Thèse doctorat informatique de l'Ecole Polytechnique (2002/12/01)
3. Pierre Dodin, « **Résolution de conflits via la programmation semi-définie** », Rapport de DEA "mathématiques appliquées" de Paris VI (1999/09/30)
4. Frédéric Médioni, « **Méthode d'optimisation pour l'évitement aérien: systèmes centralisés, systèmes embarqués** », Thèse doctorat informatique de l'Ecole Polytechnique (1998/10/8)
5. Géraud Granger, « **Résolution de conflits embarquée dans les espaces de faible densité** », Rapport de DEA IFP (1998/09/29)
6. Jean-Marc Alliot, « **Techniques d'optimisation stochastique appliquées aux problèmes du contrôle aérien** », Thèse d'habilitation INPT. (1996/05/28)
7. Nicolas Durand, « **Optimisation de trajectoires pour la résolution de conflits** », Thèse doctorat informatique INPT. (1996/05/28)
8. Gervais Gaudière, « **Résolution de Conflits Aériens par Réseaux de Neurones** », Rapport DEA IFP. (1995/07/01)
9. Olivier Chansou, « **Utilisation des Algorithmes Génétiques pour la Résolution de conflits aériens** », Rapport DEA IFP. (1995/07/01)
10. Frédéric Médioni, « **Optimisation de la résolution de conflits par algorithmes stochastiques et programmation linéaire** », Rapport DEA, ENAC, (1994/07/01)
11. Nicolas Alech, « **Construction de trajectoires d'évitement optimales par algorithmes génétiques** », Rapport DEA, ENAC, (1994/07/01)
12. Jean-Marc Alliot, D. Colin, « **L'ATM 20ans d'efforts et perspectives** », CENA, journée d'étude vers l'automatisation du vol et de sa gestion, Toulouse – ONERA - (Nov. 2003)