

UNIVERSITE SAAD DAHLEB DE BLIDA

Faculté des Sciences de l'Ingénieur

Département d'Aéronautique

MEMOIRE DE MAGISTER

(En Aéronautique)

Spécialité : Aéronautique

DETECTION ET RESOLUTION DE CONFLITS PAR LA TECHNIQUE DES ALGORITHMES A*(Astar)

Par

OTMANI Abouelkacem

Devant le jury composé de :

ZERGOUG Mourad	Directeur de recherche, CSC	Président
BOUKRAA Salah	Professeur, U de Blida	Rapporteur
REZZOUG Taher	Maître de conférences, U de Blida	Examineur
LAGHA Mohand	Maître de conférences(B), U de Blida	Examineur
MEGUELATI Smain	Chargé de cours, U de Blida	Examineur
DJATOUF Abdelouhab	Master ENAC, ENNA	Invité

La soutenance aura lieu à 09:30, le 4 février 2010 à la salle de conférence,
département aéronautique

Le publique est cordialement invité

BLIDA, Décembre 2009

TABLE DE MATIERES

RESUME	
ABSTRACT	
REMERCIEMENTS	
LISTE DES FIGURES	
LISTE DES TABLEAUX	
INTRODUCTION.....	17
CHAPITRE 1 : GENERALITES SUR L'ESPACE AERIEN.....	19
1.1 Introduction	19
1.2 Définitions	20
1.3 La gestion des flux du trafic aérien	23
1.3.1 Le filtre ATFM stratégique	24
1.3.2 Le filtre ATFM pré-tactique	24
1.3.3 Le filtre ATFM Tactique	24
1.3.4 Le filtre ATC.....	24
1.3.5 Le filtre d'urgence	25
1.4 La gestion de l'espace aérien	25
1.5 Division de l'espace aérien	29
1.5.1 Espace aérien contrôlé	29
1.5.2 Espace aérien non contrôlé	32
1.6 Division verticale de l'espace aérien.....	33
1.6.1 Espace aérien inférieur.....	33
1.6.2 Espace aérien supérieur.....	33
1.7 Classification des espaces aériens.....	34
1.8 Espace à statut particulier.....	35

1.8.1 Zones interdites	35
1.8.2 Zones réglementées	35
1.8.3 Zones dangereuses	36
1.9 Les services du contrôle de la circulation aérienne.....	36
1.9.1 Le contrôle d'aérodrome	37
1.9.2 Le contrôle d'approche	37
1.9.3 Le contrôleur en route.....	38
CHAPITRE 2 : ETAT DE L'ART	41
2.1 Introduction	41
2.2 Améliorer l'écoulement du trafic.....	42
2.2.1 L'organisation de l'espace et des flux	42
2.2.2 Le contrôle aérien	42
2.3 Différentes approches pour l'aide à la résolution	42
2.3.1 Approche anti-calculatoire	42
2.3.2 Automatisation centralisée complète	43
2.3.3 Délégation de tâches	43
2.3.4 Approche autonome	43
2.4 Analyse biographique des approches du problème de résolution de conflits	44
2.4.1 Approches opérationnelles	44
2.4.2 Approches théoriques.....	49
CHAPITRE 3 : METHODES ET TECHNIQUES DE RESOLUTION DE CONFLITS AERIEN.....	54
3.1 Introduction	54
3.2 Les Algorithmes génétiques.....	56
3.2.1 Principes généraux	56
3.2.2 Description de la résolution par algorithmes génétiques	58
3.3 Méthode des intervalles	63
3.3.1 Le principe	64
3.3.2 Evitement par point tournant.....	65
3.3.3 Evitement par offset.....	72
3.4 Les Algorithmes A* (AStar)	74
CHAPITRE 4 : L'ALGORITHME A* (ASTAR).....	76

4.1 Introduction	76
4.2 Les étapes intermédiaires de la création du A*	77
4.2.1 L'algorithme de Dijkstra et l'algorithme « Best-First-Search »	77
4.2.2 L'algorithme A*	81
4.3 L'heuristique	82
4.3.1 L'heuristique pour une carte de grille.....	82
4.4 Etude d'application de l'algorithme A*	85
4.4.1 Zone de recherche.....	85
4.4.2 La notion de recherche	86
4.4.3 Continuer la recherche	88
4.4.4 Récapitulatif de la méthode A* (algorithme)	92
4.4.5 Simulation de l'algorithme A* (AStar).....	93
CHAPITRE 5 : SIMULATIONS ET COMMENTAIRES	99
5.1 Introduction	99
5.2 Les manœuvres utilisées	100
5.3 Détection des conflits.....	101
5.4 L'Algorithme A*	102
5.4.1 Arbre de recherche	104
5.4.2 L'évitement	104
5.4.3 Développement d'un nœud de l'arbre.....	105
5.4.4 Exemple de développement d'un arbre pour un avion.....	106
5.4.5 Trajectoires d'évitement discrétisables	107
5.5 Exemple d'arbre de recherche	108
5.6 Simulation d'application de l'algorithme A* au cas de conflit aérien	112
5.6.1 Cas de deux avions	112
5.6.2 Cas de n avions.....	128
CONCLUSION	142
APPENDICE A	
APPENDICE B	
BIBLIOGRAPHIE	

RESUME

Le problème d'optimisation de trajectoires pour la résolution de conflits aériens en route reste aujourd'hui un problème ouvert et ce a cause de l'augmentation du trafic aérien qui génère des problèmes très complexes pour le contrôle aérien.

Dans ce travail l'approche principale est de considérer le problème de conflits entre avions comme un problème d'optimisation avec contraintes en utilisant un algorithme d'intelligence artificielle A* (AStar).

Pour cela, deux codes de calculs sont élaborés, le premier représente la simulation de l'algorithme A* pour la génération des trajectoires de coûts minimum appliqué sur une case de grilles, le deuxième code de calcul pour la détection et la résolution du conflit aérien par l'Algorithme A* (AStar) en utilisant la manœuvre « point tournant » qui est la plus facile à transmettre au pilote.

Ce travail peut être aussi considéré comme une revue des travaux déjà étudiés dans le domaine de résolution de conflits, ainsi que des différentes méthodes et techniques de résolutions qui existent.

ABSTRACT

The problem of optimisation of plane trajectories for solving en route air traffic conflicts is still an open problem essentially because of the increase of air traffic which generates complex problems for control.

In this work, the principal approach considers conflicts between planes as an optimization problem with constraints using an algorithm of artificial intelligence which is the A* algorithm.

For this, two codes have been developed, the first one is a simulation by the A* algorithm for generating trajectories with minimal cost applied for grids, the second one is for detecting and solving of air traffic conflict using the A* algorithm and the "rotating point method" which is the most simple one for the pilot.

This work can also be considered as a review of the different works which studied the problem of solving conflicts, and as a presentation of the methods and techniques that exist.

ملخص

إن مثالية الطريق الجوي تعتبر مشكلة عند إيجاد حلول للتصادم في الطريق الجوي، الذي يبقى إشكال مفتوح بسبب الزيادة في الحركة الجوية التي تولد مشاكل معقدة كثيرا في المراقبة الجوية.

في هذا العمل المبدأ الرئيسي هو اعتبار مشكلة التصادم بين الطائرات كمشكلة للمثالية بشروط و ذلك باستعمال خوارزمية الذكاء الاصطناعي * A (AStar).

من أجل هذا قمنا بإنجاز برنامجي حاسوب، الأول يمثل محاكاة خوارزمية من أجل إنجاز مسار جوي بأقل تكلفة مطبقة على خريطة مربعات أما البرنامج الثاني من أجل رصد و حل مشكلة التصادم الجوي بواسطة خوارزمية * A باستعمال « حركة النقطة الدوارة » و التي تعتبر سهلة الفهم من طرف الطيار هذا العمل يعتبر كذلك مجلة بكل الدراسات المنجزة في ميدان التصادم الجوي، إضافة إلى مختلف الطرق و التقنيات المستعمل في حل مشكلة التصادم الجوي.

REMERCIEMENTS

Je tiens à remercier en premier lieu, ALLAH le tout puissant de m'avoir donné la force, la foi, le courage et la volonté afin de réaliser ce mémoire, sans oublier mes très chers parents.

Je tiens à remercier Mr Salah BOUKRAA, Professeur au département d'aéronautique et directeur de département à l'U.S.D.B, qui a accepté d'être mon directeur de mémoire et qui m'a aidé beaucoup pour la réalisation de mon travail, aussi pour son grand et total soutien, son suivi et ses conseils précieuses durant mon projet et pour toute sa confiance qu'il m'a témoigné aussi ; merci beaucoup Monsieur.

J'exprime mes vifs remerciements, à Mr Smain MEGUELATI, Docteur d'état au département d'aéronautique à l'U.S.D.B, pour ses conseils éclairés et sa large expérience scientifique qui ont permis l'aboutissement de ce travail, ainsi que sa disponibilité totale pour m'avoir aidé à la réalisation de cet mémoire, de m'avoir pris aussi sous son aile.

Je tiens à remercier aussi, Mr Mourad ZERGOUG, Directeur de recherche au centre de recherche de contrôle et de soudage à Cheraga, pour son aide précieuse et de m'avoir encouragé durant les durs moments que j'ai passé, pour la réalisation de ce mémoire ainsi que pour son humour.

A ces grands messieurs j'ai à leur dire aujourd'hui :

Les épis pleines se courbent humblement et les épis vides lèvent la tête orgueilleusement. Je pense que la première partie de cette vieille citation, leur convient parfaitement.

Mes grands remerciements vont aussi pour les autres membres de jury qui m'ont honoré d'avoir accepté et consacré de leur temps pour examiner mon travail, et la rédaction des rapports de ce mémoire : Monsieur REZZOUG Taher et Monsieur LAGHA Mohand du département d'aéronautique de Blida.

Je tiens à remercier Mr Abdelouahab DJATOUF, directeur d'exploitation de la navigation aérienne à l'ENNA, de m'avoir donné l'idée du premier bout de fil, pour choisir ce sujet.

Je tiens à remercier profondément le Docteur Abdelkarim OTMANI expert en médecine légale au laboratoire régional de police à Constantine, de m'avoir aidé et encouragé durant mes études.

Je remercie Mr Mourad SAMAI, Professeur en génie civil à l'université de Constantine.

Je remercie énormément, le capitaine MOUADH et le commandant TAREK, pour leurs soutiens et de m'avoir aidé dans mes débuts des études de magistère.

Nombreux sont ceux qui n'ont pas mesuré leur temps pour m'apporter leur aide, notamment les enseignants (Rachi BENAÏSSA, Elhadi KALI, NOUIWA Ismail) et le personnel du département d'aéronautique de Blida (Mme Leila LAZREG, Mme Habiba BATEL, Mme Fatima Zohra NEFOUCI, Mme DEGUICHE Nadia) que je remercie infiniment, ainsi que mes amis proches (Soltan ZAIDI, Kamel BRAKNI, Ilyes SAHRAOUI, Chakir HADJERSI, Lyes KAID, Zizou, Mohamed KHERROUBI et leurs familles respectives) sans oublier Mr Mohamed RELIMA, Melle Ilhem LABIOD et Melle Djamila MENNA.

Je remercie fortement Mr youcef, chef service au vice rectorat chargé de poste de graduation, pour son aide précieuse ainsi de ne pas hésité de m'aider à n'importe quel moment que j'ai besoin.

Je remercie aussi tous les membres de ma famille, notamment mes parents qui m'ont soutenu tout au long de ces longues années. Je ne saurais dire si j'aurai pu finir un jour mes études sans leur aide précieuse et leur soutien moral. Merci pour toutes vos leçons de la vie qui me servent aujourd'hui de repères ; je n'oublie pas aussi mon petit frère HOUSSEM pour ses petits services.

Je remercie ma tante Yamina et Khalti Aicha pour tous leurs aides, soutient et conseils précieuses qui mon offert, également mes remerciement vont a Messaoud YEHYAOUI.

Enfin, je remercie tous ceux ou celles qui m'ont donné leur soutien et leur amitié tout au long de ce travail; qu'ils trouvent ici l'expression de mes remerciements les plus sincères.

Pour toutes personnes aimes l'ALGERIE, one two three viva l'Algérie.

« Marche avec des sandales
jusqu'à ce que la sagesse te
procure des souliers »

Avicenne

LISTE DES FIGURES

Figure 1-1 : Classes de l'espace	26
Figure 1-2 : Routes et points de reports.....	27
Figure 1-3 : Schéma de routage.....	28
Figure 1-4 : Secteurs de l'espace algérien.....	29
Figure 1-5 : Espace aérien contrôlé.....	31
Figure 1-6 : Organisation de l'espace aérien.....	32
Figure 1-7 : Division de l'espace aérien inférieur.....	33
Figure 1-8 : Division de l'espace aérien supérieur.....	33
Figure 1-9 : Service de contrôle de la circulation aérienne.....	40
Figure 2-1 : Force répulsive et force glissante, forces de glissement coordonnées.....	50
Figure 2-2 : Structure du réseau de neurones.....	51
Figure 2-3 : Données en entrée.....	52
Figure 3-1 : Méthode d'évitement par offset.....	55
Figure 3-2 : Méthode d'évitement par point tournant.....	56
Figure 3-3 : Les étapes de déroulement des Algorithmes génétiques.....	58
Figure 3-4 : Modèle d'offset.....	59
Figure 3-5 : Modèle de point tournant.....	59
Figure 3-6 : Surface occupé par la manœuvre.....	61
Figure 3-7 : Opérateur de croisement.....	62
Figure 3-8 : Opérateurs de croisement.....	62
Figure 3-9 : Principe de l'opérateur de mutation.....	63

Figure 3-10 : Modélisation d'une trajectoire d'évitement par point tournant.....	66
Figure 3-11 : Différentes formes possibles pour l'ensemble $P(t)$	71
Figure 3-12 : Modélisation d'une trajectoire d'évitement par offset.....	72
Figure 4-1: Exemple d'observation du A^*	77
Figure 4-2 : Trajectoire définie par l'algorithme de Dijkstra dans l'absence d'un obstacle.....	78
Figure 4-3 : Trajectoire définie par l'algorithme de « Best-First-Search » dans l'absence d'un obstacle.....	79
Figure 4-4 : Trajectoire définie par l'algorithme de Dijkstra dans la présence d'un obstacle.....	79
Figure 4-5 : Trajectoire définie par l'algorithme de « Best-First-Search » dans la présence d'un obstacle.....	80
Figure 4-6 : Trajectoire définie par l'algorithme A^* dans l'absence d'un obstacle..	81
Figure 4-7 : Trajectoire définie par l'algorithme A^* dans la présence d'un obstacle.....	81
Figure 4-8 : Trajectoire de la fonction heuristique en utilisant la distance de Manhattan	83
Figure 4-9 : Trajectoire de la fonction heuristique en utilisant la distance diagonale.....	83
Figure 4-10 : Trajectoire de la fonction heuristique en utilisant la distance euclidienne.....	84
Figure 4-11 : Trajectoire de la fonction heuristique en utilisant la distance euclidienne au carré.....	85
Figure 4-12 : Zone de recherche du chemin.....	86
Figure 4-13 : Observation de la notion de parent.....	86
Figure 4-14 : Notion de calcul.....	88
Figure 4-15 : Première étape de recherche.....	89
Figure 4-16 : Notion de recherche.....	90
Figure 4-17 : Notion d'arrêt de recherche.....	91
Figure 4-18 : Notion de tracer le chemin.....	91
Figure 4-19 : Inexistence du chemin dans le cas d'un mur infranchissable.....	93
Figure 4-20 : Nœuds de recherche balayé en absence d'obstacle.....	94
Figure 4-21 : Plus court Chemin trouver en absence d'obstacle.....	94
Figure 4-22 : Nœuds de recherche balayé en absence d'obstacle.....	95

Figure 4-23 : Plus court Chemin trouver en absence d'obstacle.....	95
Figure 4-24 : Nœuds de recherche balayé dans la présence d'un obstacle.....	96
Figure 4-25 : Plus court chemin trouver dans la présence d'un obstacle.....	96
Figure 4-26 : Nœuds de recherche balayé dans la présence d'un obstacle.....	97
Figure 4-27 : Plus court chemin trouver dans la présence d'un obstacle.....	97
Figure 4-28 : Nœuds de recherche balayé dans la présence d'un obstacle.....	98
Figure 4-29 : Plus court chemin trouver dans la présence d'un obstacle.....	98
Figure 5-1 : Trajectoire d'évitement avec un point tournant.....	100
Figure 5-2 : Développement des nœuds d'un arbre.....	105
Figure 5-3 : Développement d'un arbre pour un avion.....	107
Figure 5-4 : Notons que chaque avions à l'étape E_1 a le choix entre deux directions possibles, à droite ou à gauche.....	111
Figure 5-5 : Les nœuds de recherche de l'algorithme A^* , Avion A ($x_0=1555$ (Nm), $y_0=055$ (Nm), $v=200$ (kt), $cap=90$ (°)), Avion B ($x_0=055$ (Nm), $y_0=055$ (Nm), $v=450$ (kt), $cap=90$ (°)).....	112
Figure 5-6 : Les trajectoires résolues par l'algorithme A^*	113
Figure 5-7 : Les distances de séparation entre les deux avions avant et après résolution par rapport à la norme de séparation « 5 Nm ».....	113
Figure 5-8 : Les nœuds de recherche de l'algorithme A^* , Avion A ($x_0=055$ (Nm), $y_0=055$ (Nm), $v=390$ (kt), $cap=90$ (°)), Avion B ($x_0=6055$ (Nm), $y_0=055$ (Nm), $v=400$ (kt), $cap=270$ (°)).....	114
Figure 5-9 : Les trajectoires résolues par l'algorithme A^*	115
Figure 5-10 : Les distances de séparation entre les deux avions avant et après résolution par rapport à la norme de séparation « 5 Nm ».....	115
Figure 5-11 : Les nœuds de recherche de l'algorithme A^* , Avion A ($x_0=055$ (Nm), $y_0=055$ (Nm), $v=180$ (kt), $cap=90$ (°)), Avion B ($x_0=1555$ (Nm), $y_0=3555$ (Nm), $v=450$ (kt), $cap=180$ (°)).....	116
Figure 5-12 : Les trajectoires résolues par l'algorithme A^*	117
Figure 5-13 : Les distances de séparation entre les deux avions avant et après résolution par rapport à la norme de séparation « 5 Nm ».....	117
Figure 5-14 : Les nœuds de recherche de l'algorithme A^* , Avion A ($x_0=055$ (Nm), $y_0=055$ (Nm), $v=190$ (kt), $cap=90$ (°)), Avion B ($x_0=255$ (Nm), $y_0=1555$ (Nm), $v=480$ (kt), $cap=160$ (°)).....	118
Figure 5-15 : Les trajectoires résolues par l'algorithme A^*	119

Figure 5-16 : Les distances de séparation entre les deux avions avant et après résolution par rapport à la norme de séparation « 5 Nm ».....	119
Figure 5-17 : Les nœuds de recherche de l'algorithme A*, Avion A ($x_0=055$ (Nm), $y_0=055$ (Nm), $v=350$ (kt), $cap=90$ (°)), Avion B ($x_0=3055$ (Nm), $y_0=2055$ (Nm), $v=470$ (kt), $cap=270$ (°)).....	120
Figure 5-18 : Les trajectoires résolues par l'algorithme A*.....	121
Figure 5-19 : Les distances de séparation entre les deux avions avant et après résolution par rapport à la norme de séparation « 5 Nm ».....	121
Figure 5-20 : Les nœuds de recherche de l'algorithme A*, Avion A ($x_0=055$ (Nm), $y_0=055$ (Nm), $v=400$ (kt), $cap=90$ (°)), Avion B ($x_0=5055$ (Nm), $y_0=-1055$ (Nm), $v=220$ (kt), $cap=300$ (°)).....	122
Figure 5-21 : Les trajectoires résolues par l'algorithme A*.....	123
Figure 5-22 : Les distances de séparation entre les deux avions avant et après résolution par rapport à la norme de séparation « 5 Nm ».....	123
Figure 5-23 : Les nœuds de recherche de l'algorithme A*, Avion A ($x_0=055$ (Nm), $y_0=055$ (Nm), $v=300$ (kt), $cap=90$ (°)), Avion B ($x_0=555$ (Nm), $y_0=1555$ (Nm), $v=270$ (kt), $cap=40$ (°)).....	124
Figure 5-24 : Les trajectoires résolues par l'algorithme A*.....	125
Figure 5-25 : Les distances de séparation entre les deux avions avant et après résolution par rapport à la norme de séparation « 5 Nm ».....	125
Figure 5-26 : Les nœuds de recherche de l'algorithme A*, Avion A ($x_0=10$ (Nm), $y_0=15$ (Nm), $v=270$ (kt), $cap=150$ (°)), Avion B ($x_0=30$ (Nm), $y_0=15$ (Nm), $v=335$ (kt), $cap=220$ (°)).....	126
Figure 5-27 : Les trajectoires résolues par l'algorithme A*.....	127
Figure 5-28 : Les distances de séparation entre les deux avions avant et après résolution par rapport à la norme de séparation « 5 Nm ».....	127
Figure 5-29 : Les trajectoires résolues par l'algorithme A*, Avion A ($x_0=20$ (Nm), $y_0=0$ (Nm), $v=450$ (kt), $cap=90$ (°)), Avion C ($x_0=59$ (Nm), $y_0=-6$ (Nm), $v=420$ (kt), $cap=290$ (°)), Avion B ($x_0=55$ (Nm), $y_0=11$ (Nm), $v=420$ (kt), $cap=230$ (°)).....	131
Figure 5-30 : Les distances de séparation entre les paires d'avions « (A,B), (A,C) et (B,C) » avant la résolution par rapport à la norme de séparation « 5 Nm »....	132
Figure 5-31 : Les distances de séparation entre les deux avions A et C avant et après la résolution par rapport à la norme de séparation « 5 Nm ».....	132
Figure 5-32 : Les distances de séparation entre les paires d'avions « (A,B) et (A,C) » avant et après la résolution par rapport à la norme de séparation« 5Nm ».....	133

Figure 5-33 : Les distances de séparation entre les paires d'avions « (A,B), (A,C) et (B,C) » après la résolution par rapport à la norme de séparation « 5 Nm »....	133
Figure 5-34 : Les trajectoires résolues par l'algorithme A*, Avion C ($x_0=59$ (Nm), $y_0=-6$ (Nm), $v=420$ (kt), $cap=290$ (°)), Avion B ($x_0=55$ (Nm), $y_0=11$ (Nm), $v=420$ (kt), $cap=230$ (°)), Avion A ($x_0=20$ (Nm), $y_0=0$ (Nm), $v=450$ (kt), $cap=90$ (°)).....	134
Figure 5-35 : Les distances de séparation entre les paires d'avions « (A,B), (A,C) et (B,C) » avant la résolution par rapport à la norme de séparation « 5 Nm »....	135
Figure 5-36 : Les distances de séparation entre les deux avions A et C avant et après la résolution par rapport à la norme de séparation « 5 Nm ».....	135
Figure 5-37 : Les distances de séparation entre les paires d'avions « (A,B) et (A,C) » avant et après la résolution par rapport à la norme de séparation« 5Nm ».....	136
Figure 5-38 : Les distances de séparation entre les paires d'avions « (A,B), (A,C) et (B,C) » après la résolution par rapport à la norme de séparation « 5 Nm »....	136
Figure 5-39 : Les trajectoires résolues par l'algorithme A*, Avion A ($x_0=30$ (Nm), $y_0=0$ (Nm), $v=400$ (kt), $cap=90$ (°)), Avion B ($x_0=32$ (Nm), $y_0=11$ (Nm), $v=420$ (kt), $cap=140$ (°)), Avion C ($x_0=48$ (Nm), $y_0=11$ (Nm), $v=420$ (kt), $cap=210$ (°)), Avion D ($x_0=59$ (Nm), $y_0=-6$ (Nm), $v=420$ (kt), $cap=290$ (°)).....	138
Figure 5-40: Les distances de séparation entre les paires d'avions « (A,B), (A,C), (B,C), (A,D), (B,D) et (C,D) » avant la résolution par rapport à la norme de séparation « 5 Nm ».....	139
Figure 5-41: Les distances de séparation entre les deux avions A et C avant et après la résolution par rapport à la norme de séparation « 5 Nm ».....	139
Figure 5-42: Les distances de séparation entre les paires d'avions « (A,B) et (A,C) » avant et après la résolution par rapport à la norme de séparation« 5Nm ».....	140
Figure 5-43 : Les distances de séparation entre les paires d'avions « (A,D), (B,D) et (C,D) » avant et après la résolution par rapport à la norme de séparation« 5Nm ».....	140
Figure 5-44 : Les distances de séparation entre les paires d'avions « (A,B), (A,C), (B,C), (A,D), (B,D) et (C,D) » après la résolution par rapport à la norme de séparation « 5 Nm ».....	141
Figure 1: Plus court Chemin trouver en absence d'obstacle.....	146
Figure 2 : Plus court Chemin trouver en absence d'obstacle.....	147
Figure 3 : Inexistence du chemin dans le cas d'un mur infranchissable.....	147
Figure 4 : pas de solution dans ce cas.....	148
Figure 5 : Plus court chemin trouver dans la présence d'un obstacle.....	148

Figure 6 : Plus court chemin trouver dans la présence d'un obstacle.....	149
Figure 7 : Plus court chemin trouver dans la présence d'un obstacle.....	149

LISTE DES TABLEAUX

Tableau 1-1 : Classification des espaces aériens.....	34
Tableau 3-1 : Codage d'un individu à n avions.....	59
Tableau 5-1 : Données initiales des deux avions du premier cas.....	112
Tableau 5-2 : Paramètres d'évitements de l'avion central B.....	112
Tableau 5-3 : Temps total et temps de retard de l'avion central.....	112
Tableau 5-4 : Données initiales des deux avions du premier cas.....	114
Tableau 5-5 : Paramètres d'évitements de l'avion central B.....	114
Tableau 5-6 : Temps total et temps de retard de l'avion central.....	114
Tableau 5-7 : Données initiales des deux avions du premier cas.....	116
Tableau 5-8 : Paramètres d'évitements de l'avion central B.....	116
Tableau 5-9 : Temps total et temps de retard de l'avion central.....	116
Tableau 5-10 : Données initiales des deux avions du premier cas.....	118
Tableau 5-11 : Paramètres d'évitements de l'avion central B.....	118
Tableau 5-12 : Temps total et temps de retard de l'avion central.....	118
Tableau 5-13 : Données initiales des deux avions du premier cas.....	120
Tableau 5-14 : Paramètres d'évitements de l'avion central B.....	120
Tableau 5-15 : Temps total et temps de retard de l'avion central.....	120
Tableau 5-16 : Données initiales des deux avions du premier cas.....	122
Tableau 5-17 : Paramètres d'évitements de l'avion central B.....	122
Tableau 5-18 : Temps total et temps de retard de l'avion central.....	122

Tableau 5-19 : Données initiales des deux avions du premier cas.....	124
Tableau 5-20 : Paramètres d'évitements de l'avion central B.....	124
Tableau 5-21 : Temps total et temps de retard de l'avion central.....	124
Tableau 5-22 : Données initiales des deux avions du premier cas.....	126
Tableau 5-23 : Paramètres d'évitements de l'avion central B.....	126
Tableau 5-24 : Temps total et temps de retard de l'avion central.....	126
Tableau 5-25 : Données initiales des trois avions.....	128
Tableau 5-26 : Données initiales des deux avions du cas « $A>B>C$ ».....	128
Tableau 5-27 : Données initiales des deux avions du cas « $A>C>B$ ».....	128
Tableau 5-28 : Données initiales des deux avions du cas « $C>A>B$ ».....	129
Tableau 5-29 : Données initiales des deux avions du cas « $C>B>A$ ».....	129
Tableau 5-30 : Données initiales des deux avions du cas « $B>A>C$ ».....	129
Tableau 5-31 : Données initiales des deux avions du cas « $B>C>A$ ».....	130
Tableau 5-32 : Données initiales de cas de quatre avions.....	138

INTRODUCTION

L'organisation du trafic aérien et de son contrôle doit répondre à un double objectif de sécurité et d'efficacité. Concernant la sécurité, un rôle important du contrôle aérien est d'assurer la séparation des avions : on dit que deux avions sont séparés quand la distance entre eux est suffisante pour exclure tout risque de collision. Si ce n'est pas le cas, on dit qu'ils sont en conflit.

Un système d'évitement automatique devra remplir deux fonctions principales : il devra assurer une prédétection des conflits potentiels, et proposer aux avions impliqués dans ces conflits des trajectoires d'évitement. C'est cette génération de trajectoires sans conflit qui est l'objectif des méthodes d'intelligence artificielle présentées ici (Chapitre3) et c'est le but de mon travail.

L'automatisation ou la délégation à bord, du contrôle aérien dans les zones où les avions auraient la liberté de choisir leurs routes, offre une alternative qui pourrait permettre de tirer vraiment les bénéfices de ces évolutions du trafic.

L'évolution des moyens techniques disponibles permet d'envisager des modifications profondes du contrôle aérien, à travers son automatisation et l'utilisation éventuelle des systèmes embarqués, grâce auxquels chaque avion serait responsable de la route qu'il suit, et du fait qu'il reste séparé des autres avions. La position d'un avion peut être connue très précisément grâce au positionnement par satellite (GPS : Global Positioning System). La trajectoire prévue peut être suivie très fidèlement grâce au gestionnaire de vol (FMS : Flight Management System). Tous ces progrès importants qui ont été faits permettent d'envisager de nouveaux modes embarqués de surveillance de l'espace et de diffusion d'information de vol.

La méthode de génération de trajectoire d'évitement présentée dans le chapitre 5, utilise un algorithme de recherche de chemin de coût minimal dans un arbre à coût additif (l'algorithme A*), pour générer des trajectoires d'évitement. Les trajectoires d'évitements normalisées sont adaptées pour pouvoir être traduites en termes de choix successifs, ce qui permet de représenter l'ensemble des trajectoires possibles par un arbre. Les trajectoires générées sont des trajectoires d'évitement par point tournant avec retour direct vers la destination. L'algorithme A* permet de générer des trajectoires de bonne qualité, tant en terme de retard que de respect des contraintes de séparations. L'approche séquentielle donne des résultats satisfaisantes, avec toujours la forte influence de l'ordre dans lequel les trajectoires sont générées.

Le but de ce travail est de proposer une méthode séquentielle de résolution automatique basée sur un ordre de priorité permet à chaque avion de définir la manœuvre optimale en termes de retard pour atteindre sa destination sans conflit. La résolution proprement dite utilise un algorithme d'intelligence artificielle qui s'appelle A* (AStar) pour optimiser la trajectoire d'un avion, celles des autres avions étant fixées comme contraintes.

CHAPITRE 1 :

GENERALITES SUR L'ESPACE AERIEN

1.1 Introduction

Avant toute chose, précisons dans ce chapitre pour les lecteurs non familiers du monde de la navigation aérienne quelques notions utiles à la compréhension du reste de ce document.

On distingue deux types de contrôle aérien : le contrôle d'approche et le contrôle en route, correspondant à différentes étapes du vol d'un avion. Le contrôle d'approche s'applique dans les zones entourant les aéroports d'arrivée et de départ. Le contrôle en route s'applique entre ces zones. Tant qu'un avion est dans la zone entourant son aéroport de départ, et une fois qu'il a atteint celle qui entoure son aéroport de destination, il suit des cheminements bien déterminés et il est contrôlé, depuis le sol, par des contrôleurs. La densité du trafic autour des aéroports rend nécessaire le contrôle, pour assurer la sécurité des avions. Le fait que les avions suivent des cheminements prédéfinis facilite la tâche des contrôleurs.

La circulation aérienne en-route est organisée selon des itinéraires prédéterminés (on parle aussi parfois de « couloirs aériens »), jalonnés de points de report obligatoires et des balises radioélectriques. Le trafic étant moins dense qu'autour des aéroports.

L'espace aérien civil est découpé en secteurs, placés chacun sous la responsabilité d'une unité de contrôle d'espace. Il s'agit d'une équipe de deux contrôleurs chargée de régler les conflits de trajectoire entre les avions qui suivent des routes traversant le secteur, ou le groupe de secteurs, à sa charge.

1.2 Définitions

Le problème de la génération des trajectoires sans conflit met en jeu un certain nombre de termes dont il est utile de préciser le sens [1,2]:

Définition 1.1 : – Plan de vol – Il contient tous les éléments indicatifs décrivant le vol prévu pour un avion. Ces informations sont déposées avant le départ auprès des services ATC avec, notamment les informations fondamentales suivantes :

- L'heure de départ ;
- Le niveau de vol demandé pour la phase de croisière ;
- La route prévue qui est décrite par une série de balises.

Définition 1.2 : – La prévision de trajectoire – Pour la plupart des projets d'automatisation, il faut, pour résoudre les conflits, commencer par les détecter, et, pour les détecter, il faut savoir prévoir la trajectoire des avions. Cette opération, sous jacente à presque tous les modèles automatisés, est source redoutable des problèmes, dont certains sont bien dissimulés. Il faut d'abord comprendre que, si la prévision de trajectoire était parfaite, il serait possible de calculer la trajectoire d'un avion depuis son point de départ jusqu'à sa destination et de résoudre alors tous les problèmes susceptibles de se poser dès le début. Il n'en est hélas pas ainsi. Prévoir la trajectoire d'un avion de façon précise est une opération extrêmement complexe en raison d'un grand nombre de facteurs non connus (consignes compagnie, masse, etc.) ou même impossibles à connaître avec précision (condition météorologique par exemple). La prévision de trajectoire se fait donc généralement sur un horizon temporel limité, et en admettant certaines incertitudes sur les paramètres du vol. Les délais temporels d'anticipation des trajectoires se situent dans une fourchette de 10 à 30 minutes, à l'exception des systèmes dits « réactifs » qui utilisent une anticipation de l'ordre de 5 à 10 minutes.

Définition 1.3 : – Séparations – Une distance horizontale, qui est exprimé en milles nautiques (Nm), appelée norme de séparation horizontale, et une distance verticale, exprimée en pieds (ft), appelé norme de séparation verticale, sont définies. On dit que deux avions sont séparés quand la distance entre leurs projections sur un plan horizontal est supérieure à la norme de séparation

horizontale ou quand la différence entre leurs altitudes est supérieure à la norme de séparation verticale.

Les valeurs de ces normes de séparation varient selon la classe de la zone de l'espace aérien dans laquelle on se place. Dans les méthodes présentées dans ce travail, nous n'utiliserons que la norme de séparation horizontale, puisque nous supposerons que les avions volent dans un plan horizontal. Dans le cas du trafic en route cette norme dépend de plusieurs facteurs, mais est souvent proche de 5 milles nautiques.

Définition 1.4 : – **Conflit élémentaire** – On dit que deux avions sont en conflit dès qu'ils ne sont plus séparés (Définition 1.3). Pour une durée t_f fixée, on dira que deux avions sont en conflit potentiel si la probabilité qu'un conflit ait lieu entre eux à l'intérieur de cette durée est non nulle. La résolution de conflit consiste à modifier les trajectoires des avions de manière à supprimer tout conflit potentiel entre eux.

Définition 1.5 : – **Cluster** – Un cluster est la fermeture transitive d'avions en conflit potentiel. Si un conflit potentiel (pour une durée temporel t_f donné) existe entre un avion a_1 et un avion a_2 , et que cet avion a_2 est lui même impliqué dans des conflits potentiels avec les avions a_3 et a_4 , alors les avions a_1, a_2, a_3 et a_4 forment un cluster. Dans toute la suite, pour plus de simplicité, on appellera conflit à n avions un cluster contenant n avions.

Toute tentative de résolution de conflit (automatiquement ou non) doit tenir compte de cette notion de clusters. En effet, en reprenant l'exemple donné dans la définition 1.5, la résolution séparée du conflit entre a_1 et a_2 serait inutile puisque la résolution du conflit entre a_2 et a_3 , en modifiant la trajectoire de l'avion a_2 risquerait de faire apparaître de nouveau un conflit entre a_1 et a_2 .

Les méthodes présentées ici ont pour but d'assurer une telle résolution de conflits, c'est-à-dire de fournir des trajectoires sans conflit à des avions impliqués dans des conflits potentiels. Dans le cadre d'une application au trafic réel de telles méthodes, la pré-détection de conflit jouerait un rôle très important. Par pré-détection des conflits, on entend une évaluation de la probabilité que deux avions soient en conflit pendant une durée de temps t_f .

Définition 1.6 : – Routes directes – La route directe d'un avion est le chemin le plus court, ou le plus avantageux pour l'avion (compte tenu des vents, par exemple) entre le point de sortie de la zone qui entoure l'aéroport de départ et le point d'entrée de celle qui entoure l'aéroport d'arrivée.

On envisage, dans le cadre des évolutions de l'organisation du trafic aérien de laisser les avions libre de choisir leur route dans certaines zones de l'espace aérien (celles où le trafic est moins dense). Ces routes choisies par les avions sont appelées free routes. Les avions pourront alors, s'ils souhaitent (c'est-à-dire si les compagnies aériennes le souhaitent), emprunter des routes directes (définition 1-6).

La sectorisation de l'espace aérien, une fois les couloirs aériens fixés, est un problème difficile et intéressant. En effet, de nombreuses contraintes existent : la charge de travail correspondant à chaque secteur doit rester raisonnable pour que la sécurité soit assurée, la géométrie des secteurs et celle des couloirs aériens doit tenir compte des besoins de coordination entre contrôleurs au moment du passage d'un avion d'un réseau à l'autre. L'optimisation de l'utilisation des couloirs aériens, c'est-à-dire le choix a priori de routes suivant différents couloirs aériens entre un aéroport de départ et un aéroport de destination, peut être utilisée, pour réduire la congestion de l'espace aérien.

C'est souvent aux environs des balises, points de passage obligés de nombreux avions, qu'apparaissent les conflits.

Définition 1.7 : – Système d'évitement automatique – On appellera système d'évitement automatique tout système visant à automatiser la résolution de conflit, que cette automatisation soit partielle (considérée comme une aide aux contrôleurs) ou totale.

Les définitions qui suivent correspondent à différentes approches de l'évitement aérien. Les différences entre ces approches sont réelles et généralement distinguées. Cependant les termes qui les désignent ne sont pas figés. On pourra trouver ailleurs les termes que nous employons ici (évitement centralisé, embarqué, distribué) utilisés pour désigner des notions différentes, où les notions que nous désignons par ces termes sont désignées par d'autres

termes. Les définitions que nous donnons ici précisent le sens qui devra être donné à ces termes dans la suite de ce mémoire.

Définition 1.8 : – Système d'évitement centralisé – On dira qu'un système d'évitement est centralisé, ou supervisé, quand les trajectoires sans conflits de tous les avions d'une zone donnée de l'espace aérien sont générées par une entité unique (que cette génération soit automatisée, ou assuré par un centre de contrôle). Ces trajectoires sans conflit sont alors imposées aux avions.

Définition 1.9 : – Système d'évitement embarqué – On dira qu'un système d'évitement est embarqué (ou éventuellement distribué) lorsque la trajectoire d'évitement de chaque avion est générée individuellement (dans le cadre d'une application d'une telle méthode, elle serait la plus souvent générée à bord de l'avion lui même, d'où le terme embarqué). On parle d'évitement réactif quand les actions d'évitement sont le résultat d'une commande continue, ou de commandes successives mais très rapprochées dans le temps.

Définition 1.10 : – Intervalles de manœuvres – Dans le cadre de l'évitement par point tournant (voir section 3.1), on appelle intervalles de manœuvres d'un avion les intervalles T_0 et T_1 (voir section 3.3).

Définition 1.11 : – Approche globale – Cette méthode génère, pour un conflit impliquant n avions, les trajectoires simultanément, en minimisant le retard entraîné par les déviations dues à l'évitement. Elle correspond à l'approche centralisé d'un système d'évitement.

Définition 1.12 : – Approche séquentielle – Elle consiste à générer successivement les n trajectoires des avions, selon un ordre de priorité, les $(i - 1)$ trajectoires déjà générées imposant des contraintes pour la génération de la trajectoire du $i - \text{ème}$ avion. Cette approche peut être utilisée dans une forme séquentielle d'un système d'évitement embarqué.

1.3 La gestion des flux du trafic aérien

Les aéronefs peuvent librement déposer leurs plans de vol et choisir leurs itinéraires. Que se passe-t-il si tous les avions veulent passer au même endroit au même moment ? En effet, la capacité de contrôle est limitée : à travers un même secteur on ne peut faire passer qu'un certain nombre d'avions par heure. Si plus d'avions veulent passer, et que rien n'est rien fait, alors le risque de collision

augmente. Il faut trouver des méthodes qui vont nous assurer que jamais une telle situation ne se produira. On appelle cela la gestion des flux (Air traffic Flow Management ou ATFM). [3,4]

Pour cela la gestion du trafic aérien est souvent décrite comme une imbrication de filtres (d'organisation ATM) dont le but est d'éviter la collision des avions circulant dans l'espace aérien, chaque filtre a des objectifs différents et gère des espaces et des horizons temporels distincts.

On peut grossièrement distinguer cinq niveaux :

1.3.1 Le filtre ATFM stratégique

C'est le filtre le plus grossier. Son but n'est pas au sens strict d'éviter des conflits mais plutôt d'organiser le trafic de façons microscopique à moyen et long terme (supérieure à 6 mois) en terme de routes, secteurs, zones, etc. et de l'affectation des flux de trafic sur le réseau de routes en publiant par exemple les schéma d'évolution du trafic.

1.3.2 Le filtre ATFM pré-tactique

Effectué un ou deux jours avant, il consiste à organiser une journée de trafic J, la veille (J-1) ou l'avant-veille (J-2). A ce moment on dispose de plus d'information sur les plans de vols, en fonction du trafic prévu et d'un schéma prévisionnel d'ouverture des secteurs aériens par les centres de contrôle.

1.3.3 Le filtre ATFM Tactique

Le but de ce filtre est de protéger les contrôleurs de la surcharge de travail. Les compagnies déposent leurs demandes d'horaires de décollage de 1 à 3 heures avant le vol. L'ATFM tactique attribue des créneaux, en tenant compte de la quantité de trafic, de la capacité de contrôle, des autres créneaux de décollage, il s'agit plutôt de mesures d'ajustement qui prennent en compte des événement encore mal connus la veille. Ainsi, le trafic est mal connu à une journée avant le vol mais est bien connu entre 3 à 6 heures avant son arrivée.

1.3.4 Le filtre ATC

Le contrôle aérien est une action à court terme, dont le but est d'éviter les conflits. C'est la phase clé consistant à un changement de cap d'un avion, un changement de niveau de vol, ou une modification de la vitesse. Ce processus est

effectué tout au long de la période de trafic considéré, et doit donc être fait en temps réel. La visibilité du contrôleur est un peu supérieure puisqu'il dispose des plans de vol quelques minutes avant l'entrée de l'avion dans le secteur.

1.3.5 Le filtre d'urgence

Ce filtre n'est censé intervenir que lorsque le système de contrôle est absent ou a été défaillant. L'objet de ce filtre n'est plus de séparer les avions mais plutôt d'éviter un abordage potentiel. La prédiction temporelle est inférieure à la minute et varie entre 25 et 40 secondes. Il est trop tard pour que le contrôleur intervienne puisque l'on estime qu'il lui faut entre 1 et 2 minutes pour analyser une situation, trouver une solution et la communiquer aux avions.

1.4 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 [45,46,15]:

- Vol VFR: signifie « Visual Flight Rules ». Les avions de la catégorie VFR doivent assurer eux-mêmes leur séparation les uns par rapport aux autres par des moyens visuels. Ils doivent être équipés d'un moyen de radiocommunication pour entrer la plupart des zones entourant les aéroports.
- Vol IFR: signifie « Instruments 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 du survol.

L'espace a été ensuite divisé volumiquement par l'OACI en six classes, 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 (quelques aéroports).
- Classes B, C et D : Elles 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 de vol 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.

La figure 1-1 résume cette classification :

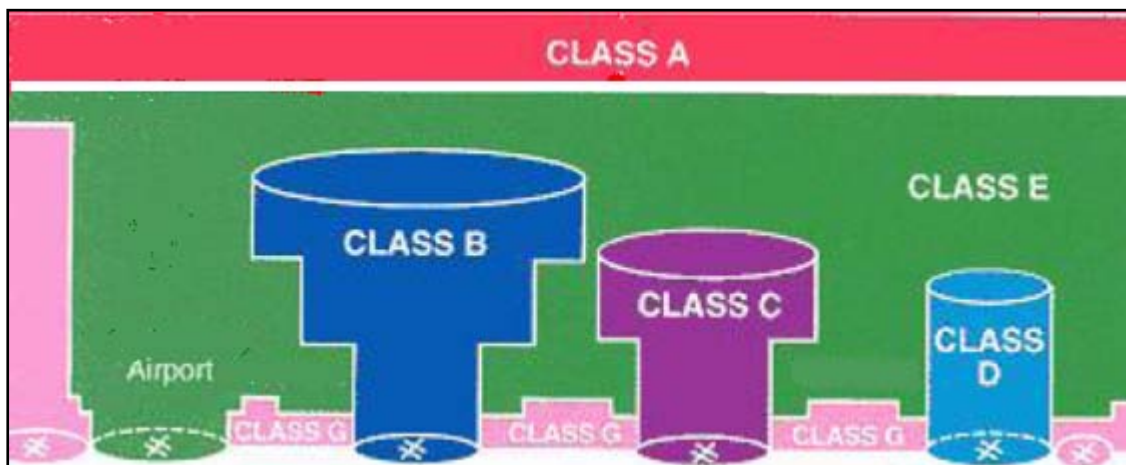


Figure 1-1 : Classes de l'espace

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. La première (indiquée comme étant une Victor airway) correspond à l'ancien modèle de route : les points de report (petits cercles) sont des balises de radionavigation qui existent physiquement au sol et sur lesquels les avions se repèrent grâce à leurs moyens de radionavigation (VOR, TACAN, etc.). Les chiffres indiqués dans les encadrés (109.2 par exemple) correspondent aux fréquences d'émission de ces balises. Pour des raisons souvent topographiques, il était impossible de positionner ces balises n'importe où.

Le second exemple (appelé Area Navigation Route, ou RNAV) correspond aux nouveaux modèles de route rendus possibles par l'évolution des moyens de radionavigation des avions. Les points de report (waypoint1, etc.) indiqués par des petits losanges 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 inertielles 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 de 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 pour leur entraînement.

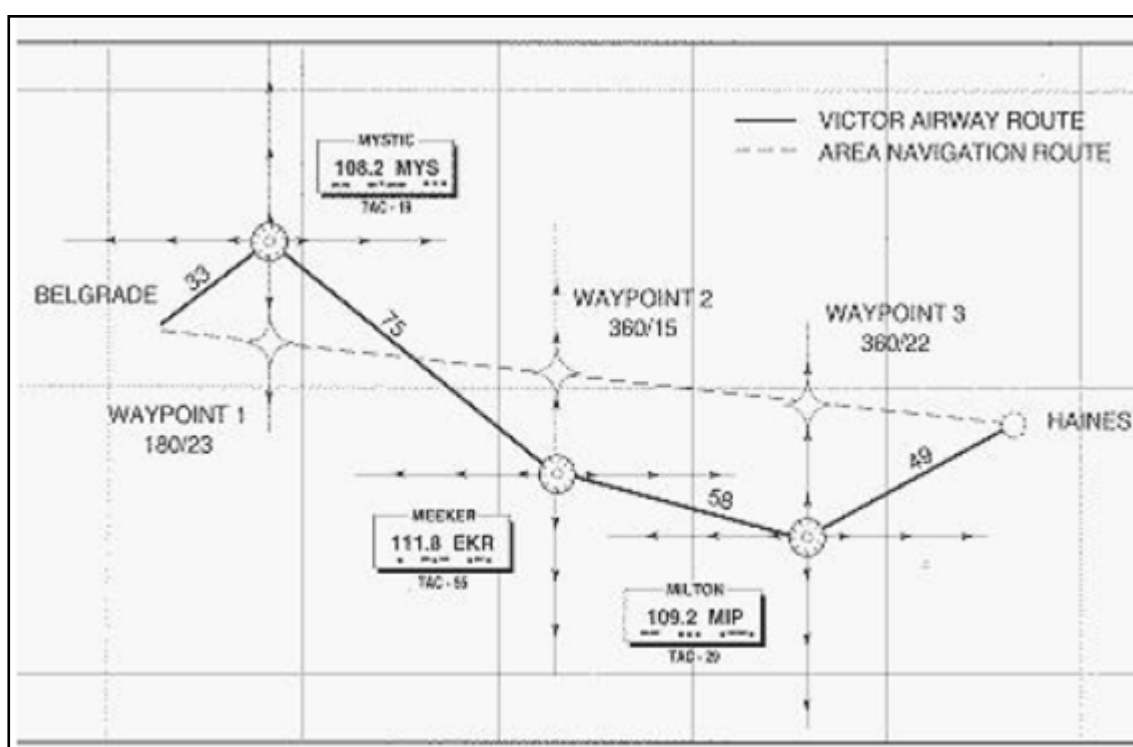


Figure 1-2 : Routes et points de reports

Cette cohabitation civil-militaire est d'ailleurs aujourd'hui difficile et génère des négociations délicates entre les deux autorités de tutelle, que ce soit au niveau national ou international.

Les réseaux de route sont aujourd'hui relativement rigides. Sur la figure 1-3, on voit le schéma de routage (Standard Routing Scheme) défini au dessus de la Turquie. On constate qu'il n'existe souvent qu'une seule route pour aller d'un point à un autre. Ces schéma de routage sont employés par les pilotes lors des dépôts de plan de vol.

L'espace aérien contrôlé est en effet découpé en tranches d'espace volumiques, appelées secteurs de contrôle. La figure 1-4 montre la projection 2D du découpage de l'espace Algérien. 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. Nous allons maintenant détailler cette tâche, qui est le contrôle du trafic aérien, ou ATC (Air Traffic Control).

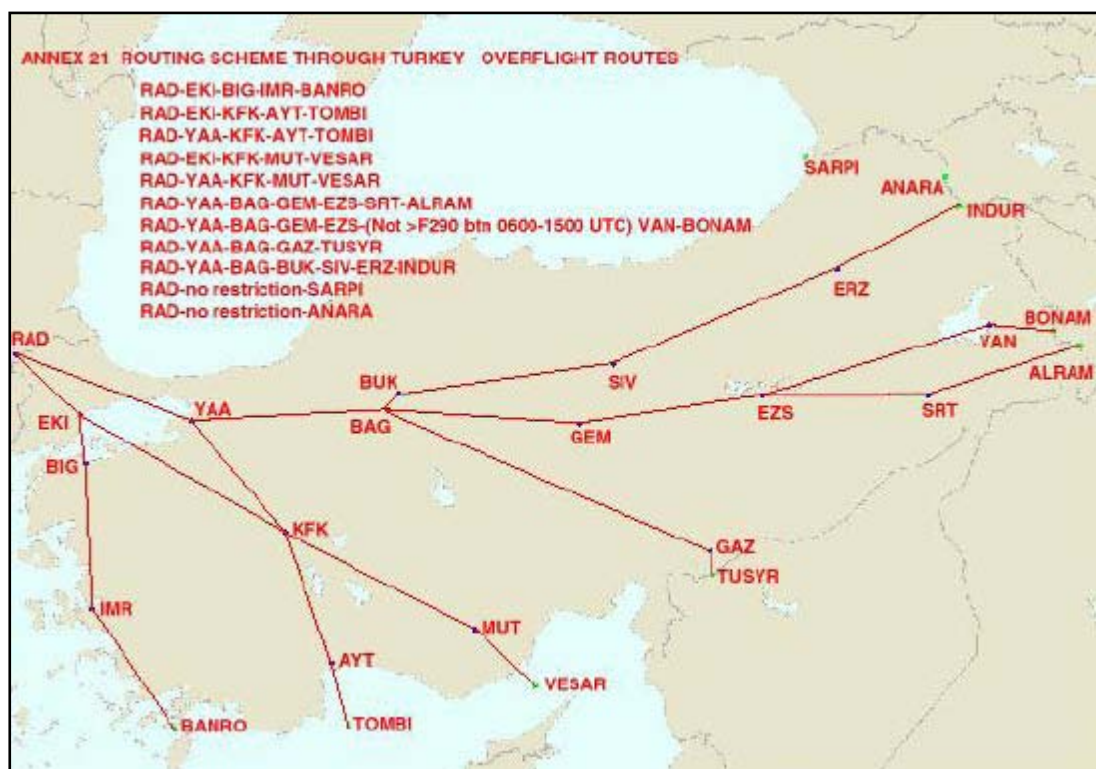


Figure 1-3 : Schéma de routage

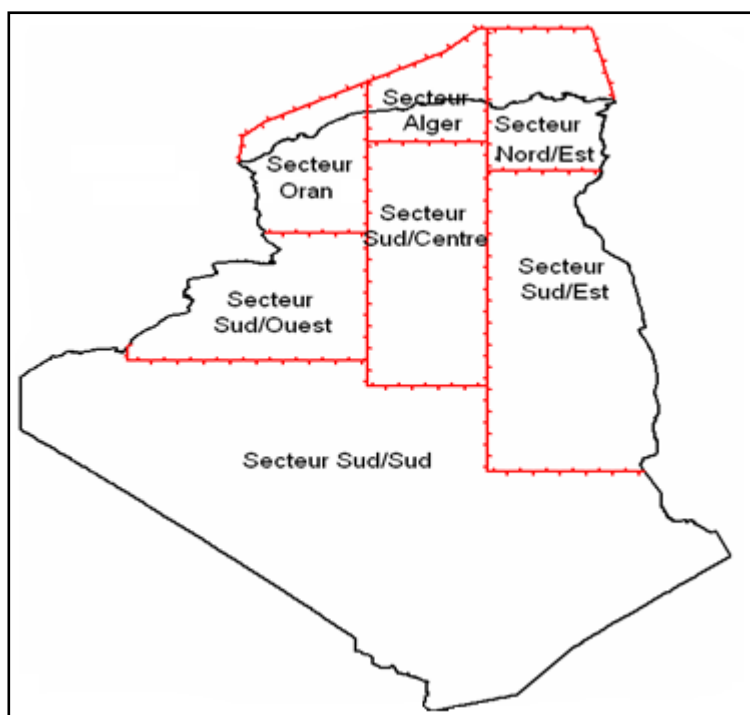


Figure 1-4 : Secteurs de l'espace aérien

1.5 Division de l'espace aérien

L'espace aérien n'est pas entièrement contrôlé, seules des portions d'espace le sont. En 1992, l'OACI (Organisation de l'Aviation Civile Internationale) a uniformisé l'appellation des différents espaces aériens (contrôlés ou non) et les a répartis en différentes classes de A à G qui correspondent à différents niveaux de service rendu. L'espace aérien est divisé en deux grandes parties : [26]

- FIR (Flight Information Region) : Région d'information de vol inférieure.
- UIR (Upper Information Region) : Région d'information de vol supérieure.

L'espace aérien est divisé en régions et zones, dont lesquelles les services de la circulation aérienne sont spécifiques et différents (figure 1.5). On peut diviser l'espace aérien sur la base des services rendus en deux types :

- Espace aérien contrôlé.
- Espace aérien non contrôlé.

1.5.1 Espace aérien contrôlé

Espace dans lequel un vol bénéficie des services rendus par un organisme chargé du service de contrôle de l'espace dans lequel il est entré, un espace

aérien n'est contrôlé que pendant les horaires de fonctionnement de l'organisme chargé d'y assurer le service de contrôle de la circulation aérien. En dehors de ces horaires l'espace aérien est non contrôlé, ce type d'espace comprend (figure 1-5, figure 1-6, figure 1-7, figure 1-8) :

- Les régions de contrôle CTA ;
 - Les régions de contrôle terminal TMA ;
 - Les vois aériennes AWY.
- Les zones de contrôle CTR ;
- Les régions supérieures de contrôle UTA.

■ Les Régions de Contrôle (CTA : Control Traffic Area)

Sont déterminées de manière à englober un espace aérien suffisant pour contenir les trajectoires des aéronefs en régime de vol IFR et au profit desquelles on juge nécessaire d'exercer la fonction contrôle, leurs limites géographiques sont fixées par les autorités compétentes.

Dans la CTA en distingue :

- Les régions de contrôle terminal (TMA : Terminal Management Area)

Région de contrôle établie en principe, au carrefour de routes ATS aux environs d'un ou plusieurs aérodromes importants.

Elles englobent les trajectoires d'arrivées et de départs d'un ou plusieurs aéroports.

- Les voies aériennes (AWY : Air Way) : Région de contrôle ou portion de région de contrôle présentant la forme d'un couloir et jalonné par des aides de radionavigations.

Désignation des AWY :

Les AWY sont désignées par une couleur suivie d'un numéro d'identification.

- « A » : Ambre (Amber).
- « B » : Bleue (Blue).
- « G » : Verte (Green).
- « R » : Rouge (Red).
- « W » : Blanche (White).

- Les AWY « A » et « B » ont une orientation générale Nord / Sud.
- Les AWY « G » et « R » ont une orientation générale Est / West.
- Les AWY « W » sont des voies aériennes saisonnières.

■ Les Zones de Contrôle (CTR)

C'est un espace aérien contrôlé s'étendant verticalement à partir de la surface de la terre jusqu'à une limite supérieure spécifiée.

Les limites latérales sont représentées généralement par un cercle d'au moins 9,3 km (5 Nm) à partir du centre de référence de l'aérodrome ou des aérodromes concernés, dans les directions à partir desquelles l'approche peut être effectuée.

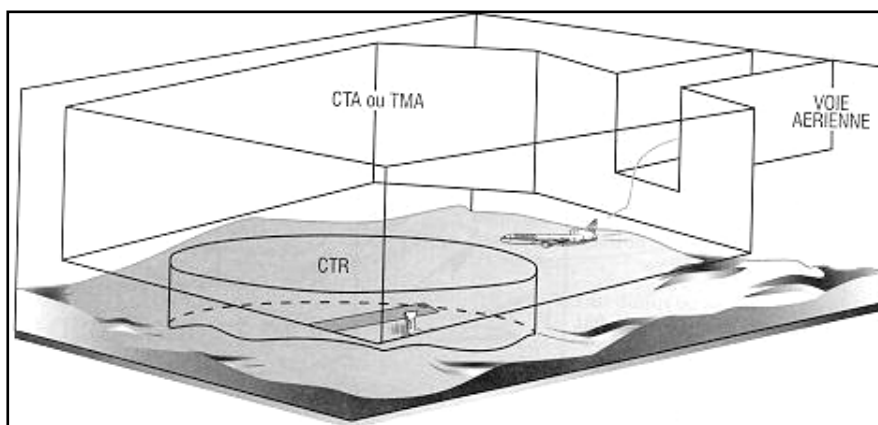


Figure 1-5 : Espace aérien contrôlé

■ Les régions supérieures de contrôle (UTA)

Afin de limiter le nombre de régions de contrôle que les avions volant à haute altitude auraient à traverser, il a été créé une région de contrôle supérieure englobant tout l'espace aérien supérieur, l'UTA ayant pour base le niveau de vol FL 245 (CTA), et pour sommet le FL 460.

Il s'agit des espaces contrôlés en espace supérieur. Leur limite inférieure est représentée par les CTA.

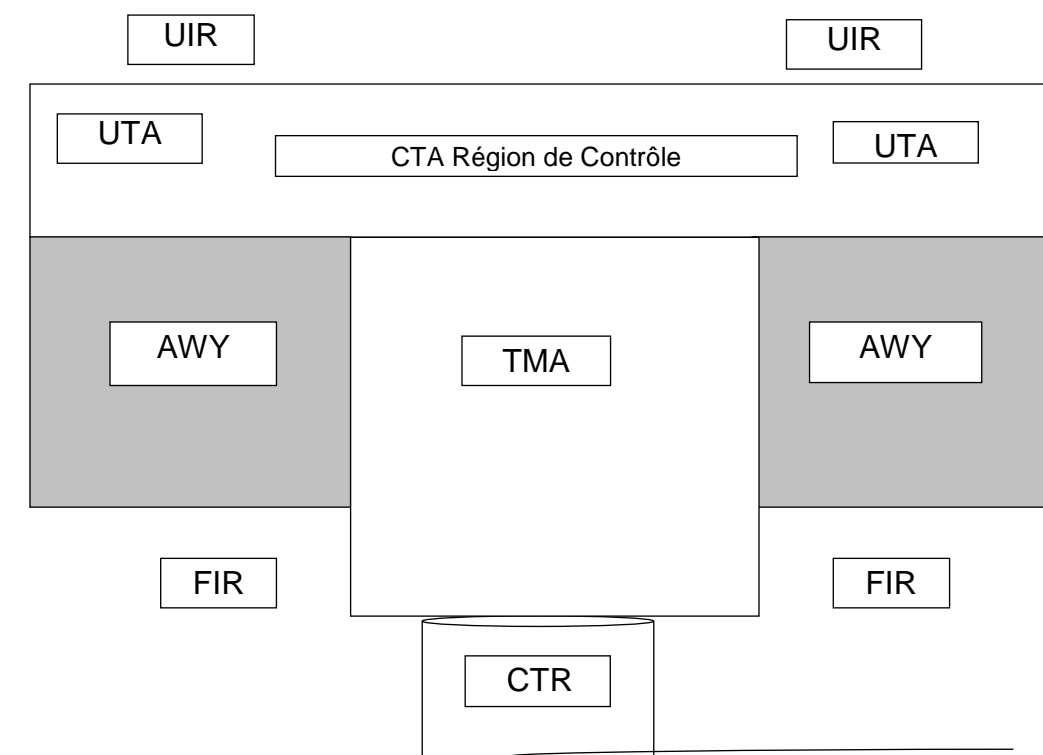


Figure 1-6 : Organisation de l'espace aérien

1.5.2 Espace aérien non contrôlé

Les espaces aériens non contrôlés sont des espaces de trafic moindre, où l'intervention des services de la circulation aérienne est limitée à l'information et l'alerte, ils se divisent en :

- Région d'information de vol F.I.R
- Région supérieure d'information de vol U.I.R
- Région d'information de vol (F.I.R)

Régions dans lesquelles les services d'information de vol sont assurés. Leurs limites géographiques sont déterminées en fonction des caractéristiques de portée du moyen de liaison au sol, ils sont généralement adjacents.

En Algérie l'espace aérien non contrôlé est constitué d'une seule FIR dont les limites latérales et verticales sont publiées dans l'AIP Algérien.

- Région supérieur d'information de vol (U.I.R)

Il a été créé afin de limiter le nombre de régions d'information de vol (F.I.R) que les aéronefs volant à très grande altitude aurait à traverser.

Une région supérieure d'information de vol (U.I.R) englobe l'espace aérien situé à l'intérieur des limites latérales d'un certain nombre de F.I.R.

1.6 Division verticale de l'espace aérien

1.6.1 Espace aérien inférieur

- Il va de la surface de la terre jusqu'au niveau FL245.

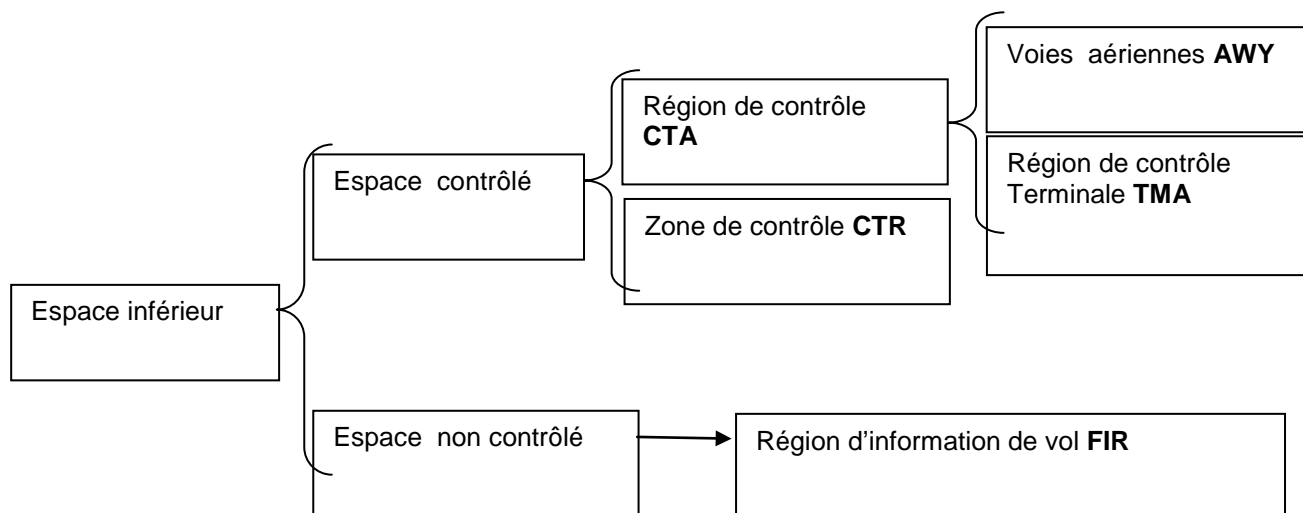


Figure 1-7 : Division de l'espace aérien inférieur

1.6.2 Espace aérien supérieur

- Il va de niveau FL245 sans limitation de plafond –

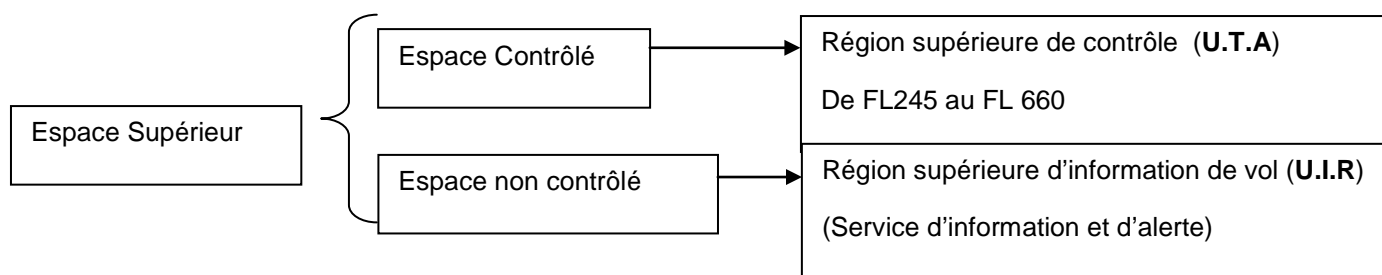


Figure 1-8 : Division de l'espace aérien supérieur

1.7 Classification des espaces aériens

Le (Tableau 1.1) ci-après fait état des conditions auxquelles sont assujettis les aéronefs dans les diverses classes d'espace aérien [15,26]:

Tableau 1-1 : Classification des espaces aériens

Classe	Type de vol	Séparation Assurée entre	Service assuré	Autorisation ATC obligatoire
A	IFR seulement	Tous aéronefs	ATC	OUI
B	IFR	Tous aéronefs	ATC	OUI
	VFR	Tous aéronefs	ATC	OUI
C	IFR	IFR et IFR IFR et VFR	ATC	OUI
	VFR	VFR et IFR	1) ATC pour la séparation des aéronefs IFR ; 2) Information de trafic VFR/VFR (et sur demande avis d'évitement de trafic)	OUI
D	IFR	IFR et IFR	ATC avec informations de trafic au sujet des vols VFR (et, sur demande, avis d'évitement de trafic)	OUI
	VFR	Sans objet	Information de trafic entre les vols VFR et IFR (et, sur demande, avis d'évitement de trafic)	OUI
E	IFR	IFR et IFR	ATC avec information de trafic au sujet des vols VFR dans la mesure du possible	OUI
	VFR	Sans objet	Information de trafic dans la mesure du possible	NON
F	IFR	IFR et IFR Autant que possible	Service consultatif de la circulation aérienne; service d'information de vol	NON
	VFR	Sans objet	Service d'information de vol	NON
G	IFR	Sans objet	Service d'information de vol	NON
	VFR	Sans objet	Service d'information de vol	NON

1.8 Espace à statut particulier

Ce sont des espaces aériens réservés à une utilisation spéciale, à des besoins de la défense nationale et à certaines catégories de manœuvre à savoir :

- Des espaces à procédure par réacteur : ce sont des espaces conçues pour le décollage et l'atterrissage des avions militaires ;
- Des volumes propres d'aérodrome ;
- Des secteurs de descente ;
- Des pinceaux de montée ;
- Des zones réservées d'aérodrome.

Pour cela il existe pratiquement trois types des zones à statuts particuliers :

- Les zones dangereuses ;
- Les zones réglementaires ;
- Les zones interdites.

1.8.1 Zones interdites

L'établissement d'une zone interdite devrait être soumis à des conditions particulièrement strictes, car l'usage de cet espace absolument interdit aux survols de tout aéronef, la pratique générale consiste donc à n'établir ce type de zones que pour protéger des installations importantes d'un état, les complexes industriels critiques dont les dommages, qu'entraîne un accident d'avion, risquerai de prendre des proportions catastrophiques ou des installations particulières sensibles qui sont indispensables pour garantir la sécurité du pays.

On les identifie par une lettre « P » suivie d'un numéro.

Exemple : DAP 60 de OUARGLA

1.8.2 Zones réglementées

Ce sont des zones définies au dessus du territoire ou des eaux territoriales d'un état, le vol des aéronefs y est subordonné à certaines conditions spécifiées qui peuvent aller jusqu'à l'interdiction de pénétration.

Une zone réglementée protège les activités militaires, elle peut être perméable a l'aviation civile, un processus de coordination doit être établie dans

ce cas entre les organismes militaires et civils intéressés. L'aéronef sera sous la responsabilité du gestionnaire de cette zone.

On les identifie par une lettre « R » suivie d'un numéro de la zone.

Exemple : DAR 49 à BOUSFER.

1.8.3 Zones dangereuses

Certaines zones ont un caractère particulièrement dangereux pour la navigation aérienne au vu de l'activité qui s'y déroule. La pénétration dans une zone dangereuse réclame une vigilance accrue du pilote et dans certains cas il est souhaitable de l'éviter lorsqu'elle est active.

Les zones dangereuses en espace supérieur ne sont pas gérées de la même façon qu'en espace inférieur.

Dans les lettres d'accord avec les organismes militaires, il est précisé que pendant les créneaux d'horaires d'activité, ces zones sont imperméables au trafic civil même si dans les règles de l'air rien n'interdit d'y pénétrer.

On les identifie par une lettre « D », suivie d'un numéro de la zone.

Exemple: DAD 74 TAFARAOU « entraînement ou pilotage: voltige vrille ... ».

La zone dangereuse implique un degré minimale de réglementation, tandis que la zone interdite constitue la forme la plus stricte. On notera toutefois que cette définition ne s'applique qu'à l'espace aérien situé au dessus du territoire d'un état, dans les régions qui ne sont soumises à aucun droit de souveraineté (haut mer) seules des zones dangereuses peuvent être établies par l'organisme responsable des activités qui motivent leur établissement.

1.9 Les services du contrôle de la circulation aérienne

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

Le but premier du contrôle du trafic aérien est d'assurer la sécurité du trafic et donc d'éviter les abordages entre les aéronefs opérant dans le système, puis d'optimiser les flux de trafic (figure 1-9).

Les services de la circulation aérienne ont pour but :

- d'empêcher les abordages entre les aéronefs ;
- d'empêcher les collisions entre les aéronefs au sol et les obstacles sur l'aire de manœuvre ;
- d'accélérer et régulariser la circulation aérienne ;
- de fournir des avis et des renseignements utiles à l'exécution sûre et efficace des vols ;
- d'alerter et d'aider les organismes appropriés pour les besoins de recherches et sauvetage.

La nécessité des services de la circulation aérienne sera déterminée par les considérations ci-après :

- Types de trafic en cause ;
- Densité de la circulation aérienne ;
- Conditions atmosphériques particulières ;
- Toutes autres conditions particulières.

La mission première reste donc la sécurité, la notion de capacité n'intervenant qu'ensuite. Le contrôle aérien peut être divisé en trois catégories :

1.9.1 Le contrôle d'aérodrome

Le contrôle d'aérodrome s'effectue à partir des tours de contrôle (TWR : TOWER), il assure la sécurité et le respect des procédures dans les phases de décollage, d'atterrissage et de roulage, afin :

d'empêcher les collisions sur l'aire de manœuvre ;

d'assurer l'acheminement sûr, ordonné et rapide de la circulation aérienne.

1.9.2 Le contrôle d'approche

Le contrôle d'approche : Service du contrôle de la circulation aérienne pour les aéronefs en vol contrôlé à l'arrivée et au départ, il s'effectue par un bureau d'approche ou un centre de contrôle régional (CCR). La fonction de ce service est de prévenir les abordages entre aéronefs, ordonner la circulation aérienne sur les phases d'arrivée et de départ.

1.9.3 Le contrôleur en route

Il s'agit du contrôle hors zone d'approche. C'est celui qui nous intéressera directement, et dont nous allons détailler maintenant le mode opératoire.[15,26]

Les contrôleurs en routes sont regroupés dans des centres de contrôle. Chacun de ces centres contrôle 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 contrôleur « organique » s'occupe principalement de la gestion de trafic à moyen terme. Pour ce faire, il dispose de petites bandes de papier (appelées « strips ») posées sur des réglettes rouges ou vertes en fonctions du sens de déplacement des vols et qui contiennent toutes les informations fondamentales pour chaque avion appelé à travers son secteur. Ces informations sont : l'heure d'entrée dans le secteur, la liste des balises de la route de l'avion, 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, pour ce faire, maintenir en permanence une séparation minimale entre les avions. Cette séparation est d'environ 5Nm (1Nm=1852m) 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 maintenir cette séparation, il dispose d'une image radar qui lui montre l'ensemble du trafic présent dans son secteur. Chaque avion est représenté par un plot et une étiquette qui contient l'ensemble des informations utiles : cap, indicatif, altitude, attitude (montée, stable, ou descente) et balise suivante. La vitesse est représentée graphiquement par une comète, qui est la trainée laissée

par l'avion derrière lui et correspond aux positions passées. 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 :

- 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 croisière.

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'un avion à gauche ou à droite, puis à lui faire reprendre sa trajectoire.

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.

Cette décomposition explique pourquoi on ne peut augmenter indéfiniment la capacité en multipliant le nombre de secteurs. En effet, pour des secteurs trop petits, la charge de coordination devient prépondérante, et empêche toute gestion du trafic.

Remarque : La mise en place de la RVSM a cependant tendance à augmenter le nombre de manœuvres en niveau utilisées.

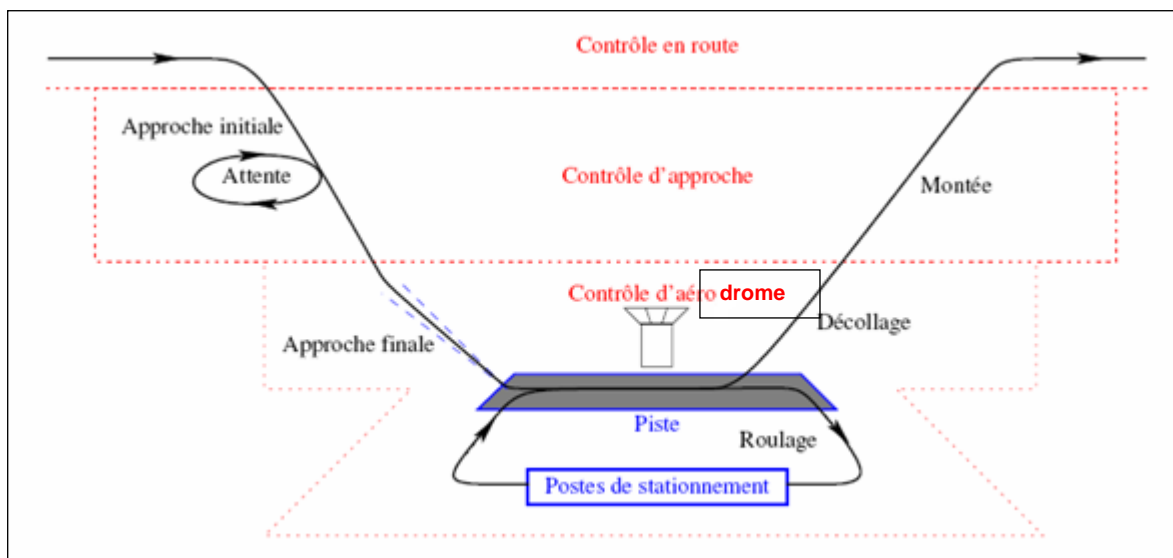


Figure 1-9 : Service de contrôle de la circulation aérienne

CHAPITRE 2 : ETAT DE L'ART

2.1 Introduction

Le système de gestion de la circulation aérienne est un système complexe par bien des aspects : il comporte un grand nombre d'agents humains et techniques. Il doit pouvoir traiter des aéronefs ayant des performances et des équipements variés ; c'est un système ouvert dont l'arrêt rapide dans un état sécuritaire est impossible.

Depuis 40 ans l'informatique a été introduite dans la gestion de la circulation aérienne. L'automatisation a porté sur le traitement plan de vol et radar pour distribuer l'information pertinente aux contrôleurs sur les secteurs et faciliter les transferts des vols entre secteurs. Elle a permis de faire passer le trafic, mais a peu augmenté la productivité des contrôleurs. Dans les 10 prochaines années des outils d'assistance au contrôleur et au pilote permettront la sécurité et la capacité dans une mesure que l'on ne sait pas évaluer.

L'automatisation plus poussée a fait l'objet de nombreux projets de recherche (AERA aux USA, ARC 2000 Eurocontrol, etc..).

La fonction principale du contrôle de la circulation aérienne est d'assurer un écoulement sûr et ordonné des vols, en évitant les abordages entre aéronefs en air ou au sol. Sa raison d'être est avant tout la sécurité.

L'automatisation consiste à transférer à un système (totalement ou partiellement) des tâches effectuées jusqu'à présent par un opérateur humain. Tant que l'automatisation n'est pas totale, l'opérateur garde un rôle et on parlera d'assistance automatisée.

La simple apparition du radar peut être déjà considérée, dans ce cadre, comme une forme élémentaire d'automatisation. L'automatisation est donc un long processus qui a ses racines aux origines du contrôle aérien et continue depuis sa lente progression.

2.2 Améliorer l'écoulement du trafic

Il existe plusieurs approches pour tenter d'améliorer l'écoulement du trafic. Elles se divisent en deux grandes catégories : d'une part l'amélioration de l'organisation de l'espace aérien et des flux de trafic, d'autre part, l'amélioration du contrôle aérien lui-même.[10,14]

2.2.1 L'organisation de l'espace et des flux

On peut tenter d'abord d'améliorer la façon dont l'espace est découpé en secteurs. Jusqu'à présent, ce découpage a souvent été fait de façon empirique, et une approche plus scientifique peut permettre d'espérer une amélioration capacitive.

On peut également tenter d'améliorer l'écoulement des flux en améliorant les modèles de capacité secteur, en proposant des tarifications du contrôle différentes en fonction des horaires, ou en proposant des routes alternatives aux aéronefs pour décongestionner certaines zones. Enfin, on peut imaginer un changement radical du découpage de l'espace en prévoyant par exemple une zone de type « Free-Flight » où le contrôle serait assuré différemment.

2.2.2 Le contrôle aérien

La principale raison des retards observés aujourd'hui en Europe et aux USA est la saturation des secteurs de contrôle « en route ». Pour tenter d'augmenter la capacité du contrôle en route, de nombreuses solutions sont à l'étude. Les problèmes à traiter sont bien identifiés : il faut savoir prévoir les trajectoires des avions, détecter les conflits, regrouper ces conflits par clusters, donner des manœuvres simples pour les résoudre, etc.

Si les problèmes liés à la résolution de conflits aériens sont donc bien identifiés, les techniques à employer le sont moins. Avant de rentrer dans les détails techniques du problème, nous allons nous employer à présenter quelques concepts généraux.

2.3 Différentes approches pour l'aide à la résolution

On peut grossièrement classer les différentes approches de la façon suivante : [2,5,6,7,8,9]

2.3.1 Approche anti-calculatoire

La première approche consiste à construire un modèle, dit « cognitif », du contrôleur, spécialement dans ses modes d'appréhension des paramètres du conflit, et d'utiliser ce modèle dans un calculateur afin de construire des outils de filtrage de l'information et d'assistance électronique au contrôleur. Cette 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, fiabilité du système expert; d'autre part, laissant l'homme totalement en charge de toutes les décisions, elle se contentera de repousser le « mur de la capacité » dont nous parlions plus haut, sans faire disparaître les causes fondamentales de son existence. Il s'agit donc d'une approche essentiellement courte ou moyen terme.

2.3.2 Automatisation centralisée complète

Il existe maintenant des débuts de preuves montrant que la réalisation d'un système automatisé de contrôle n'est pas une pure chimère. Dans une telle hypothèse, 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-3 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 à long ou à très long terme.

2.3.3 Délégation de tâches

Dans ce cas, l'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. [52,53]

2.3.4 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 Recherche et de développement 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 transocéaniques, 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". Il subsiste cependant de nombreux problèmes : d'une part, chaque avion n'a qu'une vision limitée du monde qui l'entoure ; il est donc parfaitement susceptible de choisir des manœuvres d'évitement à court terme qui peuvent se révéler désastreuses sur le long terme, soit en terme de sécurité, soit en terme d'efficacité ; il n'existe encore aujourd'hui aucun algorithme ayant prouvé son efficacité sur des échantillons du trafic réel. Enfin, les problèmes d'interface entre les zones ``libres" et les zones contrôlées ont été peu abordés.

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. [52,53]

2.4 Analyse biographique des approches du problème de résolution de conflits

On distingue dans la littérature deux types d'approches du problème de résolution de conflits. Les approches opérationnelles, historiquement plus anciennes, généralement à l'initiative des autorités du contrôle aérien, tiennent largement compte du contexte opérationnel, mais ne s'attaquent pas toujours au problème d'optimisation sous-jacent. Les approches théoriques, apparues plus récemment, s'attaquent efficacement à la complexité du problème, en faisant parfois des hypothèses irréalistes dans un contexte opérationnel. De rares approches abordent le problème combinatoire tout en essayant de respecter les contraintes opérationnelles.

2.4.1 Approches opérationnelles

5.6.2.1. Des systèmes centralisés au sol

■ La tentative américaine abandonnée AERA

Les travaux de recherche de la MITRE sur AERA ont été financés par la FAA. La première phase de ce projet, AERA 1 permettait de prévoir les trajectoires des avions en

fonction des intentions des pilotes et de détecter d'éventuelles violations de séparation standard ou de restriction de flux pour des plans de vol existant ou désignés par un contrôleur. C'était donc avant tout un outil d'aide à la décision qui ne proposait pas de solution aux opérateurs.

AERA 2 proposait d'importantes nouvelles aides aux opérateurs, et notamment une liste de « résolutions recommandées par l'ordinateur » aux éventuels conflits élémentaires détectés par AERA 1. AERA 2 introduisait également des outils d'aides à la coordination entre contrôleurs.

Dans AERA 3, la responsabilité de séparer les avions était laissée à la machine. La structure AERA 3 était hiérarchique. Au niveau national, l'ATMS restait en l'état. Il assurait une gestion des flux de trafic acceptable pour AERA 3.

AERA 3 se décomposé en trois niveaux hiérarchiques :

- ASF (Automated Separation Function) séparait les paires d'avions ;
- MOM (Maneuver Option Manager) assurait le respect du contexte global par l'ASF (qui ne savait gérer que des paires d'avions séparées) ;
- AMPF (Airspace Manager Planning Function) s'assurait que MOM pouvait opérer avec succès en prévenant les trop fortes densités de trafic.

ASF et MOM étaient les deux niveaux dont l'automatisation complète était envisagée. Néanmoins, le seul niveau détaillé clairement d'AERA 3 était l'ASF qui ne résolvait que des conflits à deux avions. Les objectifs de MOM étaient assez clairement définis. Par contre, son fonctionnement est toujours resté très flou et peu convaincant. La recherche d'optimalité global n'apparaissait pas du tout dans AERA 3.

L'ASF était assuré par un algorithme appelé « GENTLE-STRICT » (GS) dont le but était de résoudre automatiquement des conflits de croisement horizontal entre deux avions en utilisant des manœuvres latérales d'offset. GS prévoyait :

- Une manœuvre d'offset latéral pour un des deux avions dit « Gentle » selon un sens prédéterminé.
- Une consigne de tenue de trajectoire pour l'autre avion dit « Strict » dans le but d'éviter les déviations supérieures à un certain seuil de tolérance par rapport à sa trajectoire nominale.

GS ne résolvait un conflit que dans le plan horizontal et ne changeait jamais les vitesses des avions ou leur niveau de vol. GS agissait le plus tard possible afin d'éviter les manœuvre inutiles.

Le projet AERA n'a jamais abouti. Les algorithmes utilisés ou envisagés dans le MOM n'ont jamais été évoqués dans les rapports disponibles. Sur le plan de la modélisation, AERA3 peut garder un intérêt. Par contre, ce projet n'apporte pas de réponse au problème d'optimisation globale dès que le nombre d'avions dépasse 3.[2,4,7,16,37]

■ ARC2000 et ses dérivés

Le projet ARC2000 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 « tube 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.

Si l'on est capable de prédire avec une très bonne précision la trajectoire d'un avion, on peut représenter celle-ci par une courbe dans R^4 , les trois premières variables représentant la position dans l'espace de l'avion au temps t , la quatrième variable. Les imprécisions de mesure de positions et de tenue 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 $(n+1)$.

Le principe ARC2000 était donc originellement le suivant : le premier avion entrant dans le système se voyait affecter un tube optimal respectant son plan de vol. Dès qu'un nouvel avion se présentait, son tube optimal était calculé en considérant les tubes précédemment affectés comme des contraintes fixes. Autrement dit, un tube déjà affecté n'était pas remis en cause. L'optimalité globale n'était pas donc recherchée. Dans le cas où un avion ne respectait pas le tube qui lui était affecté, il devait négocier un nouveau tube respectant tous autres tubes. La faiblesse de ce principe se caractérisait par son manque de robustesse. En effet, il est possible que le non respect d'un tube 4D soit lié à des événements météorologiques par exemple, et qu'il ne touche pas seulement un avion mais plusieurs. On pouvait alors craindre un phénomène chaotique qui remettait en cause le principe adopté.

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 conflits pour les 20 minutes à venir. Il fallait en effet trouver un juste milieu entre prévoir 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 une gestion à 20 ou 30 minutes des groupes de conflits ou de clusters. La recherche de solutions optimales a également évolué puisqu'au principe du « dernier arrivé dernier servi », a succédé une série de règles permettant de classer les avions du plus prioritaire et les manœuvres de la plus opportune.

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

Aussi le projet ARC2000 a été testé sur du trafic réel. De la même façon que pour AERA3, la modélisation du problème est intéressante mais les algorithmes de résolution de clusters ne s'attaquent pas au problème d'optimisation globale des trajectoires. [2,4,7,34,52,53]

■ Le projet SAINTEX

Dans le projet SAINTEX du CENA, trois approches d'automatisation du contrôle en route étaient abordées.

Le scénario « Détection-Résolution » était un système orienté système expert. Le système essayait de reproduire le comportement du contrôleur. Les conflits étaient détectés en extrapolant les trajectoires d'avions 10 minutes dans le futur (6 minutes pour les avions évolutifs). Aussitôt détecté, un conflit était classé suivant différents critères tels que l'angle formé par les trajectoires, le rapport des vitesses, etc. pour chaque classe de conflits, une manœuvre prédéfinie était appliquée. Le système expert ne pouvait résoudre qu'un conflit entre deux avions.

Dans le scénario « 4D », une trajectoire sans conflit était générée pour tout avion entrant dans le secteur. Une trajectoire était représentée par un ensemble de points et de contraintes verticales. Pour chaque avion, on construisait un tube 4D représentant sa trajectoire compte tenue des incertitudes sur sa vitesse et sa position. Pour construire un tube admissible (sans conflit), SAINTEX faisait diverses tentatives partant de la trajectoire idéale (directe) vers des trajectoires de plus en plus pénalisantes mais résolvant les conflits. Du temps que l'on consacrait à la recherche d'une trajectoire admissible découlait la qualité de la résolution. La trajectoire de l'avion était ensuite surveillée de manière à s'assurer que l'avion respectait bien la trajectoire qui lui avait été affectée. Ce scénario, purement algorithmique, ressemblait fortement à celui décrit dans ARC2000.

Dans le scénario hybride, les avions stables étaient gérés par le système Détection-Résolution et les autres par le système 4D.

Utiliser un système expert pour résoudre un conflit à deux avions ne se justifie pas. Le projet SAINTEX, par définition, s'intéressait à la gestion automatique d'espaces aériens peu saturés. Le problème de la résolution des clusters d'avions était évoqué, mais non résolu. [2,4,7,29,40,52,53]

5.6.2.2. Des systèmes distribués : avions autonomes et séparation embarquées

■ FREER (FREER-Route Experimental Encounter Resolution)

Le projet FREER est né en 1995 au centre expérimental Eurocontrol. L'idée est de généraliser les règles de résolution du vol à vue par des EFR (Extended Flying Rules). Les concepteurs de FREER ont complété les règles de l'air de façons à pouvoir d'une part prendre en compte toutes les configurations de conflits à deux avions, et d'autre part de définir un ordre total sur l'ensemble des avions dès lors qu'on s'intéresse à trois avions ou plus. Par exemple, si trois avions arrivent simultanément sur le même point en suivant des routes Nord, 120 degrés et 240 degrés, la règle de priorité à droite ne permet pas de définir l'ordre de priorité pour ces trois avions. La prévision et la résolution de trafic se faisait avec une anticipation de 6 à 8 minutes. Le fonctionnement de FREER dans ce contexte n'a jamais été précisément décrit. [2,4,7,24,21]

■ Le projet FACES

Le projet FACES, 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 cette ordre défini, on utilise un algorithme classique

en robotique (A*) pour résoudre les conflits. FACES a été testé en simulation arithmétique sur du trafic réel. [2,4,7,12,28,41]

2.4.2 Approches théoriques

■ Méthode des forces répulsives

Karim Zeghal a une approche originale du problème, aboutissant à une solution distribuée du problème de la séparation des avions en route. Le principe de sa méthode est l'application aux avions des forces. Cette approche offre l'avantage de ne nécessiter ni communication ni négociation entre les avions, et être très robuste aux perturbations. Elle laisse en revanche de côté l'optimisation des trajectoires et implique de pouvoir modifier à tout instant la direction de chaque avion. De plus, elle supporte mal une augmentation de la densité de trafic. [2,4,38,39,31]

Les forces utilisées sont de trois types (figure 2-1) :

- Les forces attractives :

Elles sont dirigées vers la destination des avions.

- Les forces répulsives :

Elles sont définies à partir d'un potentiel de danger, causé par un autre avion ou une zone interdite. Dans le cas d'un autre avion, par exemple, le potentiel de danger pourra être inversement proportionnel à la distance. Les forces répulsives, perpendiculaires aux lignes équipotentielles, dirigées vers les potentiels décroissant, entraînent un éloignement rapide des obstacles proches.

- Les forces de glissement :

Elles ont tangentes à l'équipotentiel passant par l'avion, deux directions sont possibles : on choisit celle qui contribue à rapprocher l'avion de sa destination. Ces forces de glissement permettent le contournement des obstacles. Il est possible de définir ces forces de manière à ce que deux avions en conflit potentiel l'un avec l'autre aient des forces de glissement complémentaires. Cela introduit une coordination d'action sans concertation et permet d'obtenir des trajectoires d'évitement assez efficaces.

A partir des forces s'appliquant à un avion, sa trajectoire est modifiée en suivant la direction donnée par une combinaison de ces forces (pour un conflit impliquant deux avions, Karim Zeghal propose d'additionner simplement ces forces, pour un conflit plus complexe, le dosage de ces forces peut être un problème délicat).

Les travaux de Karim Zeghal, très intéressants par l'originalité de la méthode proposée, ont comme atout majeur leur robustesse. Des inconvénients néanmoins subsistent encore :

- Les trajectoires obtenues, dans lesquelles le cap des avions varie constamment, sont difficiles à suivre pour les pilotes (et seraient pénibles pour les passagers). Une simplification des trajectoires obtenues serait nécessaire.
- La généralisation de la méthode à des conflits impliquant plus de deux avions, dans le cadre d'une augmentation de la densité du trafic, a été étudiée par J.-F Bosc. Les résultats obtenus montrent que les limites du système sont vite atteintes.

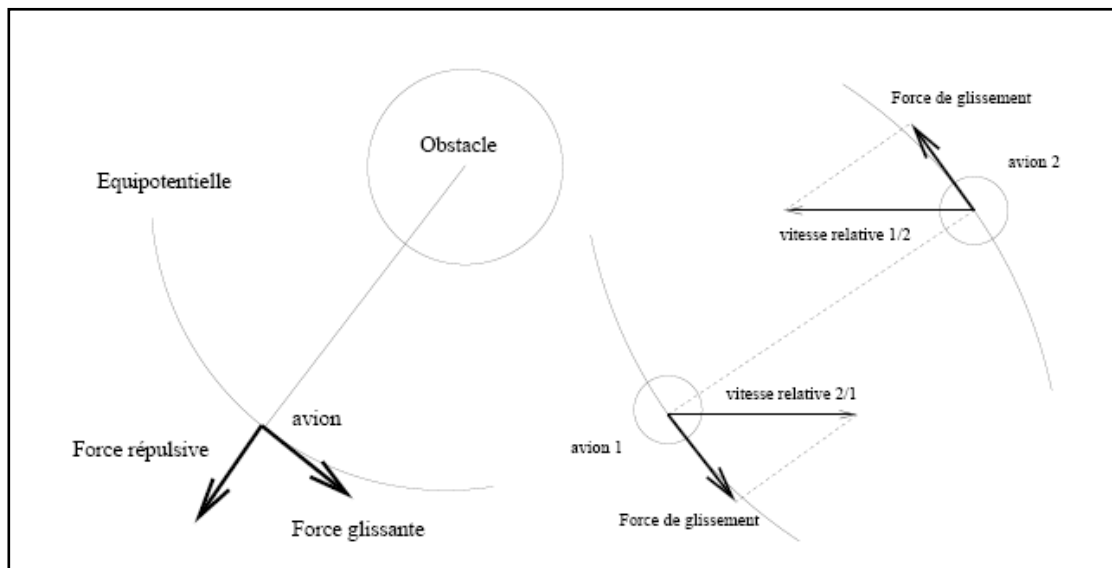


Figure 2-1 : Force répulsive et force glissante, forces de glissement coordonnées

■ Méthodes neuronales

Nicolas Durant et Frédéric Médioni utilisent un algorithme génétique pour l'apprentissage d'un réseau de neurones qui agit sur les commandes d'un avion afin d'éviter un conflit. La figure 2-2 donne une idée de la structure du réseau de neurones utilisé en conflit à deux avions. Les données en entrée sont représentées sur la figure 2-3. La distance relative λ des deux avions, le relèvement angulaire γ , l'angle β entre les deux trajectoires, ou le relèvement de la destination sont les principales données utilisées pour les entrées du réseau. Le réseau optimisé est ensuite testé sur des cas pris au hasard. Seul le plan horizontal est traité.

L'extension de la méthode au cas de conflits à trois avions est difficile. On peut tout d'abord chercher à optimiser un réseau de neurones prenant en compte pour chaque avion deux intrus au lieu d'un seul. F.Médioni montre dans sa thèse à quel point l'apprentissage devient difficile en raison de nombre important de configurations

possibles. Il semble qu'il soit difficile d'obtenir des réseaux à la fois fiables et performants. Médioni a également utilisé une méthode d'apprentissage par algorithmes génétiques utilisant la coévolution de réseaux de neurones et de configurations de conflits. Son travail a montré que la mise en œuvre de telles techniques est très lourde et nécessite beaucoup trop de réglages et de raffinements. [2,4,33,35,36]

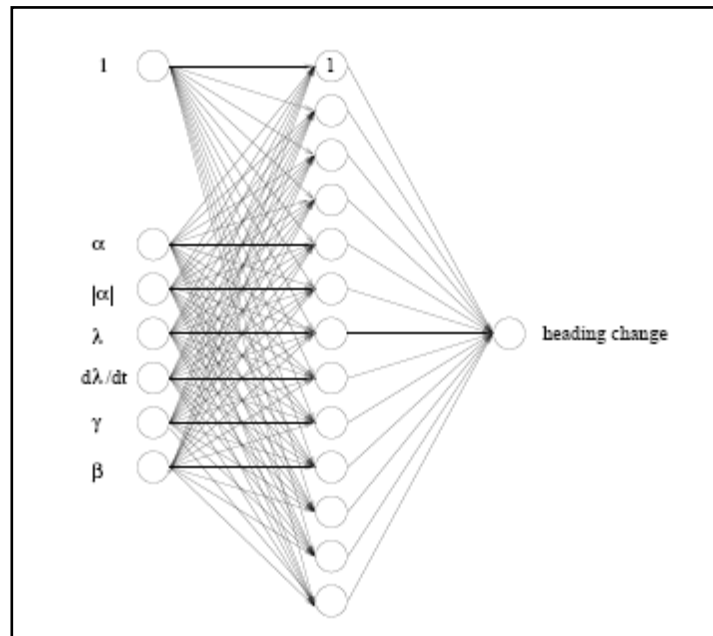


Figure 2-2 : Structure du réseau de neurones

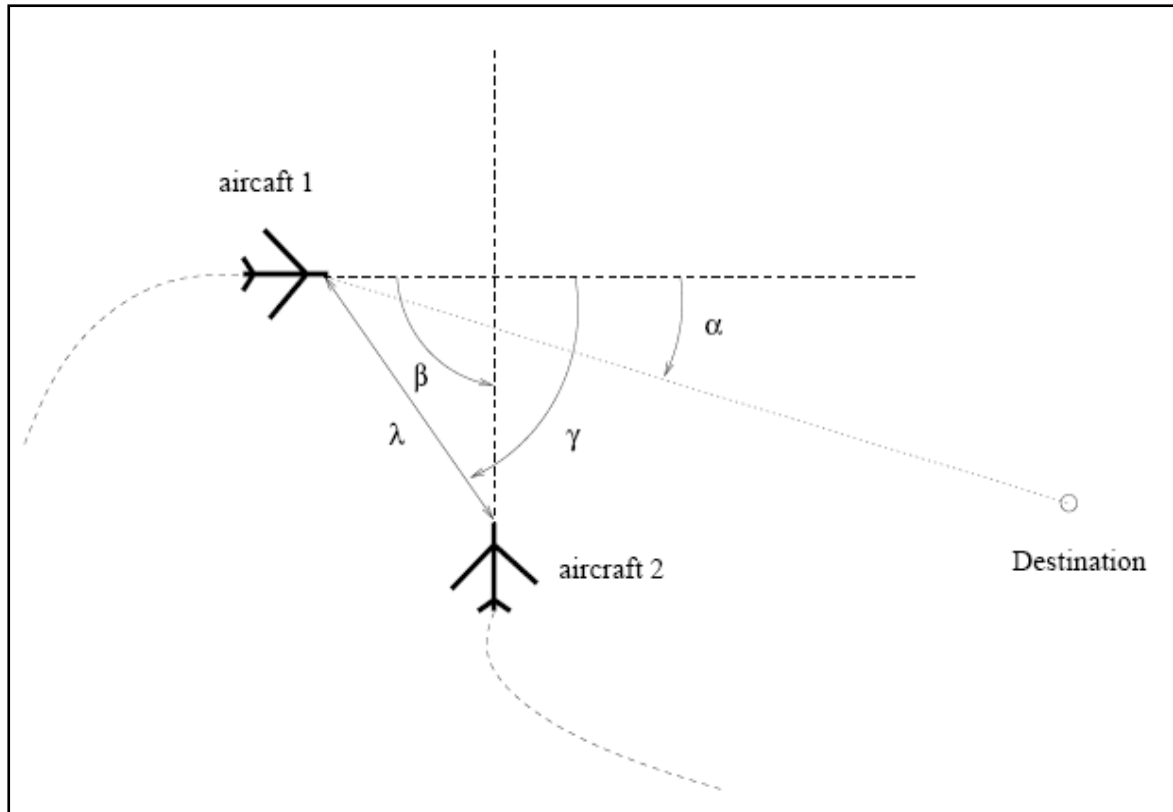


Figure 2-3 : Données en entrée

■ Une méthode utilisant la programmation linéaire

Dans le rapport de DEA, Frédéric Médioni [2,4] met en évidence la possibilité de linéariser le problème de résolution de conflits dans le plan horizontal à condition de modéliser les manœuvres d'évitement par des offsets (l'avion se met sur une trajectoire parallèle à sa trajectoire initiale pendant la durée de la manœuvre). La modélisation suppose que l'avion se déplace à vitesse constante, ce qui est loin d'être le cas dans la pratique. La linéarisation ne résout pas pour autant le problème combinatoire. Pour un conflit à deux avions dans le plan horizontal, l'espace des solutions admissibles est constitué de deux composantes connexes. Pour un conflit à n avions, l'espace des solutions admissibles se décompose en $2^{n(n-1)/2}$ composantes connexes. Il s'agit de définir pour chacune des paires si le relèvement d'un des avions par l'autre évolue dans le sens trigonométrique ou dans le sens des aiguilles d'une montre. De plus la linéarisation requiert que pour chaque avion, on choisisse son sens de déviation (droite ou gauche). Au total, le nombre de programme linéaires à résoudre pour un conflit à n avions est $2^{n(n-1)/2}$ soit par exemple 32768 pour un conflit à 5 avions.

Cette méthode permet de résoudre des cas théoriques assez facilement jusqu'à 5 ou 6 avions. Néanmoins, les hypothèses fortes faites sur la modélisation (vitesses

constantes, pas de prise en compte de l'incertitude) la rendent inexploitable dans un contexte opérationnel.

- Une méthode basée sur la programmation semi-définie

L'idée initiale a été introduite par Eric Féron et a été reprise par Pierre Dodin au LOG. Elle consiste à utiliser un algorithme de programmation semi-définie pour déterminer le sens de résolution de chaque paire d'avions en conflits. Une fois ce sens de résolution déterminé, une méthode d'optimisation convexe sous contraintes convexes permet de calculer les modifications de cap à imposer aux avions. Cette méthode ne permet pas de trouver une solution admissible dans tous les cas. Un processus de bruitage ou randomisation permet d'améliorer son taux de réussite sans toutefois rendre la méthode infaillible. Le cadre très simplifié du modèle choisi (vitesses constantes, avions dans un plan horizontal manœuvrant tous au même moment, pas de prise en compte de l'incertitude) ne laisse que peu d'espoir pour des applications concrètes. [30]

CHAPITRE 3 : METHODES ET TECHNIQUES DE RESOLUTION DE CONFLITS AERIEN

3.1 Introduction

Les trajectoires d'évitement à proposer à des avions doivent répondre à plusieurs critères : elles doivent d'abord, bien entendu, être compatibles avec les performances des avions considérés. Ainsi les vitesses, le taux de montée et de descente, les angles lors des changements de direction sont bornés. Mais elles doivent être aussi simples, pour pouvoir être facilement transmises aux pilotes (que cette transmission soit effectuée par des contrôleurs ou qu'elle soit aussi automatisée), et pour que les pilotes puissent facilement les reproduire. Ainsi le nombre de changements de cap, d'altitude et de vitesse doit être aussi limité. De plus, le confort de passagers, dans le cas d'avions de ligne, est à prendre en compte, et peut entraîner des contraintes supplémentaires portant sur les paramètres déjà évoqués, pour cela on définit les différentes méthodes qui existent pour la résolution de conflit aérien et qui respectent ces paramètres.

Lorsque deux avions sont en conflit, le contrôleur doit modifier la route des avions afin d'assurer le respect des normes de séparation. On peut citer deux principes de résolution souvent utilisés : l'évitement par offset et l'évitement par point tournant.

■ L'évitement par offset

L'évitement par offset (figure 3-1) consiste à faire suivre à l'avion dévié une trajectoire parallèle à sa trajectoire optimale, déclarée par rapport à celle-ci, d'un côté ou de l'autre. L'évitement par offset comporte trois phases ; durant chacune de ces phases la trajectoire de l'avion est un segment de droite :

1. La phase de mise en offset : l'avion est dévié de sa trajectoire suivant un angle appelé angle de mise en offset, jusqu'à ce qu'il atteigne l'écartement désiré par rapport à sa trajectoire d'origine.
2. L'avion suit ensuite une trajectoire parallèle à sa trajectoire d'origine, pendant un temps variable, mais en tous cas suffisant à l'évitement du deuxième avion. La

distance entre la trajectoire déviée et la trajectoire d'origine sera appelée dans la suite de cette partie la valeur de l'offset.

3. L'avion revient sur sa trajectoire initiale.

L'angle mise en offset comme l'angle de retour sur la trajectoire peuvent être fixés. La durée de la seconde phase (temps pendant lequel la trajectoire de l'avion reste parallèle à sa trajectoire) n'a pas d'incidence directe sur l'allongement de la trajectoire. Cependant, il est préférable que ce temps soit le plus court possible, pour les deux raisons suivantes :

- On réduit ainsi le risque que pendant cette phase, la trajectoire de l'avion dévié n'interfère avec celle d'un autre avion.
- Le contrôle aérien est réparti entre plusieurs secteurs, et il est préférable, pour des raisons de coordination entre secteurs, qu'au moment où l'avion passe de l'un de ces secteurs à un autre, il rejoint sa trajectoire initiale. Les secteurs sont par ailleurs conçus de manière à ce que les points de passage de l'un à l'autre soient suffisamment éloignés des balises, au voisinage desquelles, comme on l'a vu, surviennent le plus souvent des conflits. [52,53]

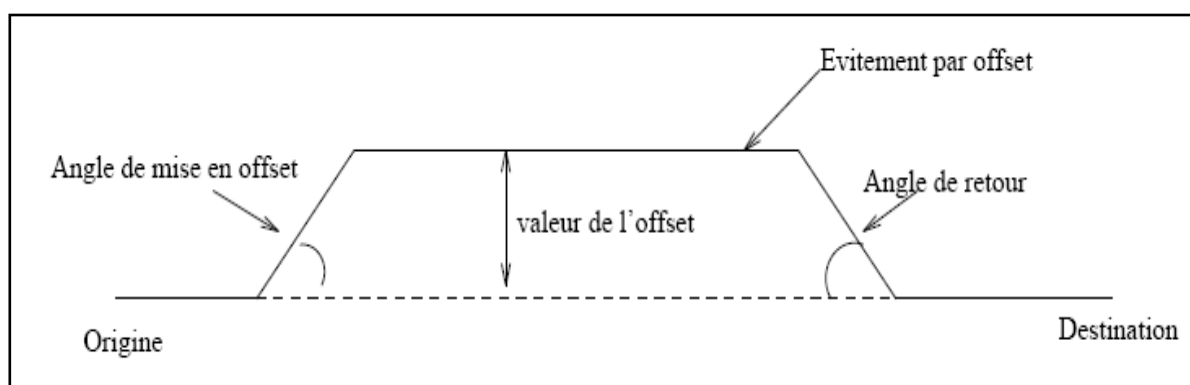


Figure 3-1 : Méthode d'évitement par offset

■ L'évitement par point tournant

Dans ce cas on choisit un point proche de la zone de conflit (figure 3-2) que l'avion dérouté survole avant de rejoindre le point de sortie initial. L'évitement par point tournant entraîne un retard moins important que l'évitement par offset. [52,53]

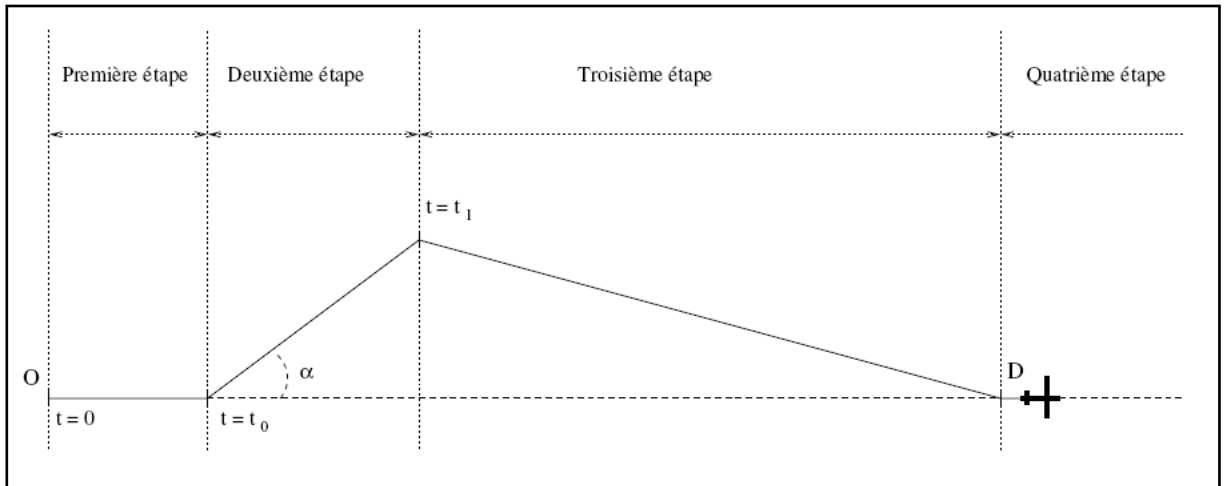


Figure 3-2 : Méthode d'évitement par point tournant

3.2 Les Algorithmes génétiques

3.2.1 Principes généraux

Les algorithmes génétiques sont des algorithmes d'optimisation s'appuyant sur des techniques dérivées de la génétique et de l'évolution naturelle, leurs principes de base s'inspirent, très schématiquement, de l'évolution naturelle. Le vocabulaire employé révèle clairement cette source d'inspiration : on parle de population, de génération, de sélection, de reproduction, de croisement, de mutation et de chromosomes.

Les origines de ces algorithmes remontent à 1962, aux travaux de Holand [17,18,19,20,22,23,25,27,32,11]. Mais ce n'est qu'à partir de 1989 qu'ils ont connu une véritable popularité avec le livre de David E. Goldberg.

Nous décrivons dans cette section le principe général d'un algorithme génétique employé dans le cadre de l'optimisation (minimisation ou maximisation) d'une fonction notée f_{opt} , d'un espace d'état noté E_{opt} dans \mathbb{R} .

L'algorithme génétique utilise une population. Les éléments de cette population, appelé individus, codent des points de l'ensemble E_{opt} . Cette population est générée aléatoirement au début de l'algorithme. On définit une fonction d'adaptation notée f_a , souvent appelé fitness.

Pour l'utiliser on doit disposer des cinq éléments suivants :

1. Un principe de codage de l'élément de population. Cette étape associe à chacun des points de l'espace d'état une structure de données. Elle se place généralement après une phase de modélisation mathématique du problème traité. La qualité du codage de données conditionne le succès des Algorithmes génétiques.

2. Un mécanisme de génération de la population initial. Ce mécanisme doit être capable de produire une population d'individus non homogène qui servira de base pour les générations futures. Le choix de la population initiale est important car il peut rendre plus ou moins rapide la convergence vers l'optimum global.
3. Une fonction à optimiser. Celle-ci appelée aussi fonction de fitness ou fonction d'évaluation de l'individu.
4. Des opérateurs permettant de diversifier la population au cours des générations et d'explorer l'espace d'état. L'opérateur de croisement recompose les gènes d'individus existant dans la population, l'opérateur de mutation a pour but de garantir l'exploration de l'espace d'états.
5. Des paramètres de dimensionnement : taille de la population, nombre totale de génération ou critère d'arrêt, probabilité d'application des opérateurs de croisement et de mutation.

A partir de la population initial, et en utilisant cette fonction d'adaptation, l'algorithme génétique suit les étapes suivantes, illustrés par la figure 3-3 :

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

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

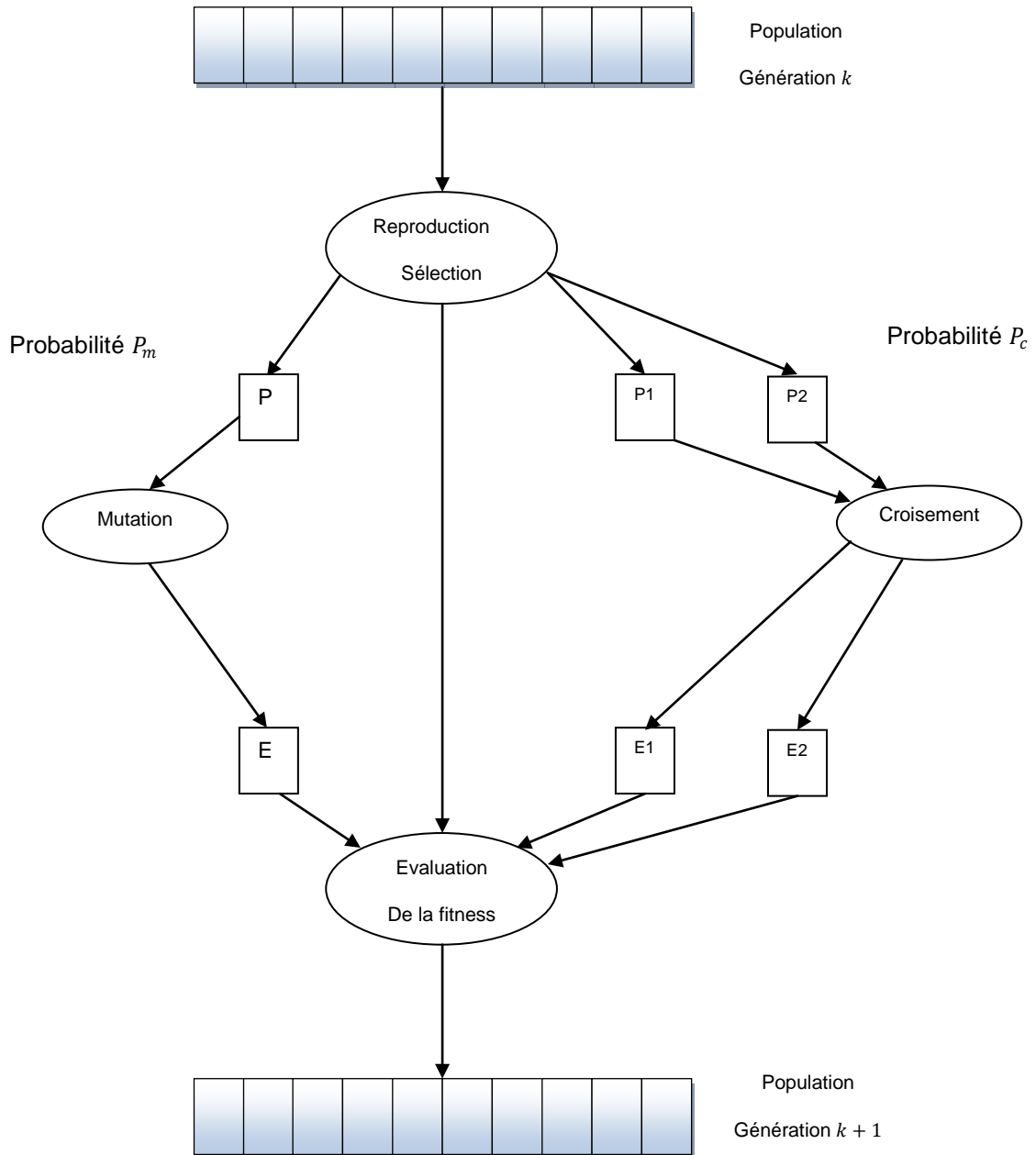


Figure 3-3 : Les étapes de déroulement des Algorithmes génétiques

3.2.2 Description de la résolution par algorithmes génétiques

On utilise les deux modèles de déviation suivants (point tournant, offset), figure 3.4 et figure 3.5, on définit les variables suivantes :

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

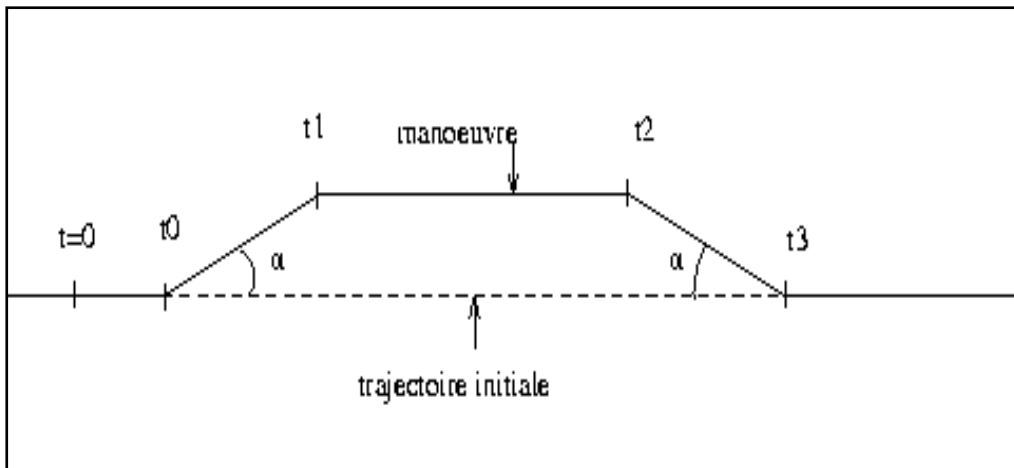


Figure 3-4 : modèle d'offset

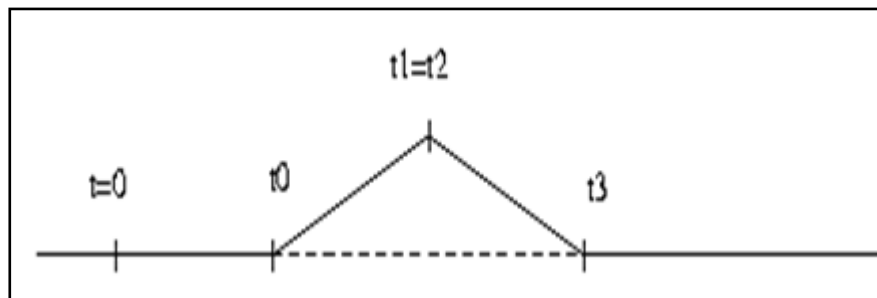


Figure 3-5 : modèle de point tournant

■ Codage

Pour le problème de résolution de conflits, les variables t_0 , t_1 , t_2 et α de chaque avion sont disposés dans une matrice qui constitue un individu de la population (Tableau 3-1). On appellera gènes les variables t_0 , t_1 , t_2 et α de chaque avion. Un élément de la population contient donc $4n$ gènes.

Tableau 3-1 : Codage d'un individu à n avions

t_{01}	t_{11}	t_{21}	t_1	α_1
t_{02}	t_{12}	t_{22}	t_2	α_2
t_{03}	t_{13}	t_{23}	t_3	α_3
t_{04}	t_{14}	t_{24}	t_4	α_4
t_{05}	t_{15}	t_{25}	t_5	α_5
.
.
t_{0n}	t_{1n}	t_{2n}	t_n	α_6

■ Génération de la population initiale

Dans le modèle de résolution par point tournant, l'algorithme de résolution de conflits est ré-initialisé pour chaque cluster tous les Δ minutes. Il se peut donc que l'algorithme ait à prendre en compte des avions déjà engagés dans certaines manœuvres de résolution. Ces avions sont alors soumis à certaines contraintes gérées a priori par une procédure chargée de vérifier leur respect à chaque étape de l'algorithme. Ces contraintes sont les suivantes :

- Lorsqu'un conflit à n avions est fourni à l'algorithme génétique, il se peut que certains gènes correspondent à des manœuvres déjà entamées. 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 façons aléatoires.
- On doit vérifier $t_0 \leq t_1 \leq t_2$.
- Enfin pour que l'algorithme génétique puisse détecter rapidement les solutions sans conflits, il sera fait en sorte que chaque avion de chaque chromosome de la population ait initialement une chance sur trois de ne pas être dévié.

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

■ 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 réalisée.

La fonction fitness est représentée par une matrice triangulaire inférieure F .

Le problème de résolution de conflits est un problème d'optimisation multi critère comme suit :

- Le retard induit pour chaque avion doit être aussi faible que possible.
- Le nombre d'avion déviés et le nombre total de manœuvres doit être aussi faible que possible, il se peut donc que les retards ne soient pas partagés par tous les avions.
- La durée d'une manœuvre doit être aussi faible que possible pour pouvoir rendre le plus tôt possible un avion disponible pour une autre manœuvre.
- Les trajectoires doivent respecter les normes de séparation.

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

- Calcul des éléments de la diagonale :

Les éléments de la diagonale sont calculés par la formule

$$F_{i,i} = S_i + kN_i \quad (3.1)$$

- S_i : La surface occupée par la résolution (figure 3-6)
- N_i : Le nombre de manœuvre
 - $N_i=2$ pour le modèle point tournant
 - $N_i=3$ pour le modèle offset

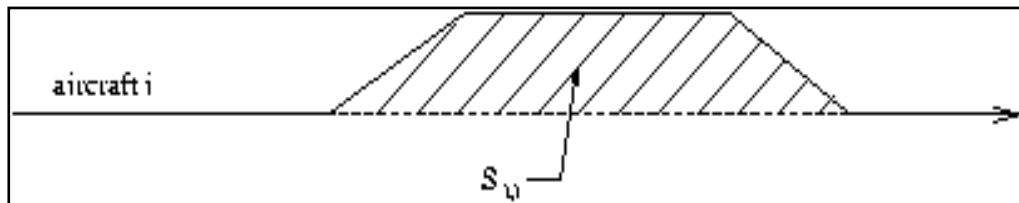


Figure 3-6 : Surface occupé par la manœuvre

- 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 à $F_{i,j}$ une mesure du conflit entre i et j .

$$F_{i,j} = \sum_{t=0}^{\text{temps total}} (C_{t,i,j}) \quad (3.2)$$

La fitness scalaire peut se calculer de la façon suivante :

$$\exists(i,j), F_{i,j} \neq 0 \Rightarrow f = \frac{1}{2 + \sum_{i \neq j} F_{i,j}} \quad (3.3)$$

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 conflits (leurs fitness sont supérieures à 0,5).

■ Opérateur de croisement

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 P_1 et P_2 . On échange ensuite les deux sous-chaines terminales de chacun des

deux chromosomes, ce qui produit deux enfants E_1 et E_2 . Ce type de croisement est représenté par la figure 3-7 suivante :

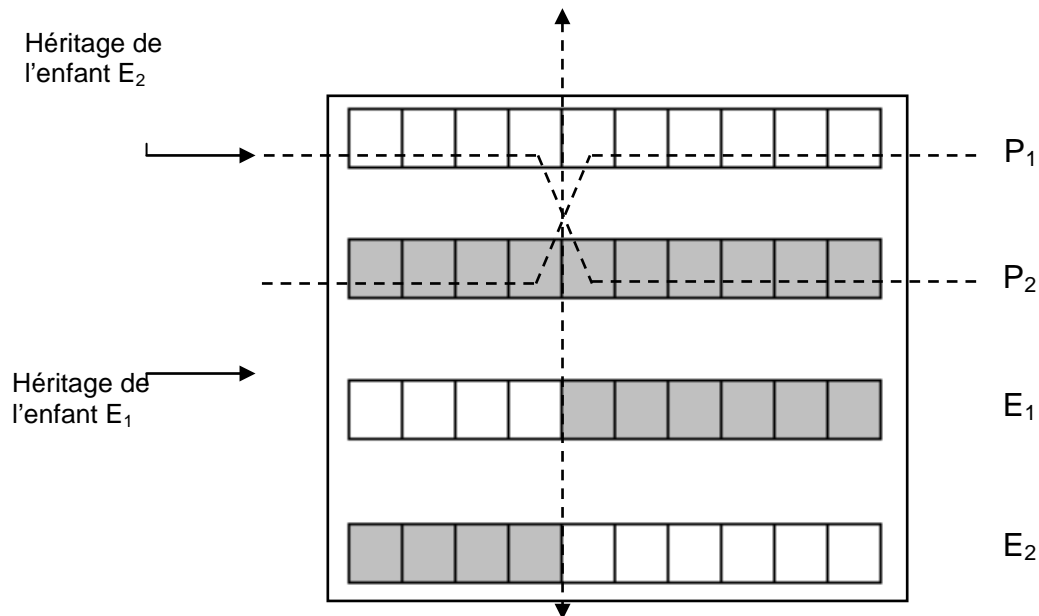


Figure 3-7 : Opérateur de croisement

On peut étendre ce principe en découpant le chromosome non pas en 2 sous chaînes mais en 3, 4 etc comme il est représenté dans la figure 3-8 :

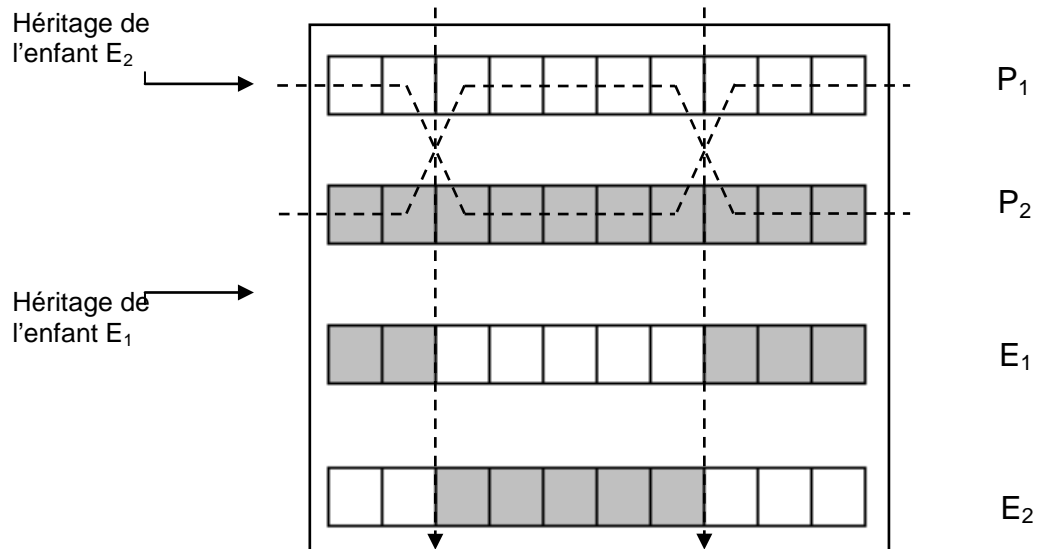


Figure 3-8 : Opérateurs de croisement

■ Opérateur de mutation

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

La propriété d'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 (figure 3-9).

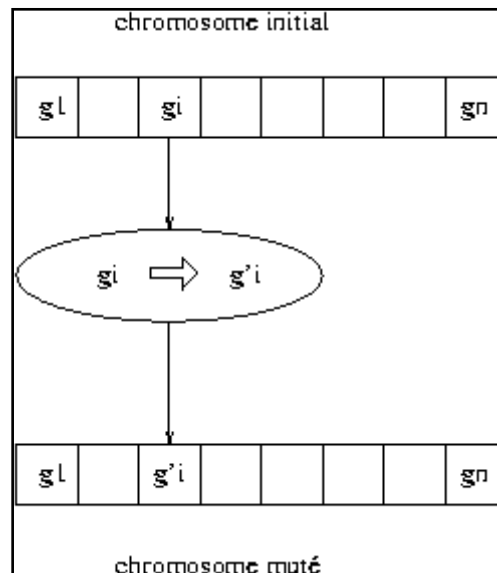


Figure 3-9 : Principe de l'opérateur de mutation

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.

3.3 Méthode des intervalles

Le problème de l'évitement aérien est un problème d'optimisation (minimiser les retards) sous contraintes (les avions devant rester séparés).

L'utilisation de la méthode des intervalles reposera sur le fait qu'elle peut calculer l'ensemble des positions possibles de l'avion à tout instant t donné lorsque les instants de manœuvre t_0 et t_1 ne sont plus définis précisément mais peuvent prendre n'importe quelles valeurs à l'intérieur d'intervalles, appelées intervalles de manœuvres. Elle pourra ainsi déterminer les conséquences de l'utilisation des intervalles de manœuvre sur le respect par les avions des contraintes de séparation, ainsi que sur leurs retards.

Ceci permet d'appliquer au problème de l'évitement des algorithmes d'optimisation utilisant des méthodes des intervalles.

L'analyse d'intervalles a été originellement développée en 1996 par Moore comme un outil permettant de prendre en compte et de contrôler les erreurs d'arrondis dans les calculs numériques effectués sur ordinateurs.

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. [2,7,47,48]

3.3.1 Le principe

Le problème proposé à résoudre est un problème classique d'évitement : minimiser les retards des avions (retards entraînés par les altérations de leurs trajectoires due à l'évitement), sous la contrainte qu'il n'y ait aucun conflit entre les avions, c'est-à-dire que les avions restent toujours séparés deux à deux.

Dans cette partie deux avions sont séparés par une distance d , si sur toute la durée de l'évitement, la distance entre eux reste toujours supérieure ou égale à cette distance d , on dira qu'ils restent séparés, sans préciser de quelle distance, quand ils sont séparés au sens de la définition ci-dessus, c'est-à-dire que la distance entre eux restera supérieure ou égale à la norme de séparation.

Le principe de l'utilisation des méthodes d'intervalles que nous allons expliquer dans cette sections repose sur les points suivants :

- 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 nombres réels, mais comme des intervalles, représentant les intervalles de temps durant 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, mais à l'intérieur d'intervalles de temps.

Nous étudierons dans cette section les deux modélisations pour les trajectoires d'évitement déjà introduites : l'évitement par offset et l'évitement par point tournant. Ces trajectoires sont définies par les instants auxquels prendront place certaines manœuvres déterminées.

3.3.2 Evitement par point tournant

■ Modélisation des trajectoires d'évitement

La trajectoire d'évitement par point tournant que nous étudierons ici est une trajectoire d'évitement normalisée, illustrée par la figure 3-10, qui comporte quatre étapes :

Jusqu'à un temps t_0 , l'avion reste sur sa trajectoire d'origine.

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 .

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 . Il est cependant que ces deux angles aient des valeurs fixées : cette méthode repose sur le fait que l'ensemble des positions possibles d'un avion, quand ces temps de manœuvres prennent leurs valeurs dans des intervalles, ont des formes simples. Dans toute la suite, ces deux angles seront égaux, ce qui permettra d'obtenir des expressions mathématiques plus simples pour les retards et les positions des avions.

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.

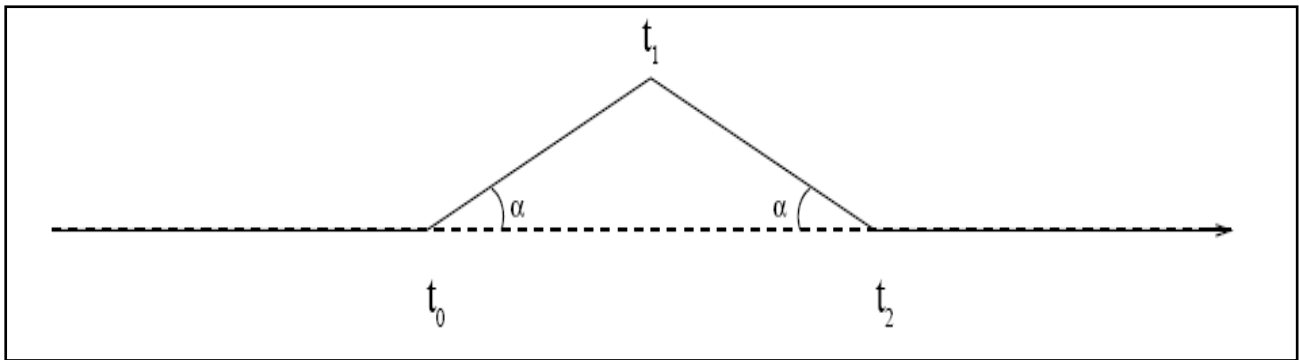


Figure 3-10 : Modélisation d'une trajectoire d'évitement par point tournant

Notons $p(t)$ la position de l'avion au temps t . Par construction de la trajectoire décrite ci-dessus, le triangle $p(t_0)p(t_1)p(t_2)$ est isocèle. La vitesse de l'avion étant supposée constante, on a donc :

$$t_2 - t_1 = t_1 - t_0 \quad (3.4)$$

Ce qui donne :

$$t_2 = 2t_1 - t_0 \quad (3.5)$$

Ainsi définie, la trajectoire d'évitement d'un avion, l'angle α étant fixé, est donc entièrement déterminée par, le sens de la déviation (à droite ou à gauche) et les instant t_0 et t_1 .

On note comme contrainte sur t_1 : $t_1 \geq t_0$. Le cas limite $t_1 = t_0$ correspond en fait à une trajectoire non déviée.

Dans cette technique, seuls t_0 et t_1 seront remplacés par des intervalles, l'angle α prendra des valeurs discrètes, selon le sens de la déviation. Le choix du sens de déviation sera intégré à l'algorithme d'optimisation.

■ Position de l'avion à un temps t

La trajectoire d'évitement d'un avion est définie par le sens de sa déviation et les valeurs de t_0 et t_1 . Le sens de déviation de l'avion est considéré comme fixé. On considère un repère orthonormé dans le plan, tel que l'avion soit sur l'origine O de ce repère au début de l'évitement. L'origine des temps ($t = 0$) correspond au début de l'évitement. Rappelons que ce que nous appelons le début de l'évitement est le moment à partir duquel la déviation peut avoir lieu. C'est donc une borne inférieure pour t_0 . On définit aussi une borne supérieure sur les instants auxquels pourront avoir lieu les manœuvres de l'avion. Cette borne supérieure sera, dans toute la suite, notée t_f . On aura donc :

$$0 \leq t_0 \leq t_1 \leq t_f \quad (3.6)$$

L'avion est de retour sur sa trajectoire d'origine au temps t_2 :

$$t_2 = 2t_1 - t_0 \quad (3.7)$$

La vitesse est notée v .

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

$t \leq t_0$. On a alors :

$$\begin{cases} x(t) = vt \\ y(t) = 0 \end{cases} \quad (3.8)$$

L'avion n'a pas encore été dévié.

$t_0 \leq t \leq t_1$. On a alors :

$$\begin{cases} x(t) = v(t_0(1 - \cos \alpha) + t \cos \alpha) \\ y(t) = (t - t_0) \sin \alpha \end{cases} \quad (3.9)$$

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

$t_1 \leq t \leq t_2$. On a alors :

$$\begin{cases} x(t) = v(t_0(1 - \cos \alpha) + t \cos \alpha) \\ y(t) = (2t_1 - t_0 - t) \sin \alpha \end{cases} \quad (3.10)$$

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

$t \geq 2$. On a alors :

$$\begin{cases} x(t) = v(t - 2(t_1 - t_0)(1 - \cos \alpha)) \\ y(t) = 0 \end{cases} \quad (3.11)$$

L'avion est revenu sur sa trajectoire d'origine

Une fois l'avion revenu sur sa trajectoire d'origine, on peut calculer le retard entraîné par la trajectoire d'évitement définie par les temps de manœuvres t_0 et t_1 . On obtient donc :

$$r(t_0, t_1) = 2(t_1 - t_0)(1 - \cos \alpha) \quad (3.12)$$

■ Conséquences de l'introduction des intervalles

On a $t_0 \in T_0$ et $t_1 \in T_1$, où T_0 et T_1 sont deux intervalles : ($T_0 = [t_{0min}, t_{0max}]$ et $T_1 = [t_{1min}, t_{1max}]$), c'est-à-dire que les instants de manœuvres ne sont plus définis comme des instants ponctuels : on définit deux intervalles de temps (T_0 et T_1), à l'intérieur desquels prendront place les deux premières manœuvres de la trajectoire d'évitement (on aura

$t_0 \in T_0$ et $t_1 \in T_1$). Nous allons calculer l'ensemble des positions possibles d'un avion à un instant t quand ses instants de manœuvres sont donnés par de tels intervalles.

Définition 3-1 : – Positions possibles pour un avion à un instant t – A un instant t , et pour des intervalles de manœuvres T_0 et T_1 donnés, on appelle position possible pour l'avion tout point p du plan tel qu'il existe $t_0 \in T_0$, $t_1 \in T_1$, avec $t_0 \leq t_1$, définissant une trajectoire d'évitement telle que si l'avion suit cette trajectoire, il se trouve en p à l'instant t . L'ensemble des positions possibles à un instant t pour l'avion considéré est noté $P(t)$.

Si les trajectoires d'évitement sont décrites par des intervalles de manœuvres, plusieurs des cas énumérés dans la section précédente peuvent être possible simultanément, selon la position de t relativement à T_0 et T_1 , et de plus les valeurs de $t_0 \leq t_1$ apparaissant dans les équations ci-dessus ne sont plus uniques.

Comme on doit pouvoir avoir $t_0 \leq t_1$, nous considérons, dans cette section, qu'on a toujours $t_{0min} \leq t_{1min}$ et $t_{0max} \leq t_{1max}$.

Pour savoir quelle équation utiliser, parmi les équations, pour calculer l'ensemble des positions possibles de l'avion au temps t , il faut déterminer à quelle étape de la trajectoire d'évitement peut se trouver l'avion au temps t , en fonction des intervalles T_0 et T_1 .

Première étape : l'avion peut être en première étape de l'évitement (c'est-à-dire ne pas avoir encore quitté sa trajectoire d'origine) si et seulement s'il peut exister un t_0 tel que $t \leq t_0$, c'est à dire si et seulement si : $t \leq t_{0max}$

$P_1(t)$ représente l'ensemble des positions possibles pour l'avion au temps t correspondant à la première étape de l'évitement (l'avion n'a pas encore quitté sa trajectoire d'origine). Il faut prendre en compte les positions possibles de l'avion correspondant à la première étape de l'évitement si on a $t \leq t_{0max}$. Si l'avion est encore à la première étape, l'ensemble des positions possibles pour lui est alors réduit à un point dont les coordonnées sont données par les équations (3.8) :

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

Deuxième étape : l'avion peut être à la deuxième étape de l'évitement (c'est-à-dire être en train de s'éloigner de sa trajectoire d'origine) si et seulement s'il existe des valeurs de t_0 et de t_1 telles que $t_0 \leq t \leq t_1$, c'est-à-dire si seulement si :

$$t_{0min} \leq t \leq t_{1max}$$

$P_2(t)$ représente l'ensemble des positions possibles pour l'avion au temps t correspondant à la deuxième étape de l'évitement. Comme on l'a vu, l'avion peut être à

la deuxième étape de l'évitement si et seulement si on a $t_{0min} \leq t \leq t_{1max}$. La possibilité pour l'avion d'être en deuxième étape se traduit par le fait que les coordonnées des positions de l'avion sont données par les équations (3.9) :

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

Avec les contraintes suivantes sur t_0 :

$$t_{0min} \leq t_0 \leq t_{0max} \quad (3.13)$$

Troisième étape : l'avion peut être à la troisième étape de l'évitement (c'est-à-dire être en train de revenir vers sa trajectoire d'origine), si et seulement s'il existe des valeurs de t_0 et de t_1 telles que $t_1 \leq t \leq t_2$, où $t_2 = 2t_1 - t_0$, c'est-à-dire si et seulement si :

$$t_{1min} \leq t \leq 2t_{1max} - 2t_{0min}$$

En effet, $2t_{1max} - t_{0min}$ est la plus grande valeur possible pour t_2 .

On note $P_3(t)$ l'ensemble des positions possibles pour l'avion au temps t correspondant à la troisième étape de l'évitement. Il est possible que l'avion soit en troisième étape si et seulement si on a :

$$t_{1min} \leq t \leq 2t_{1max} - 2t_{0min}$$

Les positions possibles correspondant à cette étape sont alors données par les équations (3.10) :

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

Sous les conditions :

$$\begin{cases} t_{0min} \leq t_0 \leq t_{0max} \\ t_{1min} \leq t_1 \leq t_{1max} \\ t_0 \leq t_1 \leq t \\ t \leq 2t_1 - t_0 \end{cases} \quad (3.14)$$

Quatrième étape : l'avion peut être à la quatrième étape de sa trajectoire d'évitement (c'est-à-dire être revenu sur sa trajectoire d'origine), si et seulement si il peut exister des valeurs de t_0 et t_1 telles que $t_2 \leq t$, avec $t_2 = 2t_1 - t_0$.

Si $t_{0max} \geq t_{1min}$, on peut avoir $t_0 = t_1 = t_2$, ce qui correspond au cas où l'avion n'est pas dévié. La plus petite valeur possible pour t_2 est t_{1min} .

Si $t_{0max} < t_{1min}$, la plus petite valeur possible pour t_2 est $2t_{1min} - t_{0max}$.

On peut résumer ces deux cas en disant que l'avion peut être à la quatrième étape de l'évitement si et seulement si :

$$t \geq \max(t_{1min}, 2t_{min} - t_{0max})$$

Selon la position de t par rapport aux intervalles T_1 et T_0 , l'ensemble $P(t)$ des positions possibles de l'avion à l'instant t peut correspondre à une ou plusieurs des étapes décrites ci-dessus.

Dans les sections suivantes, on supposera que l'instant t est fixé, ainsi que les deux intervalles $T_0 = [t_{0min}, t_{0max}]$, et $T_1 = [t_{1min}, t_{1max}]$.

$P_4(t)$ représente l'ensemble des positions possibles pour l'avion t correspondant à la quatrième et dernière étape de l'évitement (l'avion est revenu sur sa trajectoire d'origine). Au temps t , l'avion peut être à la quatrième étape de l'évitement si et seulement si $t \geq \max(t_{1min}, 2t_{1min} - t_{0max})$.

Si on a $2t_{1max} - t_{0min} - t \leq 0$, toutes les positions possibles de l'avion correspondent à la dernière étape de la trajectoire de l'évitement (l'avion est revenu sur sa trajectoire d'origine).

L'ensemble des positions possibles de l'avion est décrit par les équations (3.11):

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

Avec les conditions :

$$\begin{cases} t_{0min} \leq t_0 \leq t_{0max} \\ t_{1min} \leq t_1 \leq t_{1max} \\ t_0 \leq t_1 \leq t \\ t \geq 2t_1 - t_0 \end{cases} \quad (3.15)$$

■ Ensembles des positions $P(t)$

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

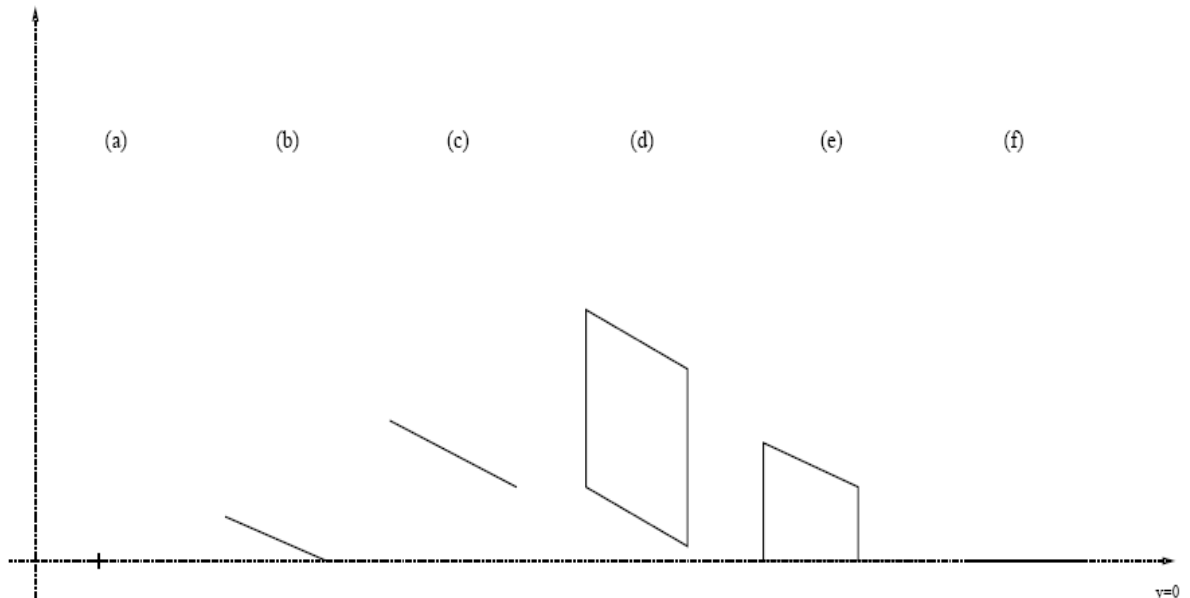


Figure 3-11 : Différentes formes possibles pour l'ensemble $P(t)$

Si $t \leq t_{0min}$, on obtient un point, correspondant à la position de l'avion non dévié. C'est le cas « a » de la figure 3-11.

Si $t_{0min} < t < t_{1min}$, on obtient un segment de droite, correspondant aux positions possibles de l'avion en deuxième étape de l'évitement (l'avion s'écarte de sa trajectoire d'origine. C'est le cas « c » de la figure 3-11. Notons que si $t \leq t_{0max}$, l'avion peut ne pas avoir été dévié, mais le point qu'il occuperait alors est contenu dans le segment de droite correspondant à la deuxième étape de l'évitement. C'est le point limite obtenu pour $t_0 = t$, ($y = 0$), on est alors dans le cas « b » de la figure 3-11.

Le cas « d » de la figure 3-11 montre la situation dans la quelle l'avion ne peut être qu'en troisième étape, tandis que dans le cas « e » l'avion peut être en troisième ou en quatrième étape.

Si $t \geq 2t_{1max} - t_{0min}$, l'avion ne peut être qu'à la quatrième étape de l'évitement, $P(t)$ est un segment de droite. C'est le cas « f » de la figure 3-11.

3.3.3 Evitement par offset

■ Modélisation de la trajectoire d'évitement

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

Jusqu'à un temps t_0 , l'avion reste sur sa trajectoire d'origine.

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 d'origine jusqu'au temps t_1 .

Au temps t_1 , il reprend son cap d'origine, et poursuit sa route sur une nouvelle trajectoire, parallèle à sa trajectoire d'origine.

Un exemple de trajectoire d'évitement par offset est donné sur la figure 3-12 :

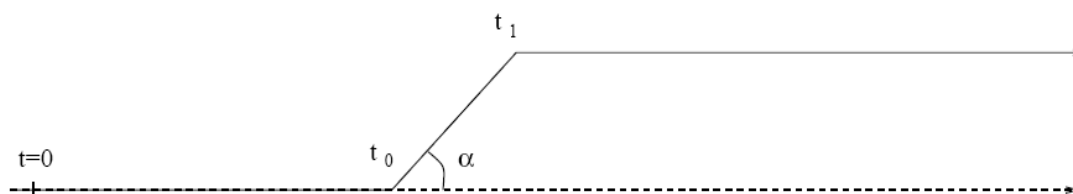


Figure 3-12 : Modélisation d'une trajectoire d'évitement par offset

Ainsi définie, la trajectoire d'évitement d'un avion sera entièrement déterminée par le sens de la déviation (à droite ou à gauche), la valeur de t_0 et la valeur de t_1 , avec $t_1 \geq t_0$.

Aussi, les trajectoires d'évitement par offset sont déterminées par les mêmes types de données que les trajectoires d'évitement par point tournant présentées dans la section précédente.

Ceci permettra d'utiliser le même algorithme avec les deux modélisations. L'algorithme gardera la même structure, seul le calcul à chaque pas de temps de l'ensemble $P(t)$ devra être modifié pour passer d'une modélisation à une autre.

De même que pour la modélisation par point tournant, on supposera que $t_0 \leq t_1$, $t_0 = t_1$ correspondant à une trajectoire non déviée.

■ Position de l'avion à un temps t

Il faut chercher dans un premier temps, pour des valeurs fixées de t_0 et de t_1 , la position $p(t)$ au temps t de l'avion s'il suit la trajectoire d'évitement par offset déterminée par t_0 , t_1 et le sens de déviation en t_0 .

L'origine des temps $t = 0$ correspond au début de l'évitement, et que les manœuvres des avions doivent avoir lieu avant un instant t_f . C'est donc, comme dans le cas de l'évitement par point tournant :

$$0 \leq t_0 \leq t_1$$

La vitesse est notée v . On note : $p(t) = (x(t), y(t))$. On distingue trois cas, correspondant aux trois étapes de la trajectoire d'évitement :

$t \leq t_0$. L'avion n'a pas encore quitté sa trajectoire d'origine, et on a alors :

$$\begin{cases} x(t) = vt \\ y(t) = 0 \end{cases} \quad (3.16)$$

$t_0 \leq t \leq t_1$. On a alors :

$$\begin{cases} x(t) = v(t_0(1 - \cos \alpha) + t \cos \alpha) \\ y(t) = v(t - t_0) \sin \alpha \end{cases} \quad (3.17)$$

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

$t \geq t_1$. L'avion a repris son cap d'origine et suit une trajectoire parallèle à sa trajectoire d'origine. Ses coordonnées sont alors les suivantes :

$$\begin{cases} x(t) = v(t - (t_1 - t_0)(1 - \cos \alpha)) \\ y(t) = v(t_1 - t_0) \sin \alpha \end{cases} \quad (3.18)$$

A partir des équations 3.16 et 3.18, donnez l'expression suivante pour le retard, en fonction de t_0 et t_1 :

$$r(t_0, t_1) = (t_1 - t_0)(1 - \cos \alpha) \quad (3.19)$$

■ Conséquence de l'introduction des intervalles

On a $t_0 \in T_0$ et $t_1 \in T_1$, avec $T_0 = [t_{0min}, t_{0max}]$ et $T_1 = [t_{1min}, t_{1max}]$.

On peut établir, en fonction de t , T_0 et T_1 , les conditions nécessaires et suffisantes pour que l'avion puisse être à une étape donnée de la trajectoire d'évitement par offset.

Première étape : il est possible que l'avion soit à l'instant t à la première étape si et seulement s'il existe $t_0 \in T_0$ tel que $t \leq t_0$, c'est-à-dire si et seulement si :

$$t \leq t_{0max} \quad (3.20)$$

Deuxième étape : il est possible qu'il soit à la deuxième étape si et seulement s'il existe $t_0 \in T_0$ tel que $t \leq t_0$, et $t_1 \in T_1$ tel que $t \geq t_1$, c'est-à-dire si et seulement si :

$$t_{0min} \leq t \leq t_{1max} \quad (3.21)$$

Troisième étape : il est possible qu'il soit à la troisième étape si et seulement s'il existe $t_1 \in T_1$ tel que $t \geq t_1$, c'est-à-dire si et seulement si :

$$t \geq t_{1min} \quad (3.22)$$

■ Construction de l'ensemble $P(t)$

Les deux premières étapes de l'évitement par offset sont en fait identiques aux deux premières étapes de l'évitement par point tournant. Si on note $P_i(t)$ l'ensemble des positions possibles de l'avion correspondant à l'étape (i), les ensembles $P_1(t)$ et $P_2(t)$ sont donc les mêmes que ceux qui ont été définis dans la sections 3-3-2.

L'ensemble $P_3(t)$ est l'ensemble des points $p(t)$ dont les coordonnées $(x(t), y(t))$ sont données par les équations :

$$\begin{cases} x(t) = v(t - (t_1 - t_0))(1 - \cos \alpha) \\ y(t) = v(t_1 - t_0) \sin \alpha \end{cases} \quad (3.23)$$

Avec les conditions :

$$\begin{cases} t_{0min} \leq t_0 \leq t_{0max} \\ t_{1min} \leq t_1 \leq t_{1max} \\ t_0 \leq t_1 \leq t \end{cases} \quad (3.24)$$

3.4 Les Algorithmes A* (AStar)

Le principe de l'algorithme A*, consiste à rechercher le meilleur chemin dans un arbre d'états, en redémarrant à chaque itération du meilleur noeud rencontré pendant la recherche. Ce principe nécessite une fonction de coût pour les transitions entre états, et une heuristique permettant d'estimer au mieux le coût des transitions restant entre le noeud courant et la meilleure solution possible dans le sous-arbre considéré. L'algorithme gère une file d'attente à priorités, dans laquelle les noeuds sont insérés de façon ordonnée, en fonction du coût des transitions passées et de la valeur de l'heuristique. Le noeud de plus haute priorité (de plus bas coût), est retiré de la tête de la file d'attente, ses noeuds fils sont calculés, ainsi que le coût des transitions correspondantes, et les valeurs d'heuristique. Les noeuds fils sont ensuite insérés dans la file d'attente et le processus est alors ré-itéré jusqu'à ce que l'on atteigne une feuille de l'arbre. Celle-ci est lors une solution de coût minimal.

Pour notre problème, l'algorithme A* sera appliqué successivement à chaque trajectoire, en cherchant un chemin de coût minimal qui évite les trajectoires précédentes.

Les états représenteront des choix de déviations éventuelles, faits à chaque étape de la trajectoire. Pour parcourir l'arbre, il nous faut définir les règles permettant de

produire des noeuds fils à partir d'un noeud père. Dans notre cas, les noeuds fils sont simplement les différentes possibilités pour la prochaine étape de la trajectoire.

Si une interférence est détectée entre le segment de trajectoire correspondant au noeud fils et une des trajectoires précédentes, le noeud est rejeté. Voir le chapitre 4 (faire un résumé sur les algorithmes A*)

CHAPITRE 4 : L'ALGORITHME A* (ASTAR)

4.1 Introduction

Nous allons présenter dans ce chapitre l'algorithme A* (AStar) qui va être appliqué dans la suite de notre travail au problème du conflit aérien.

Au premier abord, on peut dire que pour trouver le plus court chemin d'un point A à un point B, il faut commencer par ce qui dirige vers la destination. C'est justement l'idée qu'utilise l'algorithme A*. L'idée est très simple : à chaque itération de cet algorithme itératif, on va tenter de se rapprocher vers la destination. On va donc privilégier les possibilités directement les plus proches de la destination, en mettant de côté toutes les autres.

L'algorithme A* aborde donc le problème de trouver la bonne voie du point de départ à l'objectif, en évitant les obstacles, et en réduisant les coûts (carburant, temps, distance, et argent, etc).

Le mouvement pour un seul objet semble facile mais avec l'algorithme A* est complexe. Considérons la situation suivante (figure 4.1) :

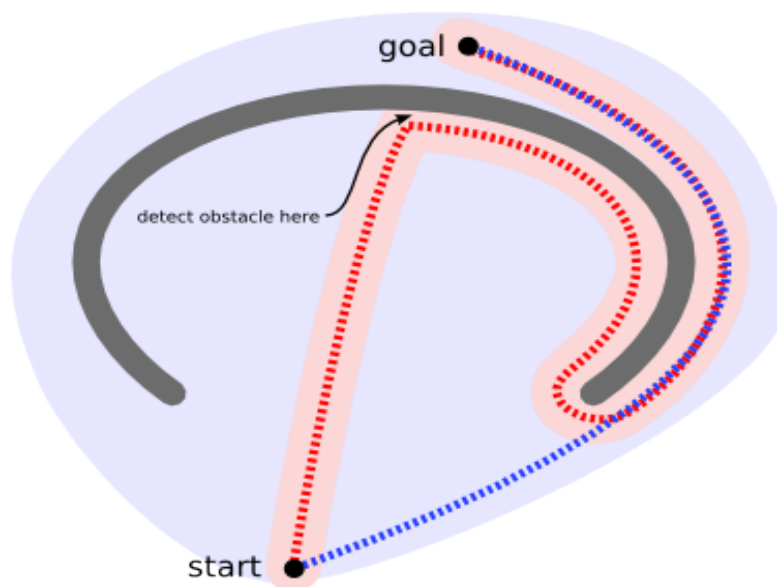


Figure 4-1: Exemple d'observation du A*

Le point de départ est en bas de la carte et cherche à atteindre le sommet. Il n'y a rien dans la zone scannée (en rose) pour indiquer que le point ne peut pas bouger, alors il continue son chemin. Près du sommet, il détecte un obstacle et change de direction. Il trouve alors son chemin de contourner l'obstacle en forme de "U", et continue son chemin (en rouge). En revanche, un éclaireur aurait scannée une plus grande surface (en bleu clair), et aura un plus court chemin (bleu).

4.2 Les étapes intermédiaires de la création du A*

4.2.1 L'algorithme de Dijkstra et l'algorithme « Best-First-Search »

L'algorithme de Dijkstra travaille en visitant les sommets dans le graphe et en commençant par l'objet du point de départ. Il étudie ensuite à maintes reprises les sommets les plus proches qui ne sont pas encore examinés, et l'ajoutant ensuite à l'ensemble de sommets examinés. Il prend de l'expansion vers l'extérieur du point de départ jusqu'à ce qu'il atteigne l'objectif. L'algorithme de Dijkstra garanti de trouver un court chemin d'un point de départ à l'objectif, aussi longtemps qu'aucun des bords n'a un coût négatif. Dans le diagramme ci-dessous (figure 4.2), le carré rose est le point de départ, le carré violet est le but, et les cas en bleu montrent les zones balayées par l'algorithme de Dijkstra. Les carreaux les plus clair dans ces domaines sont les plus éloignés du point de départ, et constituent donc la «frontière» de l'exploration: [50]

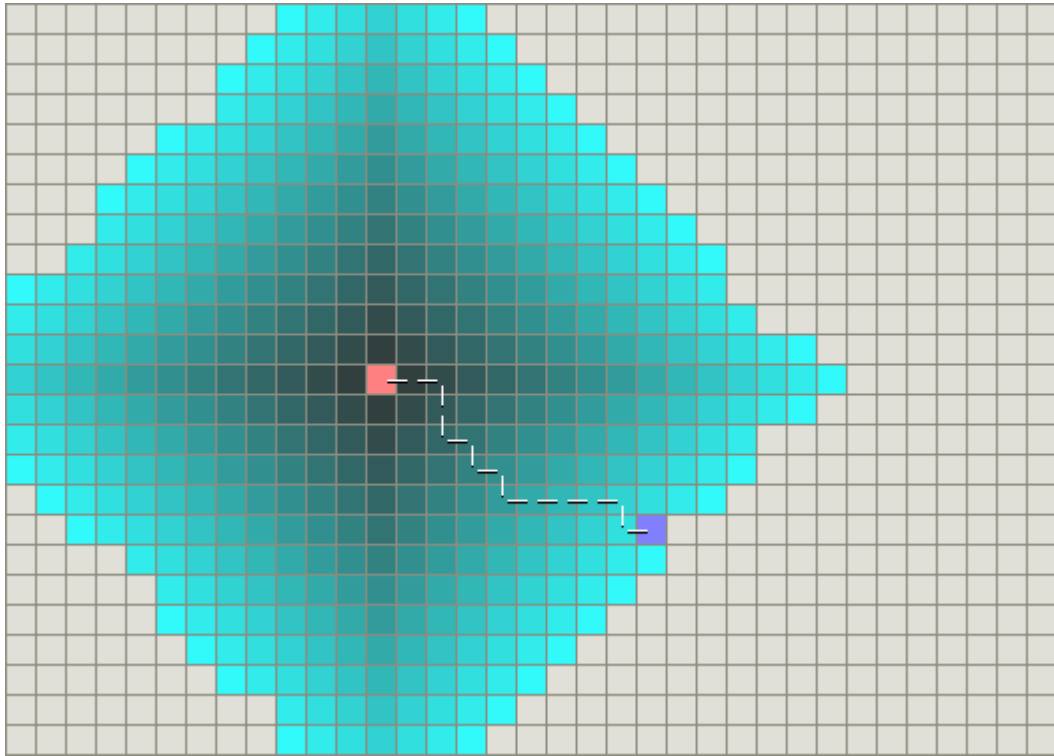


Figure 4-2 : Trajectoire définie par l'algorithme de Dijkstra dans l'absence d'un obstacle

L'algorithme « Best-First-Search (BFS) » fonctionne de manière similaire, sauf qu'il utilise une estimation (appelée heuristique) pour se rapprocher du point final. Au lieu de sélectionner le sommet le plus proche du point de départ, il choisit le sommet le plus proche de l'objectif. L'algorithme BFS ne garanti pas de trouver un plus court chemin. Toutefois, il fonctionne beaucoup plus rapidement que l'algorithme de Dijkstra, car il utilise la fonction heuristique pour guider son chemin vers le but très rapidement. Par exemple, si l'objectif est au sud de la position de départ, l'algorithme BFS aura tendance à se concentrer sur les chemins qui mènent vers le sud. Dans le diagramme ci-dessous, le jaune représente les nœuds avec une forte valeur heuristique (coût élevé pour atteindre l'objectif) et le noir représente les nœuds avec une faible valeur heuristique (faible coût pour obtenir de l'objectif). Il montre que l'algorithme BFS peut trouver les chemins très rapidement par rapport à l'algorithme de Dijkstra : [50]

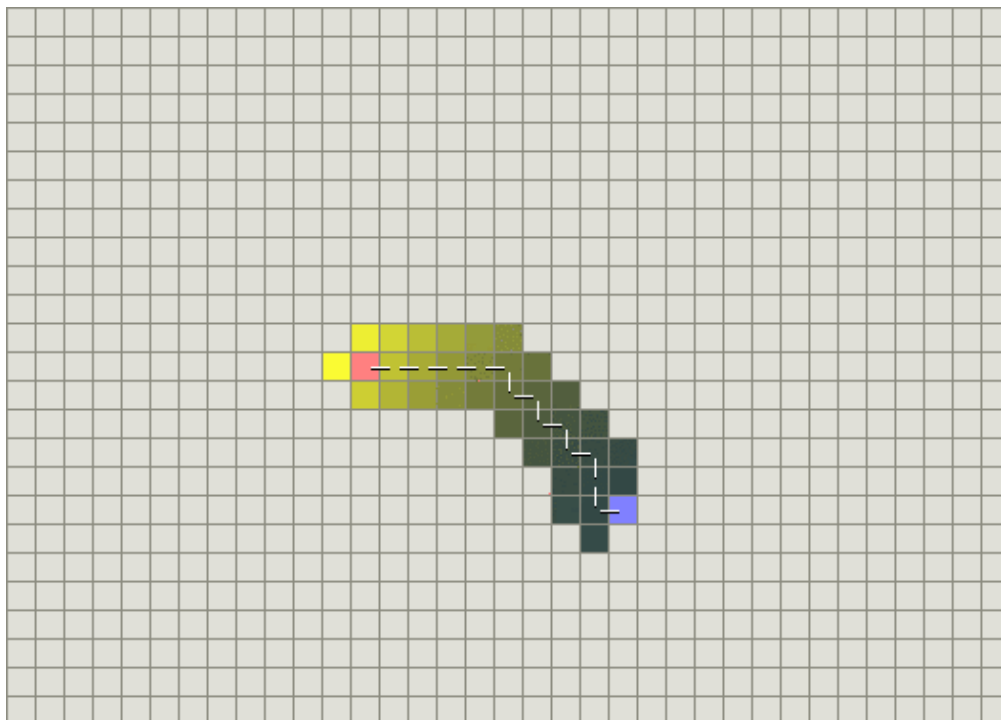


Figure 4-3 : Trajectoire définie par l'algorithme de « Best-First-Search » dans l'absence d'un obstacle

Toutefois, ces deux exemples illustrent le cas le plus simple, quand la carte n'a pas d'obstacles, et le chemin le plus court est vraiment une ligne droite. Prenons l'obstacle concave décrit précédemment. L'algorithme de Dijkstra met plus de temps mais il garanti de trouver un plus court chemin : [50]

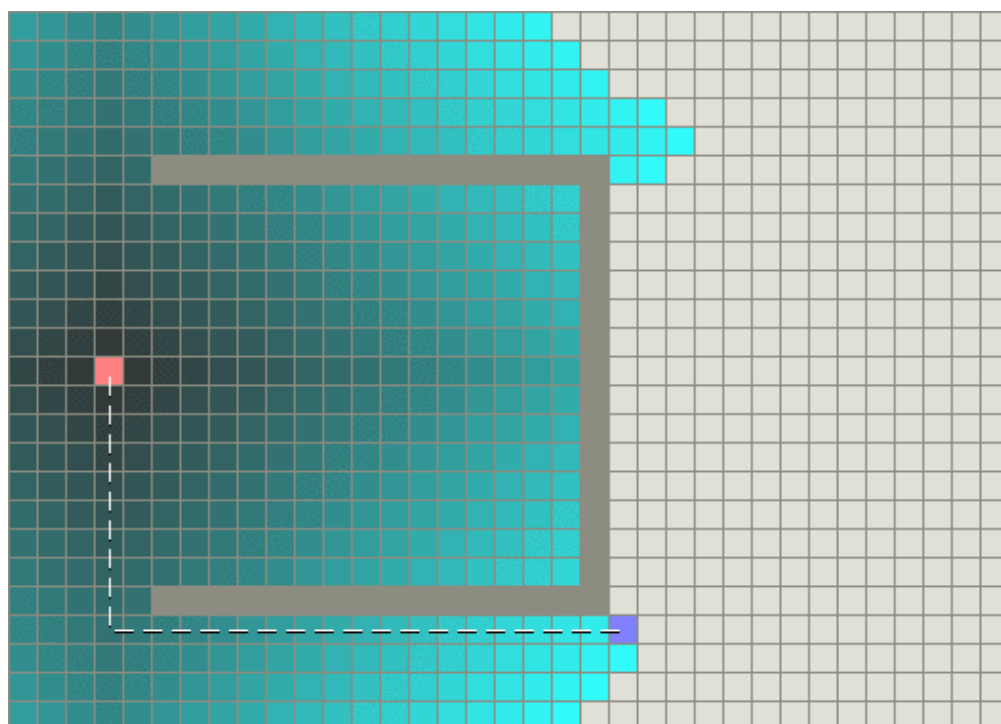


Figure 4-4 : Trajectoire définie par l'algorithme de Dijkstra dans la présence d'un obstacle

L'algorithme BFS en revanche prend moins de temps, mais son chemin n'est pas aussi bon :

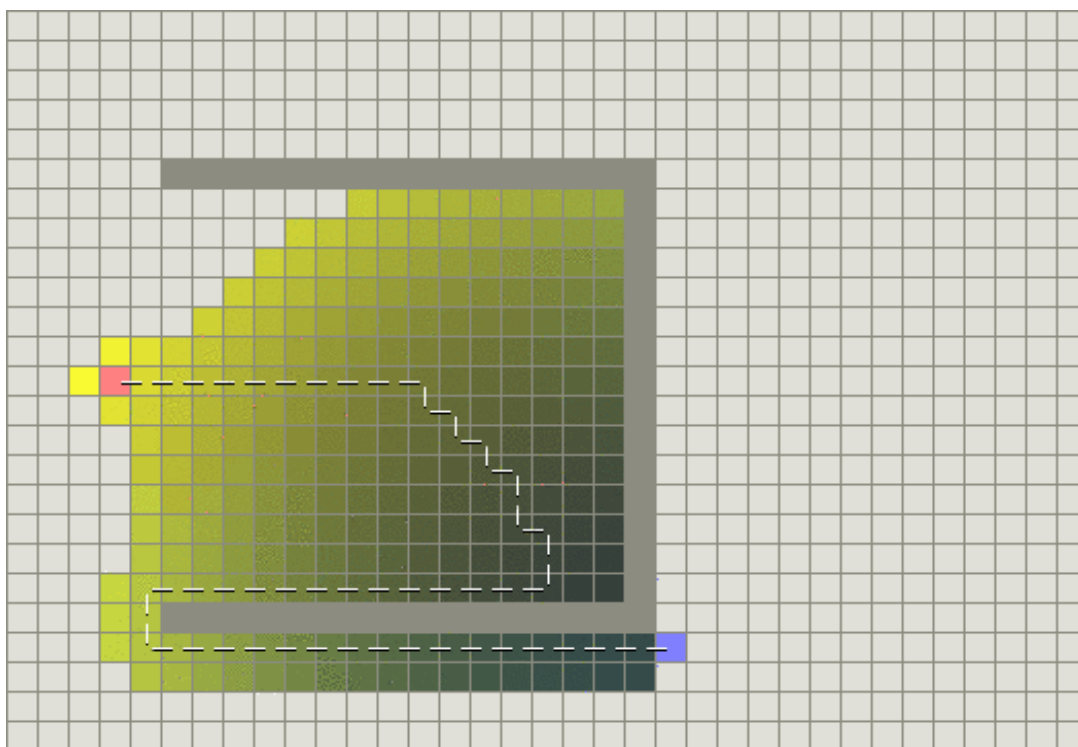


Figure 4-5 : Trajectoire définie par l'algorithme de « Best-First-Search » dans la présence d'un obstacle

Le problème est que l'algorithme BFS est gourmand en mémoire et essaie de se rapprocher de l'objectif, même si ce n'est pas la bonne voie. Étant donné qu'il ne considère que le coût pour se rendre à l'objectif et ne tient pas compte du coût de la voie déjà franchie, il continue en avant même si le chemin sur lequel est devenue vraiment très long. [50]

L'algorithme A* a été développé en 1968 pour combiner des approches heuristiques comme celles de l'algorithme BFS et des approches formelles, comme l'algorithme de Dijkstra.

4.2.2 L'algorithme A*

L'algorithme A* est similaire aux autres algorithmes de recherche de graphique (algorithme de Dijkstra et Best-First-Search).

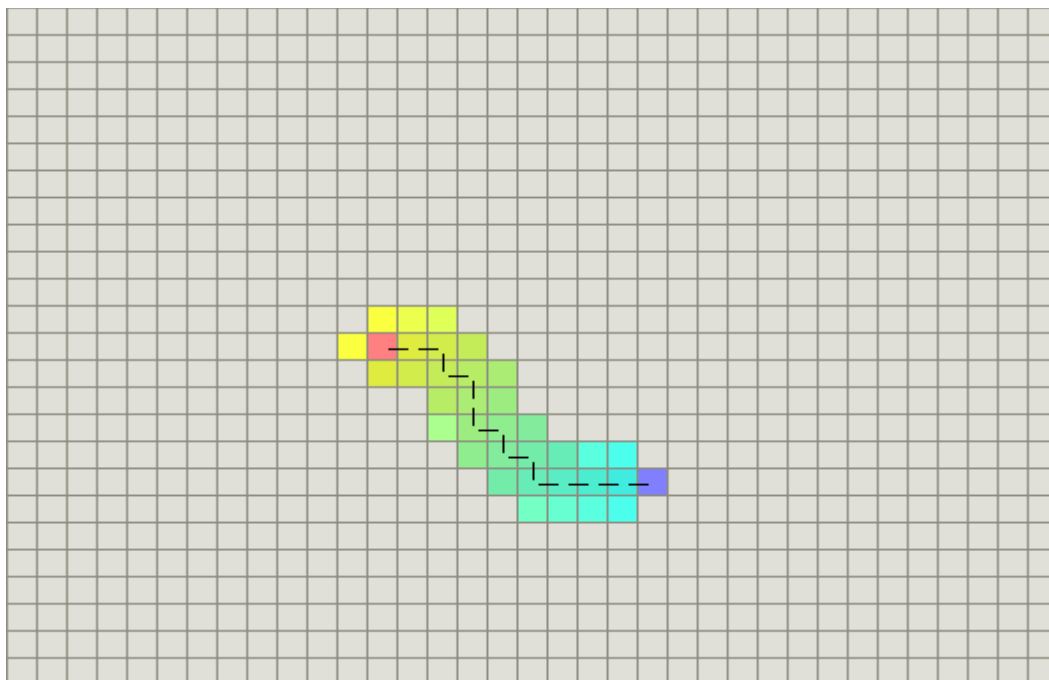


Figure 4-6 : Trajectoire définie par l'algorithme A* dans l'absence d'un obstacle

Dans l'exemple d'un obstacle concave, A* trouve une voie aussi bonne que ce que l'algorithme de Dijkstra a trouvé:

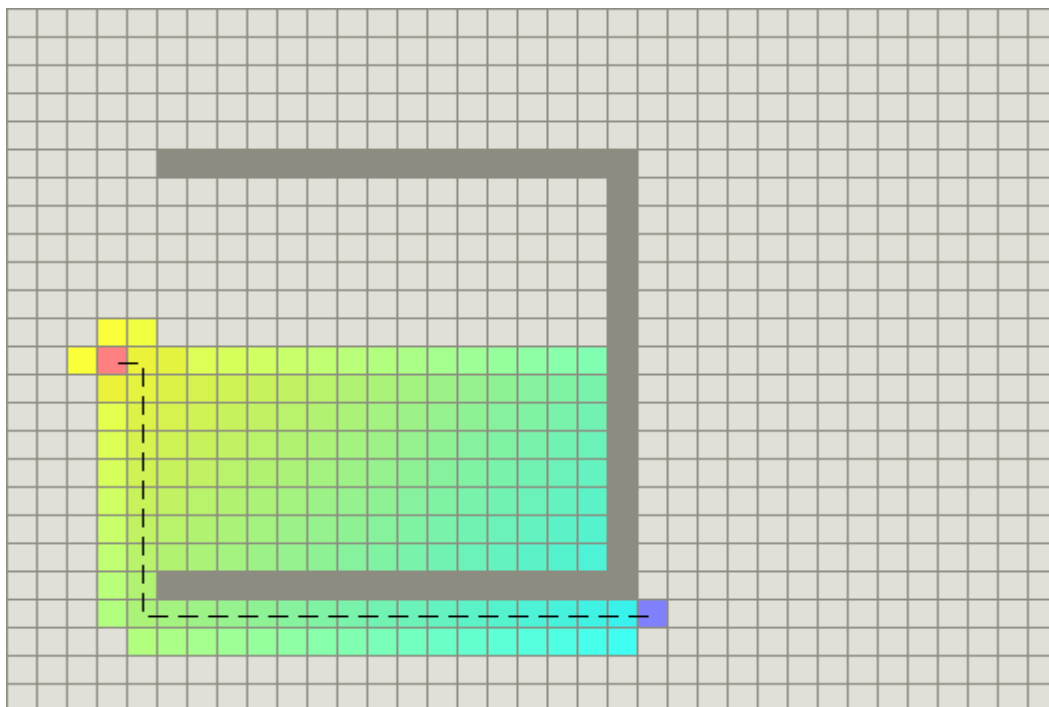


Figure 4-7 : Trajectoire définie par l'algorithme A* dans la présence d'un obstacle

Le secret du succès de l'algorithme A* est qu'il peut combiner entre les éléments d'information qu'utilise l'algorithme de Dijkstra (favorisant les sommets qui sont proches du point de départ) et les informations que l'algorithme BFS utilise (favorisant les sommets qui sont proches de l'objectif). Dans la terminologie standard lorsque on parle de A*, $g(n)$ représente le coût de la trajectoire du point de début jusqu'à n'importe quel sommet n , et $h(n)$ représente le coût estimé de l'heuristique de n'importe quel sommet n jusqu'au point d'arrivée.

Dans les diagrammes ci-dessus, le jaune (h) représente les sommets éloignés de l'objectif et le bleu (g) représente les sommets éloignés du point de départ. A* équilibre entre les deux, comme elle on se déplace du point de départ à l'objectif. Chaque fois on examine le sommet n qui a le plus faible f qui est la somme de g et de h :

$$f(n) = g(n) + h(n) \quad (4.1)$$

4.3 L'heuristique

La fonction heuristique $h(n)$ de A* indique une estimation du coût minimum du sommet jusqu'à l'objectif n . Il est important de choisir une fonction heuristique convenable.

4.3.1 L'heuristique pour une carte de grille

Prenons l'exemple d'une carte de grille, pour expliqué les différentes fonctions de l'heuristiques.

■ Distance de Manhattan

L'heuristique standard est la distance de Manhattan. La fonction coût est la distance minimal D pour le déplacement d'un espace à un espace adjacent. Par conséquent, l'heuristique dans notre cas doit être D fois la distance de Manhattan:

$$h(n) = D \cdot (|x_{depart} - x_{destination}| + |y_{depart} - y_{destination}|) \quad (4.2)$$

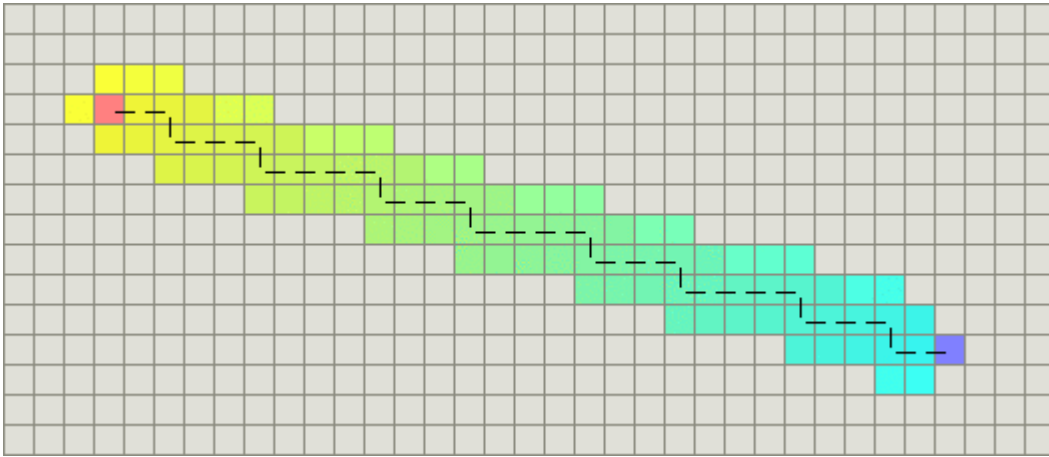


Figure 4-8 : Trajectoire de la fonction heuristique en utilisant la distance de Manhattan

■ Distance diagonale

Dans le cas où les mouvements en diagonale sont autorisés sur la carte, on a besoin d'une heuristique différente de celle de la distance de Manhattan, pour cela il existe une autre fonction heuristique de calcul de la distance diagonale. Cette fonction gère les mouvements en diagonale qui représentent le coût : [50]

$$h(n) = D \cdot \min(|(x_{depart} - x_{destination})|, |(y_{depart} - y_{destination})|) \quad (4.3)$$

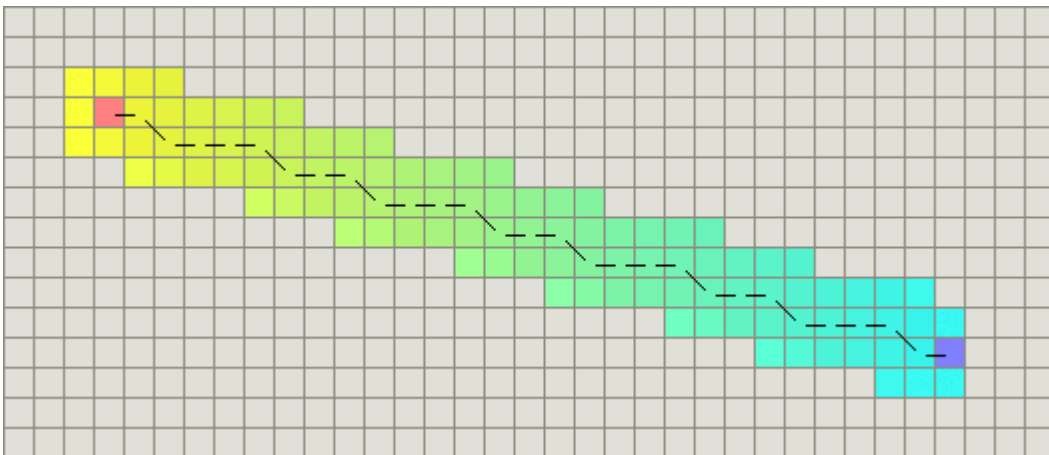


Figure 4-9 : Trajectoire de la fonction heuristique en utilisant la distance diagonale

Si la fonction de la distance diagonale seule ne peut pas résoudre le problème de fonction coût dans une carte de grille, il faut pour cela chercher quelque chose de plus sophistiqué.

Ici, $h_{diagonal}(n)$, représente le nombre de mesure qu'on peut prendre le long d'une diagonale, $h_{straight}(n)$ est la distance de Manhattan. En combinant les deux on trouve la fonction que l'on cherche à déterminer comme il est démontré ci-dessous :

$$D_2 = \sqrt{2} \cdot D \quad (4.4)$$

$$h_diagonal(n) = \min(|(x_{depart} - x_{destination})|, |(y_{depart} - y_{destination})|) \quad (4.5)$$

$$h_straight(n) = (|(x_{depart} - x_{destination})| + |(y_{depart} - y_{destination})|) \quad (4.6)$$

$$h(n) = D_2 \cdot h_diagonal(n) + D \cdot (h_straight(n) - 2 \cdot h_diagonal(n)) \quad (4.7)$$

■ Distance Euclidienne

Si notre mobile est autorisé à déplacer avec n'importe quel angle (même autre espace de recherche qu'une carte de grille), on peut utiliser une ligne droite avec n'importe quel angle, et on calcule la distance qui peut représenter le coût avec la fonction :

$$h(n) = D \cdot \sqrt{((x_{depart} - x_{destination})^2 + (y_{depart} - y_{destination})^2)} \quad (4.8)$$

La distance euclidienne est plus courte que la distance de Manhattan ou la diagonale, donc elle est toujours meilleur pour avoir les plus courts chemins, mais elle utilise plus de temps pendant l'exécution :

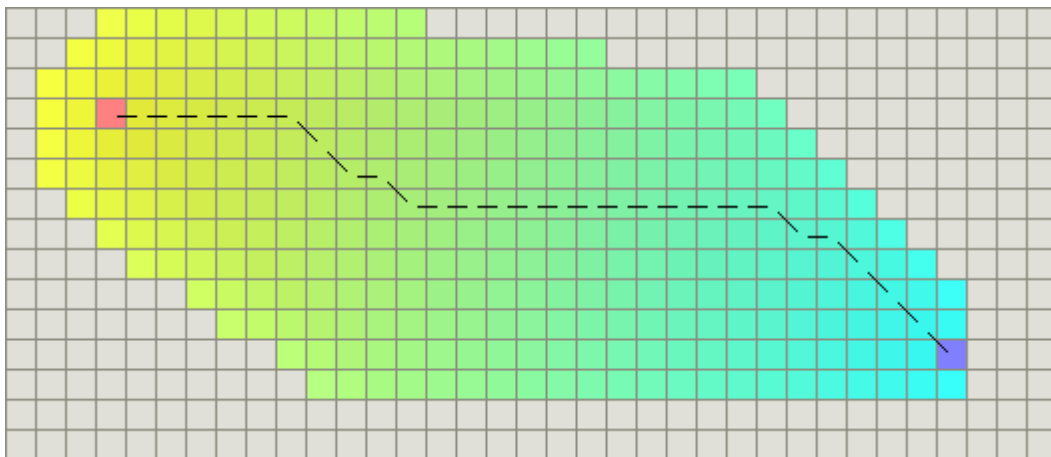


Figure 4-10 : Trajectoire de la fonction heuristique en utilisant la distance euclidienne

■ La distance Euclidienne au carré

Suivant le problème qui peut être posé, on peut utiliser une fonction heuristique qui est la distance euclidienne au carré :

$$h(n) = D \cdot ((x_{depart} - x_{destination})^2 + (y_{depart} - y_{destination})^2) \quad (4.9)$$

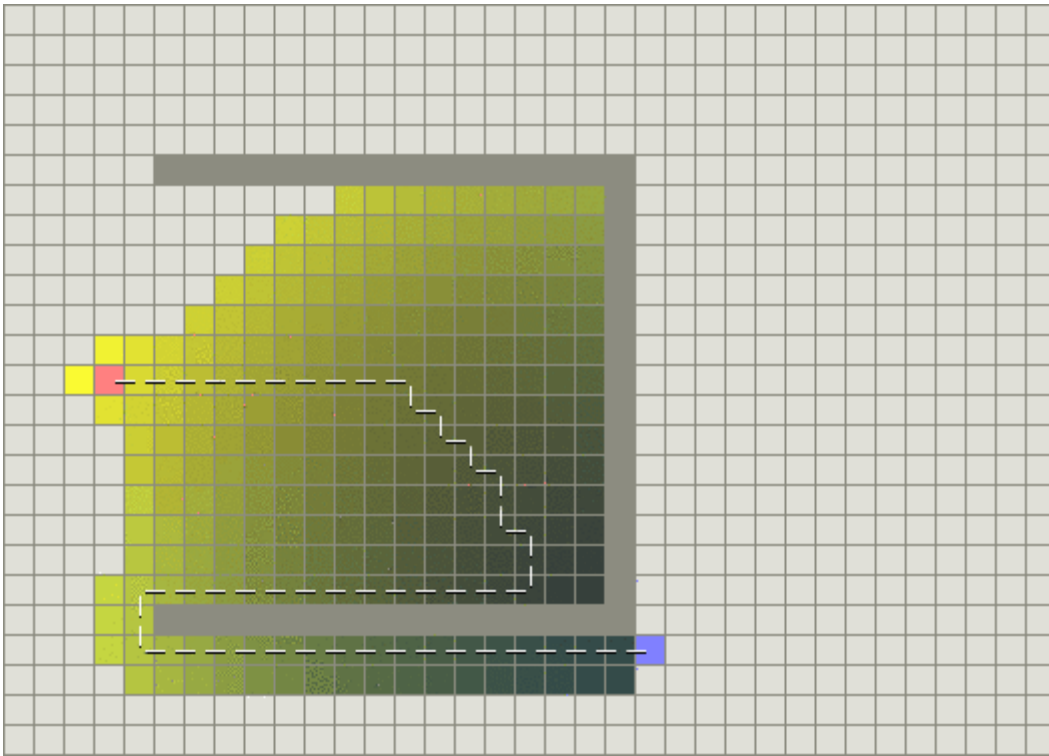


Figure 4-11 : Trajectoire de la fonction heuristique en utilisant la distance euclidienne au carré

4.4 Etude d'application de l'algorithme A*

4.4.1 Zone de recherche

Supposons qu'on veut aller d'un point A à un point B. Considérons également qu'un mur est situé entre ces deux points. Cette situation est illustrée par la figure 4-13, où le point A (départ) est affiché en vert, le point B (arrivée) est en rouge, et où les cases en bleu représentent le mur.

La première chose qu'on remarque est que la zone de recherche est divisée en cases. Simplifier la zone de recherche, comme nous l'avons fait ici, est la première étape. Cette méthode particulière réduit notre zone de recherche à un simple tableau à deux dimensions. Chaque élément du tableau représente une case de notre grille, et son statut est notifié comme étant « traversable » ou « non traversable ». Le chemin va être la succession de cases à parcourir pour passer de la case A à la case B. Une fois le chemin trouvé, notre personnage va passer du centre d'une case au centre de la prochaine case, jusqu'à ce qu'il atteigne sa cible.[12,13,49]

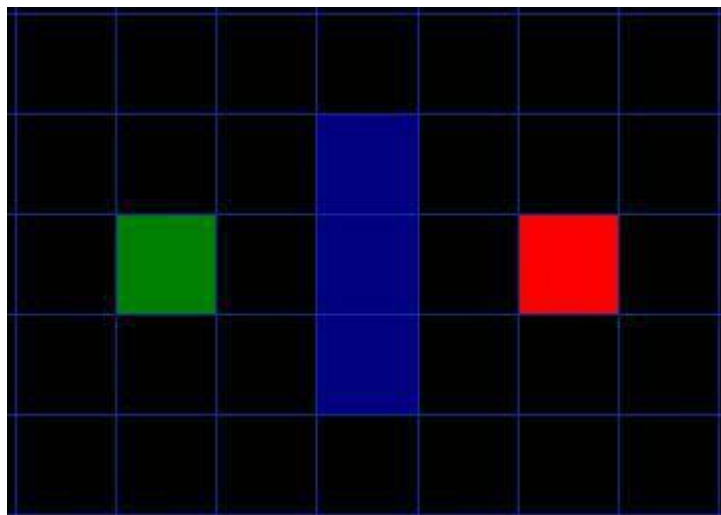


Figure 4-12 : Zone de recherche du chemin

4.4.2 La notion de recherche

Nous commençons notre recherche de la manière suivante :

Commençons au point de départ et ajoutons le à une "liste ouverte" de cases à étudier. Pour l'instant, nous n'avons qu'un seul point à l'intérieur (le point A). la liste ouverte contient une liste de cases qui pourraient éventuellement faire partie de notre chemin, mais pas forcément. En fait, c'est une liste de points que nous devons vérifier. Regardons maintenant toutes les cases traversables adjacentes au point de départ, en ignorant les cases avec les murs, ou toute case qu'on ne peut pas traverser. On les Ajoute à la liste ouverte également. Pour chacune de ces cases, on enregistre la case A comme son "parent". Cette notion de « parent » est très importante dans la suite pour pouvoir retracer le chemin.

On supprime maintenant le point A de la "liste ouverte", et on l'ajoute à une "liste fermée", qui va contenir les points que nous n'aurons plus besoin de vérifier.

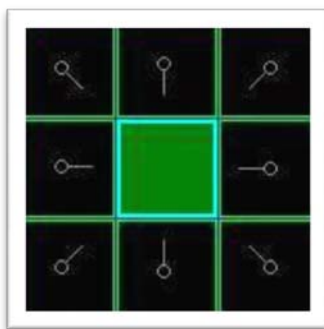


Figure 4-13 : Observation de la notion de parent

A cette étape, on aura une situation comme la figure 4-14 ci-dessus. Sur ce schéma, le carré vert foncé au centre est notre case de départ. Il est entouré de bleu clair pour signifier que la case a bien été ajoutée à notre « liste fermée ». Toutes les cases adjacentes sont maintenant dans la "liste ouverte" de case à vérifier, et sont entourées de vert clair. Chacune d'elle possède un "pointeur" vers son parent, qui n'est autre que la case de départ.

Ensuite, on choisit la case de coût F le plus bas de la "liste ouverte", et on répète plus ou moins le raisonnement précédent.

■ Coût d'un chemin :

La clé servant à déterminer quelle case utiliser parmi celles contenues dans la « liste ouverte » est l'équation suivante :

$$F = G + H \quad \text{avec :}$$

G est le coût de mouvement pour aller de la case A à une case donnée sur la grille, en suivant le chemin généré jusqu'ici.

H est le coût de mouvement pour aller d'une case donnée sur la grille jusqu'au point de destination (le point B). Il est souvent appelé « heuristique » ou coût estimé. Nous remarquons que nous ne connaissons pas vraiment la distance qu'il nous reste à parcourir, car toute sortes d'obstacles peuvent se trouver sur le parcours (mur, eau, etc..).

Notre chemin est généré récursivement en parcourant notre "liste ouverte" et en y choisissant la case ayant le coût F le plus faible.

Dans cet exemple, nous allons assigner un coût de 10 pour chaque déplacement horizontal ou vertical, et un coût de 14 pour un mouvement en diagonale. Nous utilisons ces données car la distance nécessaire pour se déplacer est la racine carrée de 2, ou approximativement 1.414 fois le coût d'un déplacement vertical ou horizontal. Nous utiliserons 10 et 14 pour des raisons évidentes de simplification. L'utilisation des nombres entiers est aussi bien plus rapide pour l'ordinateur également. L'algorithme A^* peut être très lent si on n'utilise pas ce genre d'astuces.

Etant donné que nous calculons le coût G le long d'un chemin donné, le moyen de trouver ce coût est de prendre le coût G de son parent, et de lui ajouté 10 ou 14 selon qu'il soit situé en diagonale ou pas par rapport à son parent. L'intérêt de cette méthode deviendra plus clair lorsque nous serons plus loin de la case du point de départ.

Le coût H peut être estimé d'un grand nombre de façons. La méthode que nous utiliserons ici est la méthode "Manhattan", avec laquelle on calcule le nombre de cases verticales et horizontales pour parvenir au point d'arrivée (en ignorant les mouvements en diagonale). Nous multiplions alors ce total par 10.

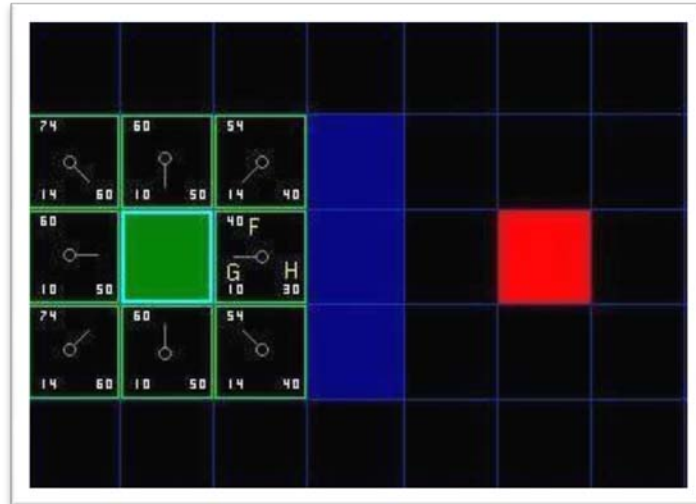


Figure 4-14 : Notion de calcul

Dans les cases qui contiennent les lettres, nous avons $G=10$. C'est parce qu'elle est située à une case à côté du point de départ, dans une direction horizontale. Les cases en dessous et au dessus de la case de départ, ainsi que celle située à sa gauche ont toutes le même coût de 10. Les cases en diagonales ont un coût de 14.

Les coûts H sont calculés en estimant la distance "Manhattan" jusqu'à la case d'arrivée, en se déplaçant horizontalement et verticalement, et en ignorant le mur qui est sur le chemin. Avec cette méthode, la case située juste à droite du départ est à trois déplacements horizontaux de l'arrivée, soit un coût H de 30. Le coût F de chaque case est calculé par une simple addition de G et H.

4.4.3 Continuer la recherche

Pour continuer la recherche, on choisit tout simplement la case ayant le coût F le plus faible parmi les cases de la « liste ouverte ». Puis, avec cette case nous faisons les processus suivants :

Nous la supprimons de la "liste ouverte" et la rajoutons à la « liste fermée ».

Nous vérifions toutes les cases adjacentes, en ignorant celles qui font partie de la « liste fermée », ainsi que celle qu'on ne peut pas traverser, que nous ajoutons à la « liste ouverte » si elles n'y sont pas déjà. Assignez la case en cours comme étant le « parent » des cases nouvellement ajoutées.

Si une des cases adjacentes est déjà dans la « liste ouverte », vérifiez si le chemin pour y arriver n'est pas meilleur. En d'autres termes, on vérifie si le coût G de cette case est inférieur, si nous utilisons la case en cours pour y parvenir. Si ce n'est pas le cas, on ne change rien.

D'un autre côté, si le coût G du nouveau chemin est inférieur, faites que la case en cours soit le nouveau parent de cette case adjacente. Pour finir, on recalcule les coûts F et G de cette case.

On étudie maintenant comment cela fonctionne. Sur les 9 cases initiales, nous avons 8 cases qui font partie de la "liste ouverte" après que la case de départ ait été envoyée vers la « liste fermée ». De ces 8 cases, celle ayant le coût F le plus faible est celle qui se situe immédiatement à droite de la case de départ, avec un coût F de 40. Donc, on choisit cette case comme étant la prochaine. Elle est entourée en bleu dans la figure 4-16 ci-dessous.

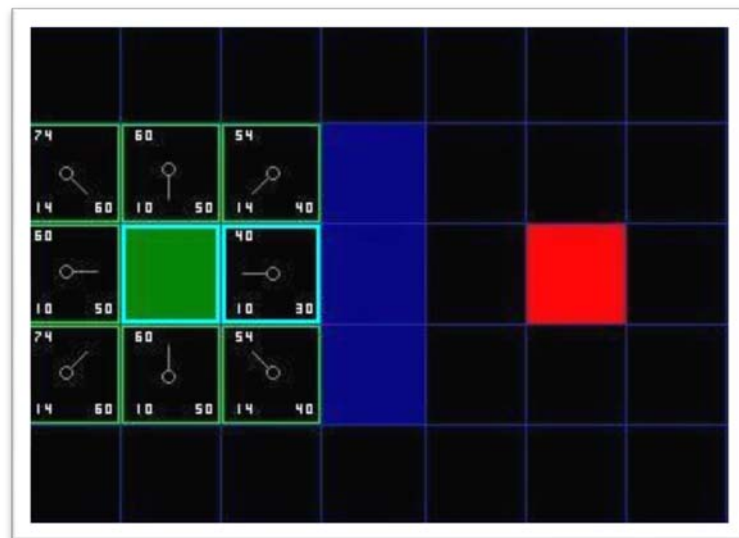


Figure 4-15 : Première étape de recherche

Tout d'abord, on supprime cette case de la « liste ouverte » pour l'envoyer dans la « liste fermée ». Puis, on vérifie ces cases adjacentes. Celle qui est immédiatement à droite est un mur, donc nous l'ignorons. Aussi celle immédiatement à gauche est la case de départ, Comme elle fait partie de la « liste fermée », on l'ignore également.

Les quatre autres cases sont déjà dans la "liste ouverte", donc on vérifie si les chemins qui y vont sont meilleurs en passant par notre case en cours, en utilisant le coût G comme point de comparaison.

Lorsque on répète ce processus pour les quatre cases adjacentes déjà dans la « liste ouverte », nous nous rendons compte qu'aucun chemin n'est amélioré en passant par la case en cours, donc on ne change rien. Maintenant que nous avons traité toutes les cases adjacentes, nous avons fini avec la case en cours, et prêts à passer à la suivante.

Donc nous parcourons à nouveau notre « liste ouverte », qui est maintenant composée de 7 cases, et nous prenons celle qui a le coût F le plus faible. Il est intéressant de constater que dans ce cas, il y a deux cases avec un coût de 54. Dans ce cas, pour des raisons d'optimisation, il peut être plus rapide de choisir le dernier qui été ajouté à la « liste ouverte ».

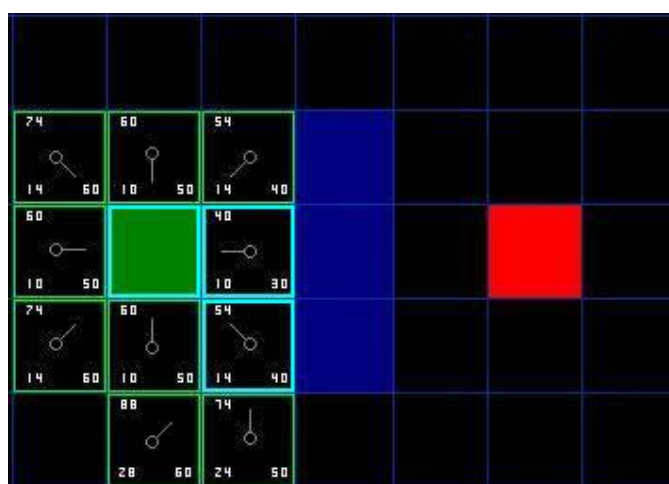


Figure 4-16 : Notion de recherche

Choisissons la case située juste en dessous et à droite de la case de départ, comme illustré dans le schéma ci-dessus.

Cette fois-ci, quand nous vérifions les cases adjacentes, nous voyons que celle située immédiatement à droite est un mur, donc nous l'ignorons. Nous faisons la même constatation pour celle située juste au dessus, nous ignorerons également celle qui est située en dessous du mur. Parce que on ne peut pas se rendre directement depuis la case en cours sans passer au dessus du coin du mur. Donc on est besoin d'aller vers le bas d'abord, pour ensuite se rendre sur cette case (Note : cette règle est optionnelle. Elle dépend en fait de l'endroit où sont situés les "noeuds").

Ceci nous laisse cinq autres cases. Les deux cases situées en dessous de la case en cours ne sont pas encore dans la « liste ouverte », donc nous les ajoutons avec la case en cours comme parent. En ce qui concerne les trois autres cases, deux sont déjà dans la « liste fermée », (la case de départ et les deux entourées en bleu sur le schéma) donc nous les ignorons, et la dernière case, immédiatement à gauche de la case en

cours, est vérifiée pour voir si le coût G est inférieur en passant par la case en cours. Ce n'est pas le cas, donc nous pouvons vérifier la case suivante dans notre « liste ouverte ».

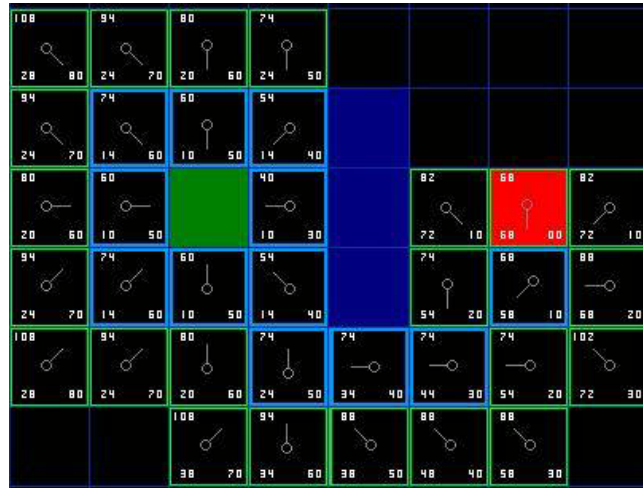


Figure 4-17 : Notion d'arrêt de recherche

Nous allons répéter ce processus jusqu'à ce que la case d'arrivée soit ajoutée à la « liste fermée », ce qui nous donne quelque chose comme le schéma ci-dessus.

On constate que le parent de la case située en dessous de deux cases de départ a changé par rapport à la figure 4-17. Avant, elle avait un coût G de 28, et pointait vers la case située en haut à droite. Maintenant elle a un score de 20 et pointe vers la case juste au dessus d'elle. Ceci est arrivé quelque part au long de notre recherche, alors que le coût G a été vérifié et modifié pour être plus faible en utilisant un autre chemin, donc le parent a été changé, et les coûts F et G recalculés.

Pour déterminer le chemin, on partant de la case d'arrivée, et en remontant le chemin en sens inverse d'une case à sa case parent, en suivant les flèches. Cela va nous ramener à notre chemin de départ.

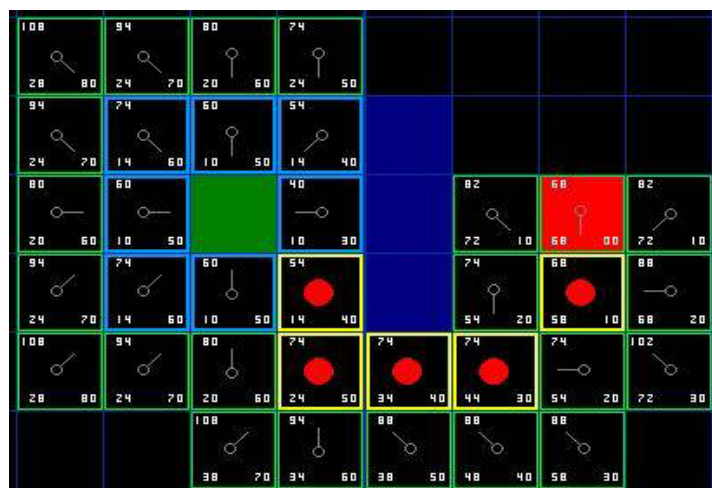


Figure 4-18 : Notion de tracer le chemin

4.4.4 Récapitulatif de la méthode A* (algorithme)

Maintenant que nous avons effectué la totalité de nos explications, faisons un petit récapitulatif étape par étape :

Ajouter le point de départ à la « liste ouverte »

Répéter les instructions suivantes :

- a. Cherche la case ayant le coût F le plus faible dans la « liste ouverte ». Elle devient la case en cours.
- b. Passer la case en cours de la « liste ouverte » à la « liste fermée ».
- c. Pour les 8 cases adjacentes à la case en cours.
 - Si on ne peut pas la traverser, on l'ignore.
 - Si elle n'est pas dans la « liste ouverte », on l'y ajoute. La case en cours devient le parent de cette case. On calcule les coûts F, G et H de cette case.
 - Si elle est déjà dans la « liste ouverte », on teste si le chemin passant par la case en cours est meilleur en comparant les coûts G. Un coût G inférieur signifie un meilleur chemin. Si c'est le cas, on change le parent de la case pour devenir la case en cours, en on recalcule les coûts F et G. Si vous conservez une « liste ouverte » triée par coût F, la liste doit être retriée à ce moment là.
- d. On s'arrête quand ?
 - La case de destination est ajoutée à la « liste fermée ».
 - Vous ne trouvez pas la case de destination et la « liste ouverte » est vide.

Enregistrez le chemin. En partant de la case de destination, remontez d'une case à son parent jusqu'à atteindre la case de départ.

4.4.5 Simulation de l'algorithme A* (AStar)

On va étudier l'application de l'algorithme A* (AStar) au cas d'une carte de grille décrite précédemment, pour cela on va donner les différents cas possibles suivantes en utilisant le langage MATLAB :

- Dans l'existence d'un obstacle infranchissable (mur fermé) ;
- Sans existence d'un obstacle (mur) ;
- Dans l'existence d'un obstacle quelconque mais pas fermé (mur).

■ Dans l'existence d'un obstacle infranchissable (mur fermé)

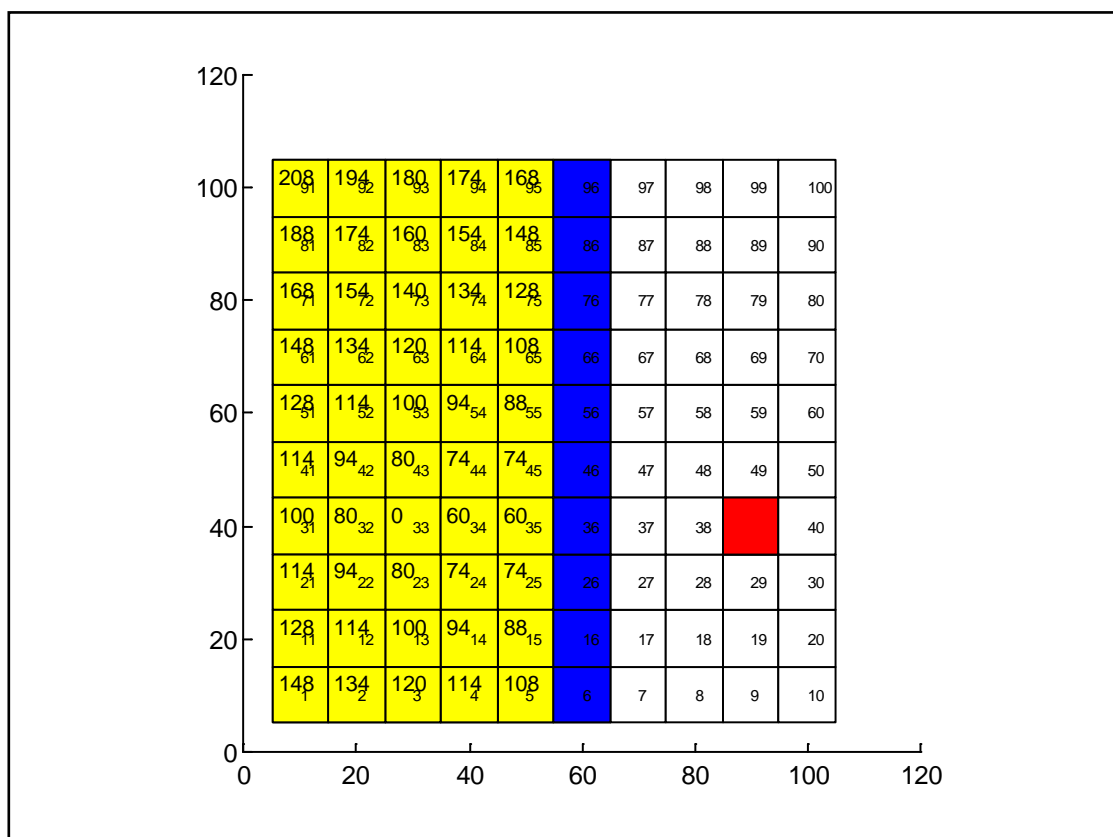


Figure 4-19 : Inexistence du chemin dans le cas d'un mur infranchissable

(Toutes les nœuds possible sont étudié par l'algorithme A*)

■ Sans existence d'un obstacle (mur) :

➤ 1er cas :

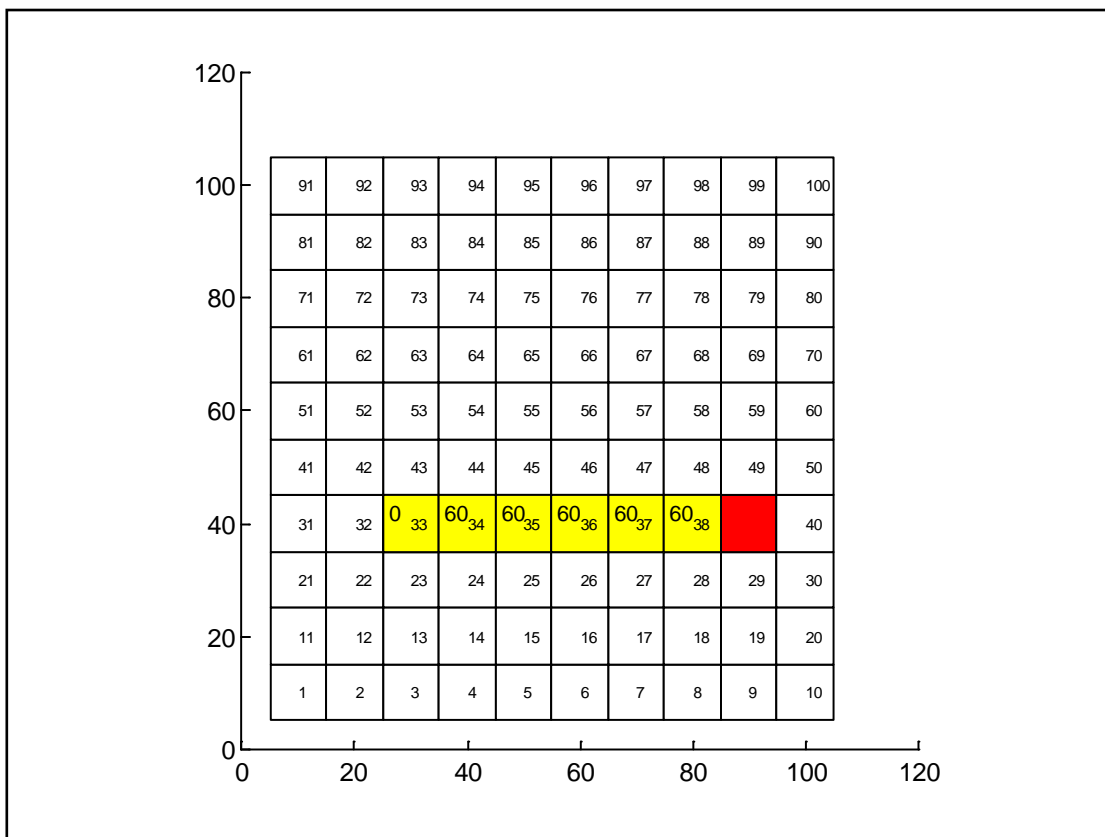


Figure 4-20 : Nœuds de recherche balayé en absence d'obstacle

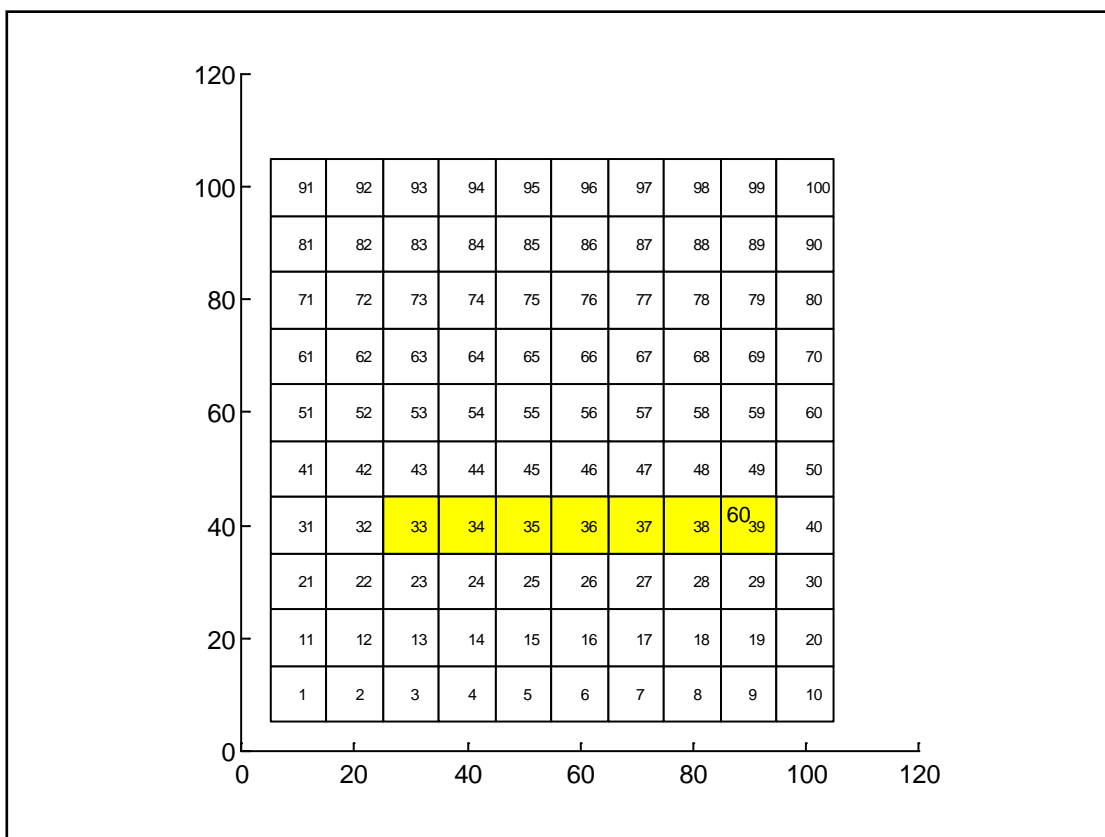


Figure 4-21 : Plus court Chemin trouver en absence d'obstacle

➤ 2ème cas :

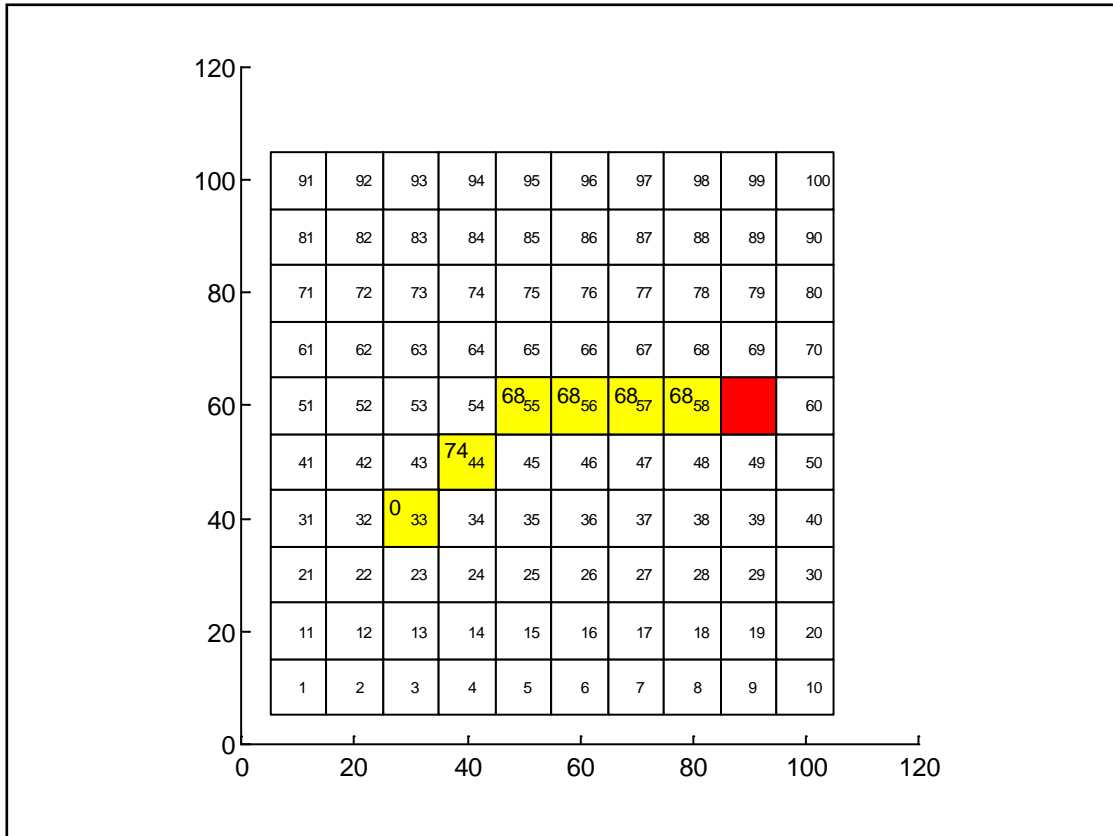


Figure 4-22 : Nœuds de recherche balayé en absence d'obstacle

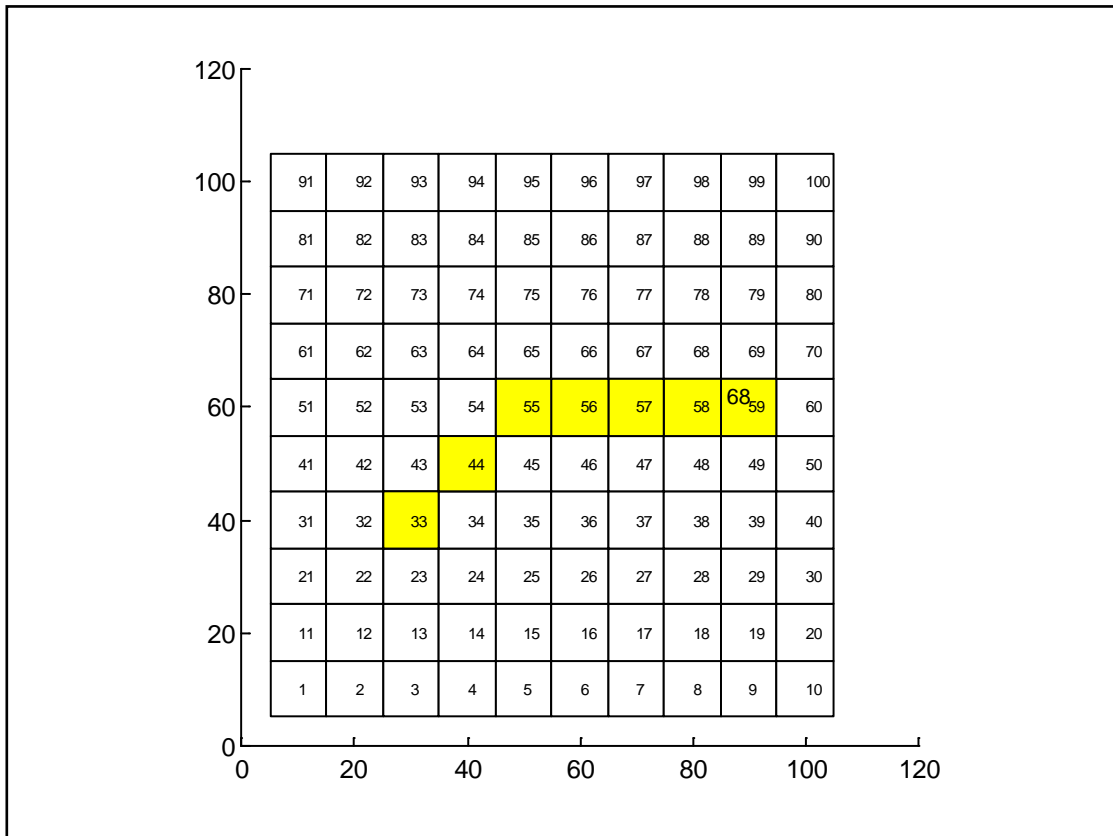


Figure 4-23 : Plus court Chemin trouver en absence d'obstacle

■ Dans l'existence d'un obstacle quelconque mais pas fermé

➤ 1er cas :

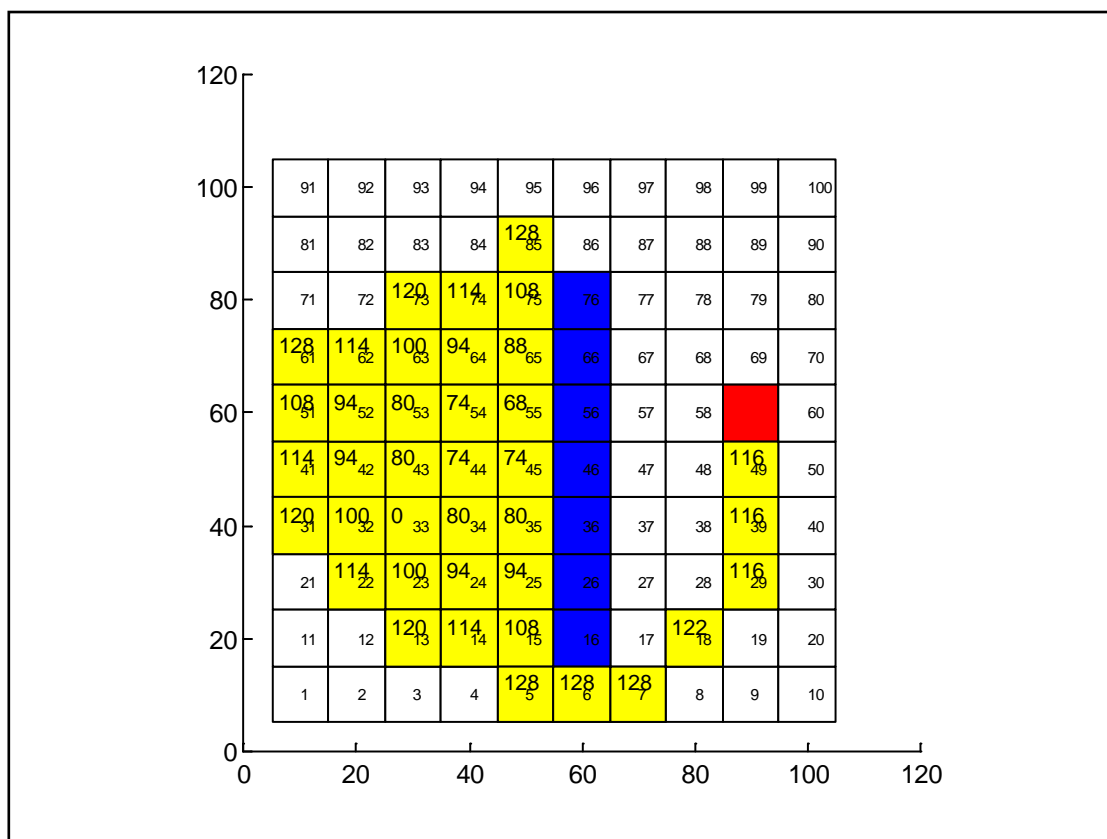


Figure 4-24 : Nœuds de recherche balayé dans la présence d'un obstacle

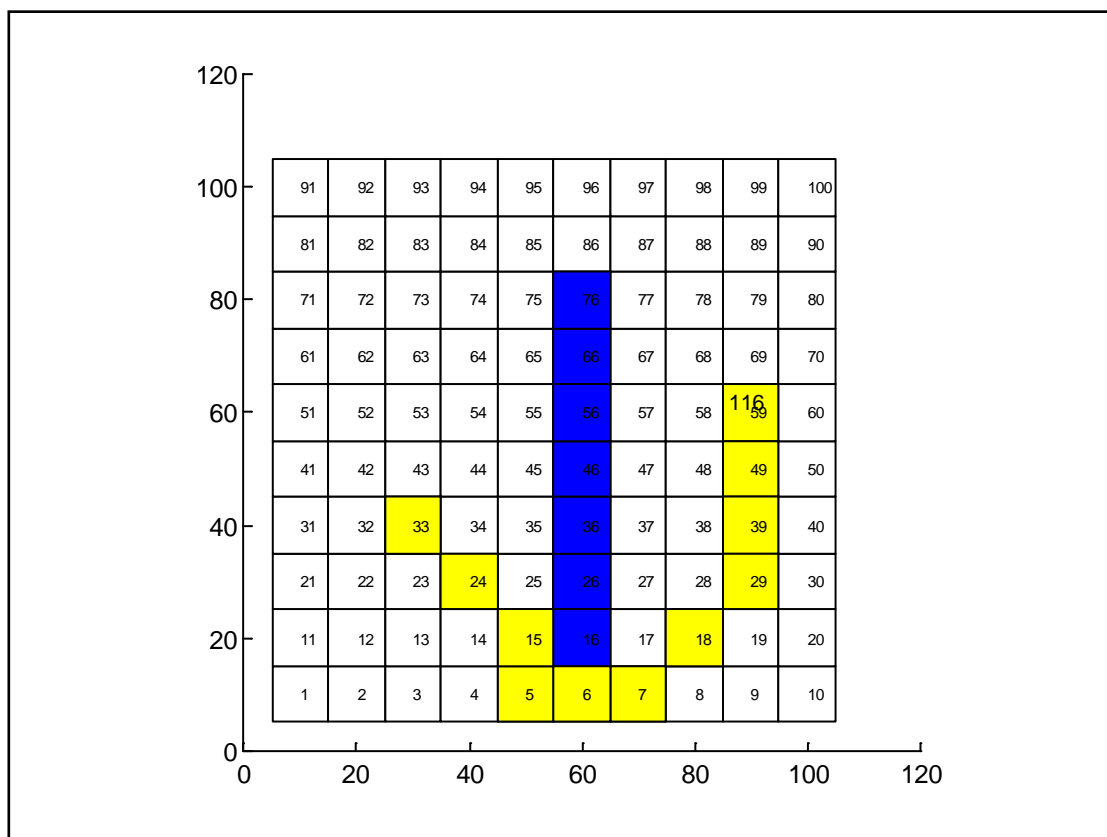


Figure 4-25 : Plus court chemin trouver dans la présence d'un obstacle

➤ 2ème cas :

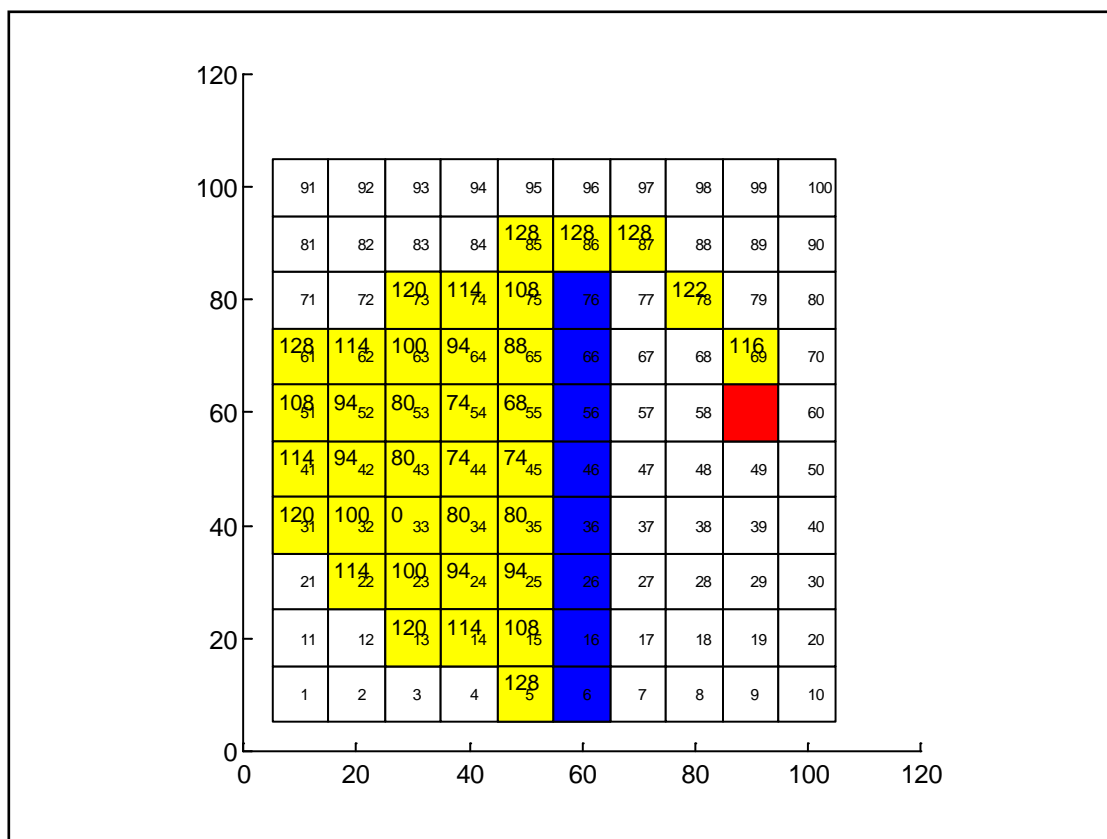


Figure 4-26 : Nœuds de recherche balayé dans la présence d'un obstacle

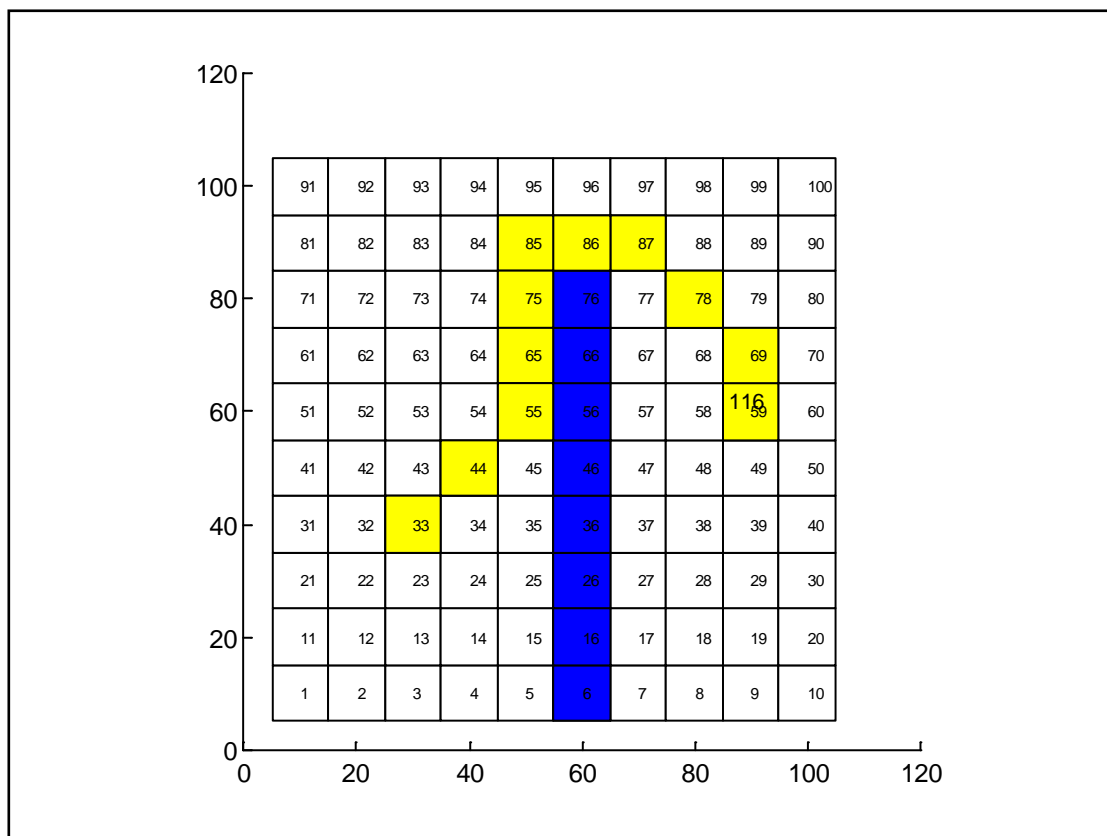


Figure 4-27 : Plus court chemin trouver dans la présence d'un obstacle

➤ 3ème cas :

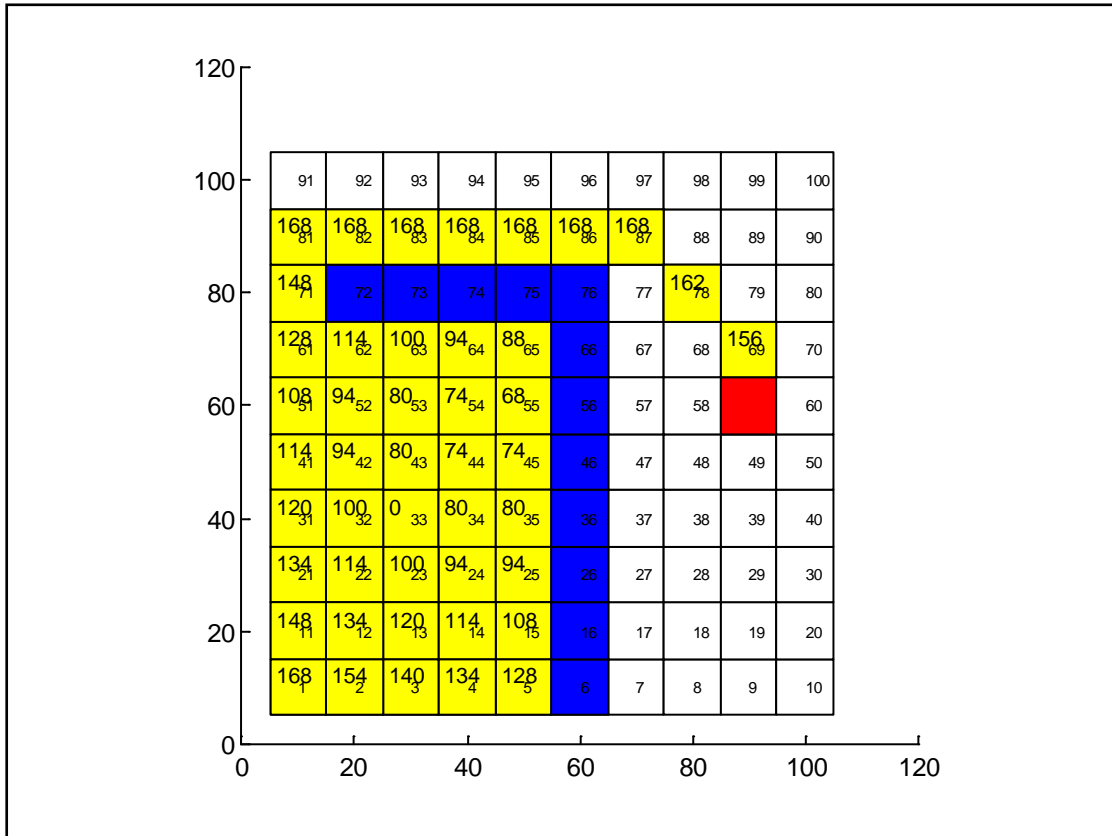


Figure 4-28 : Nœuds de recherche balayé dans la présence d'un obstacle

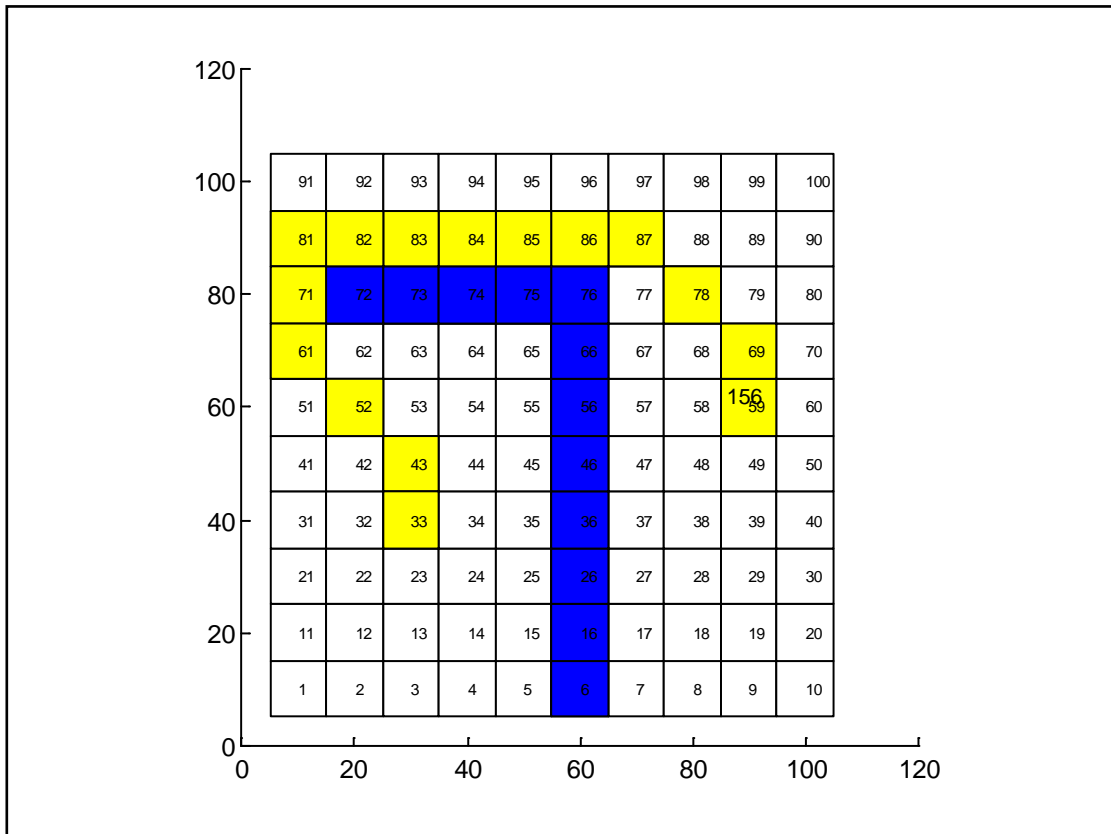


Figure 4-29 : Plus court chemin trouver dans la présence d'un obstacle

CHAPITRE 5 : SIMULATIONS ET COMMENTAIRES

5.1 Introduction

Pour résoudre le problème de l'évitement aérien, on cherche en général à construire des trajectoires sans conflit ayant une forme assez simple pour être facilement transmises et suivies par les pilotes. Ces trajectoires d'évitement sont souvent définies par des instants auxquels ont lieu des manœuvres prédéterminées. C'est par exemple le cas des deux principes cités dans le chapitre 3.

Les trajectoires d'évitement par point tournant avec retour direct sont des trajectoires répondant naturellement à cette exigence de simplicité. Elles aussi sont définies par deux instants de manœuvres, auxquels prennent place des changements de caps. Les segments de droites successifs dont sont constituées ces trajectoires sont appelés les étapes de la trajectoire. Pour un avion se trouvant à une étape donnée de la trajectoire, effectuer un virage (éventuellement parmi plusieurs virages possibles) et passer ainsi à l'étape suivante peut être vu comme effectuer un choix entre plusieurs actions possibles (un des virages possibles, ou pas de virage). Cette manière de considérer de telles trajectoires justifie le fait de chercher à les traduire en termes de choix successifs, ce qui permet de représenter l'ensemble des trajectoires d'évitement possibles pour un avion comme un arbre. On peut alors, après avoir défini le *coût* d'un tel chemin, utiliser pour générer des trajectoires d'évitement optimales un algorithme tel que l'algorithme A*, Algorithme de recherche de chemin de coût minimal dans un arbre à coût additif. Cet algorithme souvent utilisé pour la génération de trajectoires en robotique nous a semble pouvoir être utilisé pour générer des trajectoires sans conflit.

L'utilisation de cet algorithme obligera à restreindre l'ensemble des trajectoires envisageables, en discrétisant le temps, et en imposant que les instant de manœuvres prennent place sur des pas de temps. En effet, nous souhaitons traduire ces trajectoires en termes de choix successifs. En l'absence de discrétisation du temps, les choix peuvent s'opérer à tout instant du temps continu, l'ensemble des trajectoires possibles ne peut pas se représenter comme un arbre.

5.2 Les manœuvres utilisées

Nous utiliserons des trajectoires d'évitement du type point tournant avec retour direct vers la destination, c'est-à-dire que les avions suivront des trajectoires normalisées, constituées de 4 étapes définies de la manière suivante :

- Première étape (E_0) : jusqu'à un temps $t_0 \geq 0$ l'avion suit sa trajectoire d'origine.
- Deuxième étape (E_1) : 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 .
- Troisième étape (E_2) : au temps t_1 , l'avion prend le cap qui lui permet de rejoindre directement sa position finale.
- Quatrième étape (E_{fin}) : l'avion a atteint sa position finale, et poursuit sa route sur sa trajectoire initialement prévue.

La position finale d'un avion, que nous appellerons aussi sa destination, est la position qu'occuperait l'avion, en l'absence de déviation, à la fin de la période de temps sur laquelle est assurée l'évitement, cela correspond à la *durée de l'évitement* t_f .

La figure 5-1 montre une trajectoire d'évitement par point tournant avec retour direct vers la destination, répondant à la définition d'une trajectoire d'évitement normalisée donnée ci-dessus :

L'avion est en sa position d'origine, notée O , au temps $t = 0$. Sa position finale est notée D . L'avion conserve son cap jusqu'au temps $t = t_0$. Sur l'exemple de la figure 5-1, au temps $t = t_0$, il opère un virage à gauche. Il prend au temps t_1 un cap de retour direct vers sa position finale. La trajectoire originellement prévue pour l'avion est indiquée en pointillés.

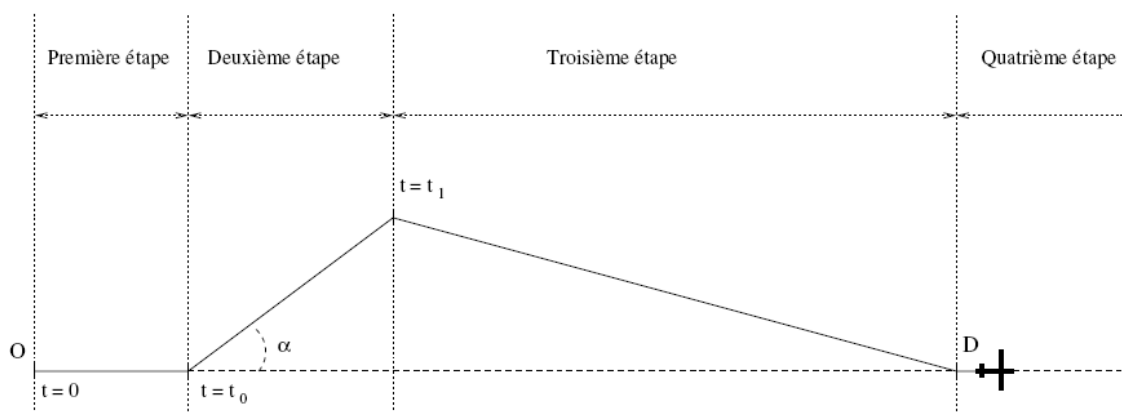


Figure 5-1 : Trajectoire d'évitement avec un point tournant

Définition 5-1 : – Trajectoire d'évitement normalisé – on appellera « trajectoire d'évitement normalisée », ou plus simplement « trajectoire normalisée » toute trajectoire correspondant à la description que nous avons donnée ci-dessus.

Dans la suite, les étapes des trajectoires d'évitement normalisées seront aussi appelées étapes de l'évitement.

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

A chaque étape de cette trajectoire correspondent certains choix possibles : opérer un virage (et passer à l'étape suivante) ou non, et éventuellement dans quelle direction opérer le virage. Le fait de voir une trajectoire comme une suite de choix mène à envisager de représenter l'ensemble des trajectoires possibles par un arbre, et d'utiliser un algorithme de recherche de chemin optimal dans un arbre pour générer une trajectoire d'évitement optimale.

Nous apportons à la modélisation par point tournant présentée ici les restrictions permettant de représenter une telle trajectoire comme un chemin d'un arbre.

5.3 Détection des conflits

Chaque avion doit pouvoir pour optimiser sa route, détecter les avions dans son entourage et connaître leurs trajectoires. Ainsi il pourra détecter les conflits éventuels avec ses voisins. Pour cela, nous devons définir un rayon de détection qui garantisse que deux avions qui ne se détectent pas ne peuvent pas se trouver en conflit durant la durée de l'horizon temporel F_a .

Si l'on suppose que les avions les plus rapides ont des vitesses voisines de 500 Kt et si l'on prend le cas le plus défavorable du conflit face à face, deux avions peuvent, en 5 minutes, se rapprocher de $\frac{1000 \times 5}{60}$ Nm soit environ 84 Nm. Avec une séparation standard de 5Nm, un rayon de détection de 90 Nm garantit que les avions ne peuvent pas être en conflit dans la fenêtre F_a .

Plus généralement, si V_{\max} est la vitesse maximale possible pour les avions, si N_h est la norme horizontale et R_d le rayon de détection, on doit avoir :

$$R_d \geq 2V_{\max} \cdot F_a + N_h \quad (5.1)$$

5.4 L'Algorithme A*

La méthode présentée ici utilise l'algorithme A*, qui est un algorithme de recherche de chemin de coût minimal dans un arbre à coût additif. Cet algorithme a été utilisé pour la première fois par Hart, Nicolson et Raphael en 1968. L'algorithme A* est un algorithme permettant de chercher 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. [6,19]

Par meilleur nœud on entend le nœud de coût total estimé minimal (cette notion de coût total estimé sera précisée ci-dessous).

Définition 5-2 : – Nœud développé – on dit qu'un nœud est développé quand tous ses successeurs ont été générés.

L'utilisation de l'algorithme A* nécessite la connaissance des coordonnées 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. Si v et u sont deux nœuds, et si v est généré à partir de u en utilisant la règle p_i , on note : $v \xleftarrow{p_i} u$.

$k(u, v)$: Si u et v sont des nœuds de l'arbre, k est une fonction donnant le coût du passage par l'arc (u, v) . Le coût total d'obtention d'un état donné est égal à la somme des coûts des arcs reliant l'état initial à cet état, le coût de chacun de ces arcs étant calculé au moyen de la fonction k . Dans le cas qui nous intéresse ici, $k(u, v)$ devra traduire la fonction que nous cherchons à minimiser, c'est-à-dire l'allongement des trajectoires.

$h(u)$: 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 $k(u, v_1) + k(v_1, v_2) + \dots + k(v_n, u_t)$ sur tous les chemins $(u, v_1, \dots, v_n, u_t)$, où u_t est un état terminal.

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 (la liste ouverte).

Premier (G) : une fonction qui renvoie le premier élément de la liste G .

D : La liste des états générés et développés (la liste fermée).

$g(v)$: Une fonction de l'état (v) donnant le coût de 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 G 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 :

$G \leftarrow (u_0); D \leftarrow \emptyset; g(u_0) \leftarrow 0; f(u_0) \leftarrow 0$

Tantque $G \neq \emptyset$ *faire*

(a) $u \leftarrow \text{premier}(G); G \leftarrow G \setminus (u); D \leftarrow D \cup (u)$

(b) *Si* $u \in T$ *alors* *donner_solution*; *fin_programme*; *fin_si*

(c) *pour* i *allant de* 1 *jusqu'à* n *faire*

i. $v \leftarrow P_i \text{---} u$

ii. *Si* $v \notin D \cup \text{Gou}[g(v) > g(u) + k(u,v)]$ *alors*

A. $g(v) \leftarrow g(u) + k(u,v)$

B. $f(v) = g(v) + h(v)$

C. $\text{père}(v) \leftarrow u$

D. *insérer_selon_f* (G,v)

iii. *fin_si*

(d) *fin_faire*

fin_faire

Les performances de l'algorithme A^* dépendent principalement de l'heuristique h . Rappelons que, pour un état u , $h(u)$ tente d'approcher $h^*(u)$, qui est le minimum de la somme :

$$k(u, v_1) + k(v_1, v_2) + \dots + k(v_n, u_t) \quad (5.2)$$

Sur tous les chemins $(u, v_1, \dots, v_n, u_t)$, u_t étant un état terminal de l'arbre.

Dans l'application que nous allons présenter de l'algorithme A^* à la résolution de conflit, cette mesure de la complexité ne sera pas facilement utilisable, car le nombre total de nœuds de l'arbre de recherche que nous utilisons sera difficile à estimer. Il existe une autre mesure de la complexité de l'algorithme A^* , donnant un ordre de grandeur du nombre de nœuds de l'arbre qui seront considérés par l'algorithme avant d'obtenir la solution, en fonction du nombre d'arcs entre l'état initial l'état de l'arbre et un état final.

Application à la résolution de conflit

5.4.1 Arbre de recherche

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* (terminer le temps de simulation) : tous les avions ont atteint leurs positions finales.

Chaque branche de l'arbre représente une trajectoire de l'avion central. 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 nous nous attacherons à minimiser l'allongement des trajectoires, et c'est donc cet allongement que devra traduire la fonction donnant le coût d'un état correspondant à un nœud de l'arbre).[2,7]

5.4.2 L'évitement

L'avion central doit établir sa trajectoire parmi d'autres avions qui ont déjà optimisé leur trajectoire ou qui, n'ayant pas de conflits, n'ont pas de résolution à effectuer. Tous ces avions sont dans le rayon de détection de l'avion central. Celui-ci n'aura qu'à conserver les normes de séparations avec les positions des avions voisins. L'algorithme

doit trouver des trajectoires libres de conflit tout en minimisant la longueur de la déviation. La plus courte de ces trajectoires est bien évidemment celle qui permet de rejoindre directement la destination, mais elle n'est pas nécessairement libre de conflit.

5.4.3 Développement d'un nœud de l'arbre

Cette partie présente les différentes étapes d'une résolution de type point tournant. Ces différents états ne peuvent pas s'enchaîner de manière quelconque. Considérons les possibilités de déroulement d'une trajectoire d'évitement :

- Lorsqu'un avion est encore à la première étape de l'évitement E_0 , il a à tous les pas de temps le choix entre 7 actions possibles :
 - Poursuivre sa route en conservant son cap d'origine, en restant dans la première étape de l'évitement.
 - Modifier sa trajectoire en opérant un virage normalisé à gauche avec un cap de 10, 20 ou 30 degrés et passer ainsi à la deuxième étape de l'évitement (E_1).
 - Opérer un virage normalisé à droite avec un cap de 10, 20 ou 30 degrés et passer à la deuxième étape de l'évitement (E_1).
- Si l'avion opère un virage, à droite ou à gauche, et passe ainsi à la seconde étape de l'évitement, il n'a plus le choix, alors, qu'entre deux actions possibles :
 - Poursuivre sa route en conservant son cap modifié, en restant dans la deuxième étape de l'évitement (E_1).
 - Opérer un virage pour suivre un cap direct vers sa position finale, il passe alors à la troisième étape de l'évitement (E_2).
- 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.

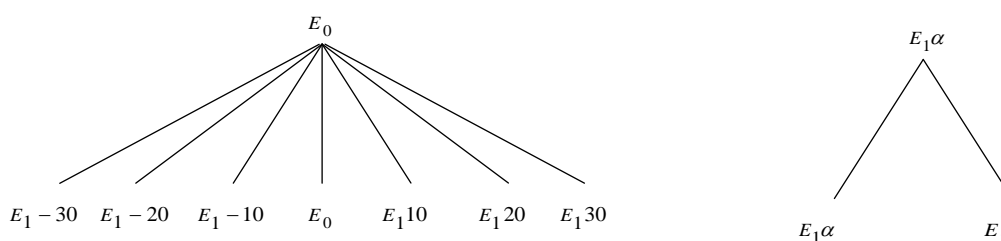


Figure 5-2 : Développement des nœuds d'un arbre

5.4.4 Exemple de développement d'un arbre pour un avion

L'exemple de la figure 5-3, présente le développement d'un arbre.

Notons que les valeurs de l'heuristique mentionné dans la figure 5-3, sont des valeurs exemples pour montré le développement et la notion de recherche du nœuds optimal choisi seulement.

Le premier nœud a 7 fils qui représentent les 7 états générés par les règles de production. Parmi 7 fils, la branche centrale est celle qui a le coût estimé jusqu'à destination (coût réalisé jusqu'au nœud courant, augmenté de l'heuristique jusqu'à la destination) la plus faible (100) (voir fig 5-3). C'est donc cette branche qui est explorée, on passe en 1.

Quand on développe le nœud en 1, on trouve 7 états distincts accessibles. L'algorithme compare le coût estimé minimum (100) parmi 7 états, à tous les coûts précédemment calculés. Le coût estimé minimum nous amène à choisir le nœud central, on passe en 2.

Quand en développe le nœud en 2, on trouve à nouveau 7 états, parmi lesquels l'état central a le coût estimé minimum(1000). L'algorithme compare ce coût estimé à tous les coûts précédemment calculés et choisi d'explorer la branche de coût minimum non explorée jusque-là. Il revient au nœud 1 et explore la branche +30° de coût estimé 105.

Quand on développe le nœud 3, on trouve 2 états. L'algorithme choisit de poursuivre la manœuvre d'évitement (coût estimé 107).

Le nœud 4 conduit l'algorithme à poursuivre l'exploration de l'arbre en ramenant l'avion sur sa destination (coût estimé 107).

Au nœud 5, on trouve un seul état possible dont le coût estimé (107) est inférieure à tous les coûts estimés des branches non explorées. L'algorithme poursuit donc sa recherche en passant en nœud 6.

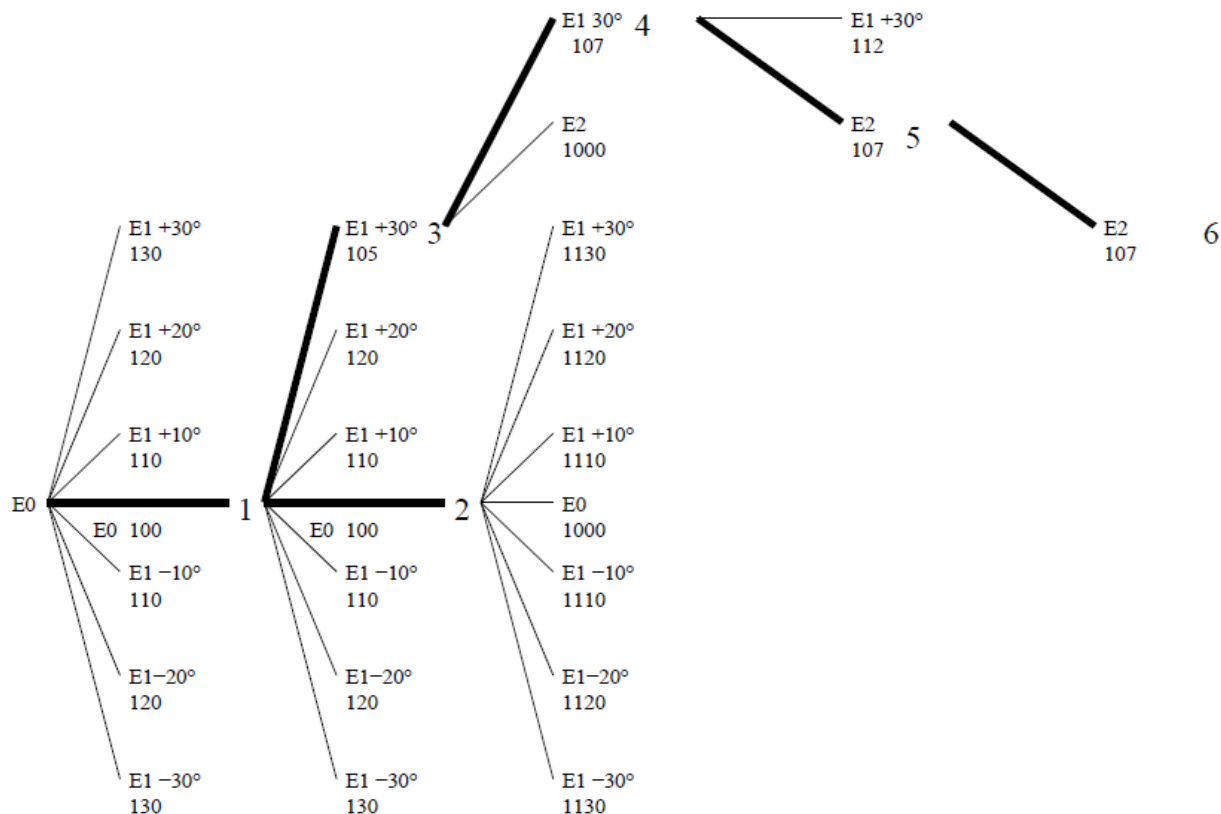


Figure 5-3 : Développement d'un arbre pour un avion

En général, beaucoup plus de chemins sont explorés, générant de nombreux retours « en arrière » dans l'arbre. La profondeur de l'arbre dépend de la durée de la prévision. Pour une prévision de 5 minutes et un pas de discrétisation de 15 secondes, la profondeur est de 16.

5.4.5 Trajectoires d'évitement discrétisables

Nous souhaitons traduire la trajectoire d'évitement normalisée en termes de choix successifs. Il est nécessaire pour cela de discrétiser le temps.

Le temps est donc divisé en *pas de temps* d'une longueur constante, notée l_t . A chaque pas de temps, chaque avion pourra modifier ou non sa trajectoire, selon l'étape de l'évitement dans laquelle il se trouvera. On a donc ainsi une suite discrète d'instantanés auxquels des choix sont opérés. Les temps t_0 et t_1 prennent des valeurs discrètes et les trajectoires d'évitement ainsi définies peuvent être représentées par les chemins d'un arbre.

Nous pouvons maintenant, à partir de la définition 5-1, donner des définitions des trajectoires d'évitement normalisées discrétisables et des jeux de trajectoires admissibles que nous utiliserons dans la suite de notre travail.

Définition 5-7 : – trajectoire d'évitement normalisée discrétisables – dans la suite de ce chapitre, l'expression trajectoire d'évitement normalisée discrétisables désignera des trajectoires telles qu'elles sont décrites dans la définition 5-1, c'est-à-dire des trajectoires comportant les différentes étapes de l'évitement, c'est-à-dire pour lesquelles les temps t_0 et t_1 correspondent à des pas du temps discrétisé.

Dans la suite de ce chapitre, on pourra, pour plus de simplicité employer l'expression trajectoire d'évitement normalisée au lieu de trajectoire d'évitement normalisée discrétisable. Toutes les trajectoires normalisées considérées seront discrétisables, il n'y aura pas risque d'ambiguïté.

Définition 5-8 : – Jeu de trajectoires admissible – Dans le cadre de la résolution d'un conflit donné, impliquant n avions, on appellera « jeu de trajectoire admissible », l'ensemble de n trajectoires d'évitement normalisées discrétisables (définition 5-7) et telles que si les avions suivent ces trajectoires, alors :

Tous les avions atteignent leurs positions finales ;

Entre le temps 0 et le temps auquel tous les avions ont atteint leurs positions finales, il n'y a aucun conflit entre des avions initialement impliqués dans le conflit que l'on s'attachait à résoudre.

Définition 5-9 : – Longueur d'une trajectoire d'évitement normalisé – on appelle « longueur », d'une trajectoire d'évitement normalisée d'un avion la longueur de la partie de cette trajectoire située entre la position initiale et la position finale de cet avion. L'unité de longueur est le mille nautique (Nm).

Définition 5-10 : – longueur d'un jeu de trajectoire admissible – on appelle « longueur » d'un jeu de trajectoires admissible la somme des longueurs, au sens de la définition précédente, des trajectoires d'évitement normalisées constituant ce jeu de trajectoires admissible.

5.5 Exemple d'arbre de recherche

Pour les raisons évoquées dans la section précédente, le temps est discrétisé. A chaque pas de temps on peut considérer l'état de l'évitement, défini de la manière suivante :

Définition 5-11 : – Etat de l'évitement – Pour un évitement impliquant n avions, notés a_i pour $i = 1, \dots, n$, un état de l'évitement est donné par la position et le cap de chaque avion a_i , et l'étape de la trajectoire d'évitement dans laquelle se trouve chaque avion a_i .

Chaque nœud de l'arbre sur lequel opérera l'algorithme A^* correspondra à un état de l'évitement et à un pas de temps donnés.

Ainsi la racine de l'arbre correspond au temps 0 et à l'état dans lequel tous les avions sont en leurs positions initiales, ont leurs caps initiaux et sont tous à l'étape E_0 .

A partir de la définition des trajectoires d'évitement normalisées (définition 5-1), on peut définir les règles de génération permettant de créer les fils d'un nœud courant de l'arbre.

On considère un nœud de l'arbre, noté u , on note j_u le pas de temps correspondant, et pour chaque avion a_i on note $(x_i(u), y_i(u))$ sa position, $c_i(u)$ son cap, et $E_{a_i}(u)$ l'étape de la trajectoire normalisée dans laquelle il se trouve.

Le pas de temps correspondant à tous le fils du nœud u sera $j_u + 1$.

Nous avons vu dans la section précédente quels sont à chaque pas de temps les choix possibles pour un avion. Selon sa position, son cap et l'étape de l'évitement dans laquelle il se trouvait dans l'état de l'évitement correspondant au nœud u , chaque avion a_i pourra donc occuper différentes positions, suivre différents caps, être dans différentes étapes de l'évitement. Les différents états correspondant aux différents fils du nœud u seront donc obtenus par la combinaison des différents choix possibles pour les avions à partir de l'état correspondant au nœud u .

L'exemple ci-dessous illustre ce qui précède.

On génère les trajectoires de 3 avions, et on a au nœud u l'état suivant :

- Avion en $(x_1(u), y_1(u))$ avec le cap $c_1(u)$, en première étape de l'évitement (étape E_0)
- Avion en $(x_2(u), y_2(u))$ avec le cap $c_2(u)$, en deuxième étape de l'évitement (étape E_1)
- Avion en $(x_3(u), y_3(u))$ avec le cap $c_3(u)$, en troisième étape de l'évitement (étape E_2)

Au pas de temps suivant, on peut choisir pour l'avion a_1 parmi 3 possibilités, pour l'avion a_2 entre 2 seulement, et l'avion a_3 , en étape E_2 , n'a aucun choix. L'avion a_3 , en effet, soit a atteint sa position finale et passe à l'étape E_{fin} , soit ne l'a pas atteinte et dans ce dernier cas poursuit sa route vers elle sans changement de cap. ces deux possibilités ne dépendent d'aucun choix, mais découlent simplement de la position de l'avion a_3 . Supposons que cette position est elle que l'avion a_3 poursuit sa route.

Le nœud u engendre 6 fils (voir figure 5-), correspondant aux 6 cas suivants :

1. Les trois avions poursuivent leurs routes en conservant leurs caps et en restant aux mêmes étapes de l'évitement.
2. Les avions a_1 et a_3 poursuivent leurs routes en conservant leurs caps et en restant aux mêmes étapes de l'évitement, tandis que l'avion a_2 prend un cap direct vers sa position finale, passant ainsi à l'étape E_2 .
3. L'avion a_1 opère un virage à gauche, passant ainsi à l'étape E_1 , les deux autres avions conservant leurs caps courants et restant aux mêmes étapes de l'évitement.
4. L'avion a_1 opère un virage à gauche et passe à l'étape E_1 tandis que l'avion a_2 prend un cap direct vers sa position finale et passe à l'étape E_2 et que l'avion a_3 conserve son cap courant et reste à l'étape E_2 .
5. L'avion a_1 opère un virage à droite et passe à l'étape E_1 , les deux autres avions conservant leurs caps courants et restants aux mêmes étapes.
6. L'avion a_1 opère un virage à droite et passe à l'étape E_1 tandis que l'avions a_2 prend un cap direct vers a position finale et passe à l'étape E_2 et que l'avion a_3 conserve son cap courant et reste à l'étape E_2 .

Au pas de temps suivant, une fois fait le choix d'opérer ou non un virage et, dans le cas du passage de l'étape E_0 à l'étape E_1 , une fois le sens du virage choisi, on est dans un des 6 cas décrits ci-dessus. Les positions et les caps des avions sont alors totalement déterminés. On a donc pour les fils du nœud u 6 états possibles parfaitement déterminés.

Ainsi se déduisent, à partir de l'état de l'évitement correspondant aux différents fils du nœud u , les différents états de l'évitement correspondant aux différents fils du nœud u . En appliquant à un nœud de l'arbre de recherche les règles définies ci-dessus, on a donc pour un nœud donné de l'arbre de recherche les règles définies ci-dessus, on a donc pour un nœud donné de l'arbre un nombre de fils dépendant de l'état correspondant à ce nœud, qu'on peut calculer de la manière suivante :

Pour un évitement impliquant n avions, si u est un nœud de l'arbre de recherche correspondant à un état dans lequel il y a n_0 avions à l'étape E_0 de l'évitement, n_1 avions à l'étape E_1 , n_2 avions à l'étape E_2 et n_{fin} avions à l'étape E_{fin} , le nœud u génère un nombre de fils égal à $3^{n_0} \times 2^{n_1}$.

Ceci découle directement du fait qu'il y-a 3 choix possibles pour chaque avion en première étape, et deux pour chaque avion en deuxième étape. On voit que les nombres d'avions en troisième étape. On voit que le nombre d'avions en troisième étape (n_3) et en

dernière étapes (n_{fin}) dans l'état de l'évitement correspondant au nœud u n'influent pas sur le nombre de fils du nœud u , les seuls choix possibles correspondant aux avions en première ou en deuxième étape de l'évitement.[51]

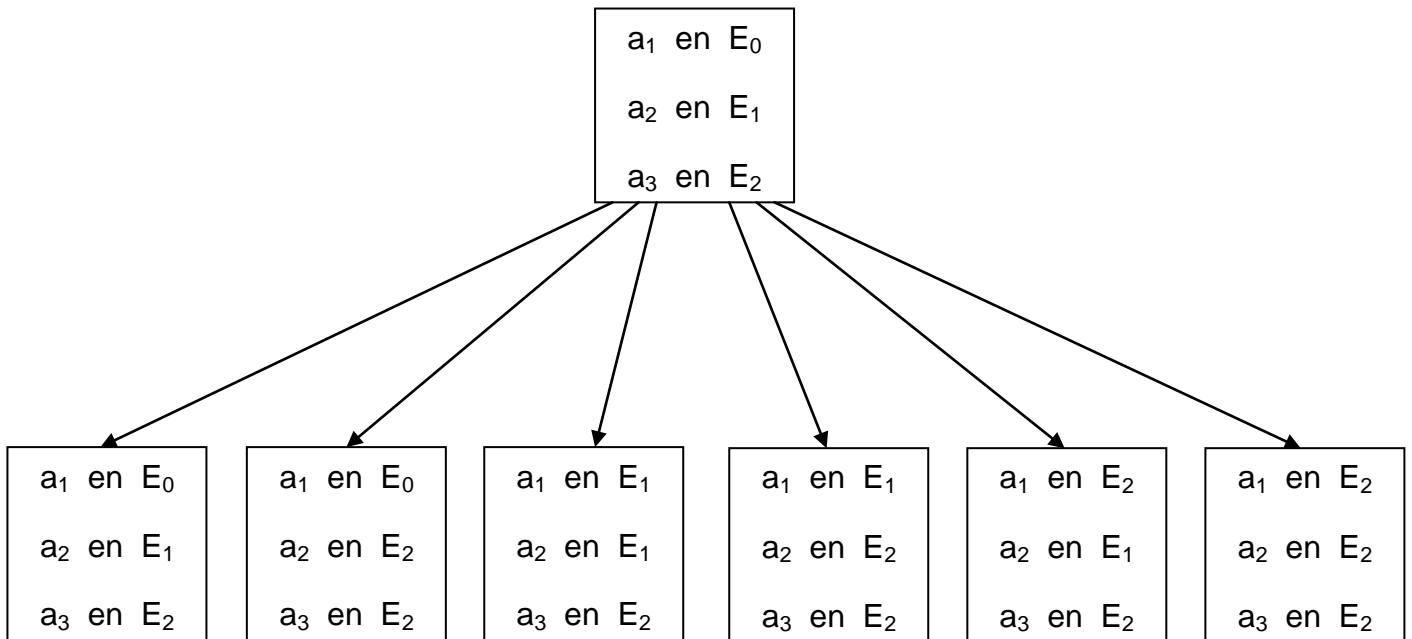


Figure 5-4 : Développement d'un arbre de recherche avec une approche globale

Notons que chaque avions à l'étape E_1 a le choix entre deux directions possibles, à droite ou à gauche

5.6 Simulation d'application de l'algorithme A* au cas de conflit aérien

5.6.1 Cas de deux avions

■ Premier cas :

➤ Tableau de données :

Tableau 5-1 : Données initiales des deux avions du premier cas

Avion	x_0 (Nm)	y_0 (Nm)	v (kt)	Cap initial(°)	Pas de temps(s)	Nbre de pas	Temps total(s)	x_{fin} (Nm)	y_{fin} (Nm)
A	15	0	200	90	15.9	33	524.7	44.15	0
B	0	0	450	90				65.59	0

➤ Tableaux des Résultats :

Tableau 5-2 : Paramètres d'évitements de l'avion central B

	Instant de débuts(s)	Durée(s)	Cap(°)
Tronçon n°1 (Etape E_0)	0	31.8	90
Tronçon n°2 (Etape E_1)	31.8	143.1	110
Tronçon n°3 (Etape E_2)	174.9	349.8	82.22

Tableau 5-3 : Temps total et temps de retard de l'avion central

Durée de trajectoire rectiligne(s)	Durée avec résolution de conflit(s)	t_{retard} (s)
524.7	536.66	11.96

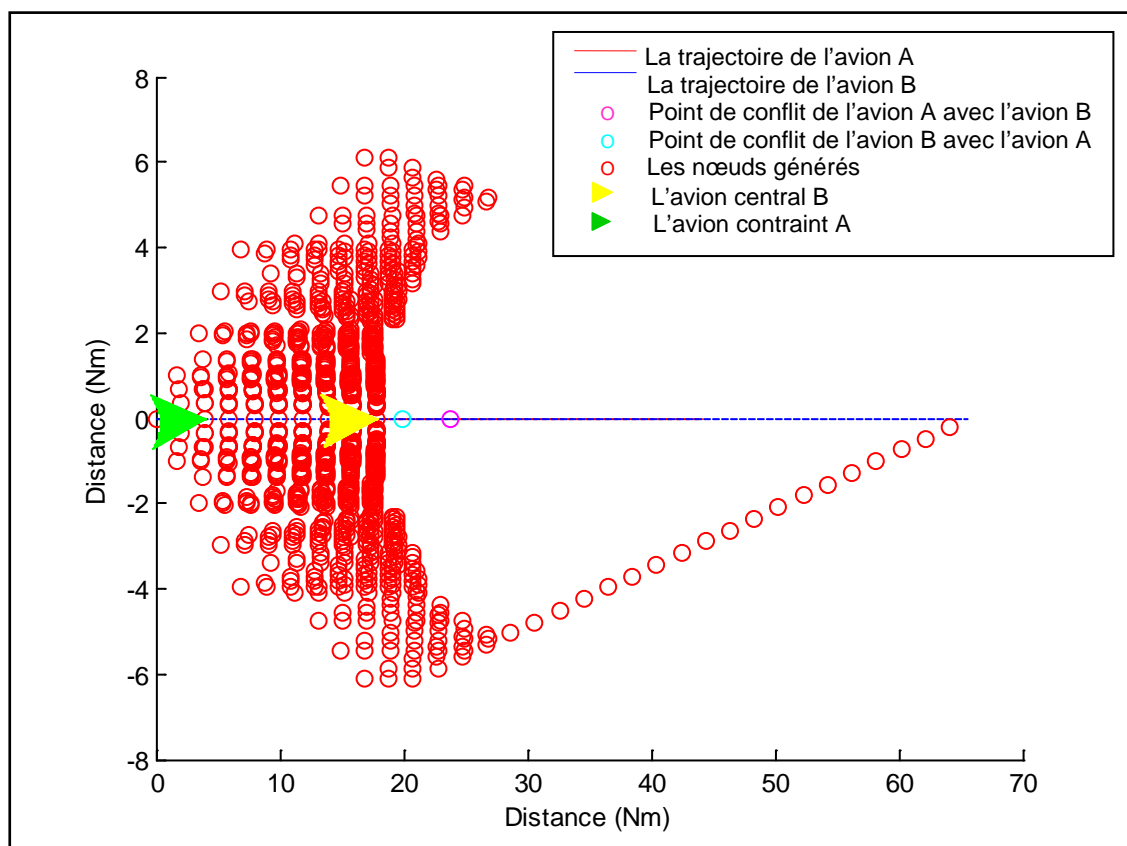


Figure 5-5 : Les nœuds de recherche de l'algorithme A*

Avion A ($x_0=15$ (Nm), $y_0=0$ (Nm), $v=200$ (kt), $cap=90$ (°))

Avion B ($x_0=0$ (Nm), $y_0=0$ (Nm), $v=450$ (kt), $cap=90$ (°))

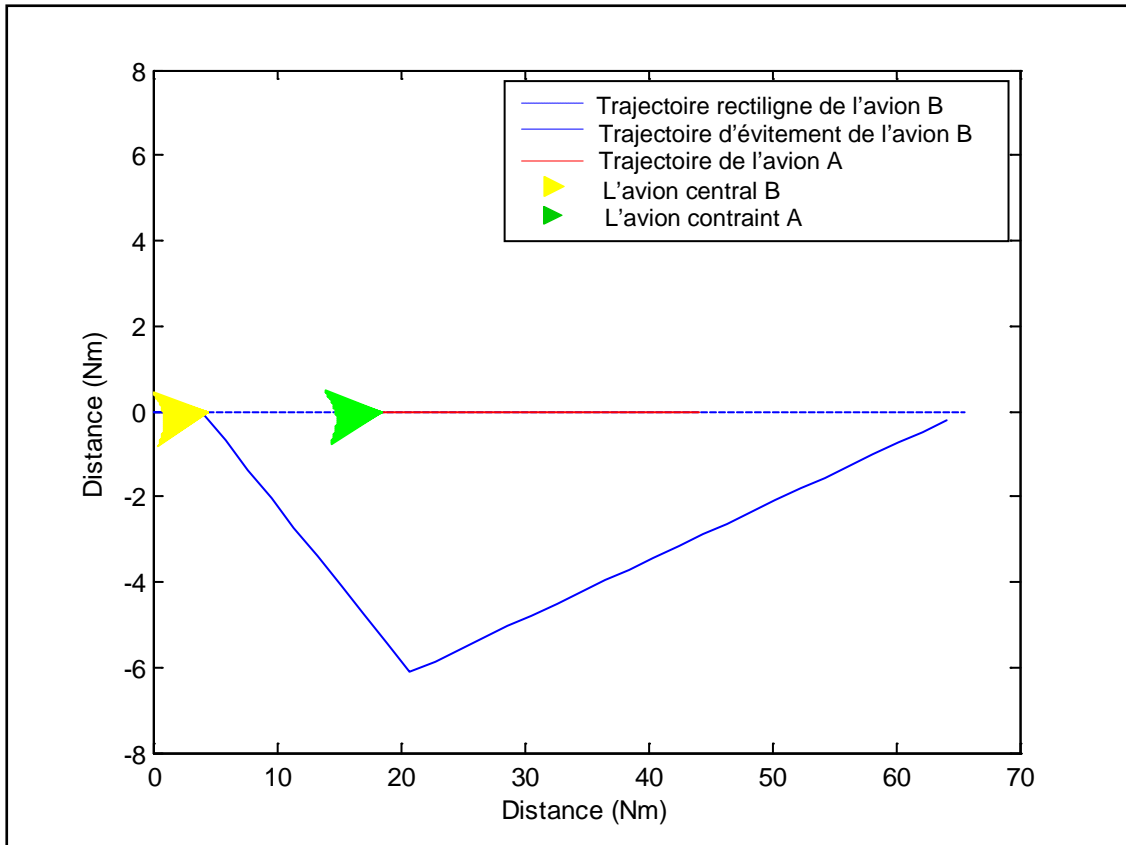


Figure 5-6 : Les trajectoires résolues par l'algorithme A*

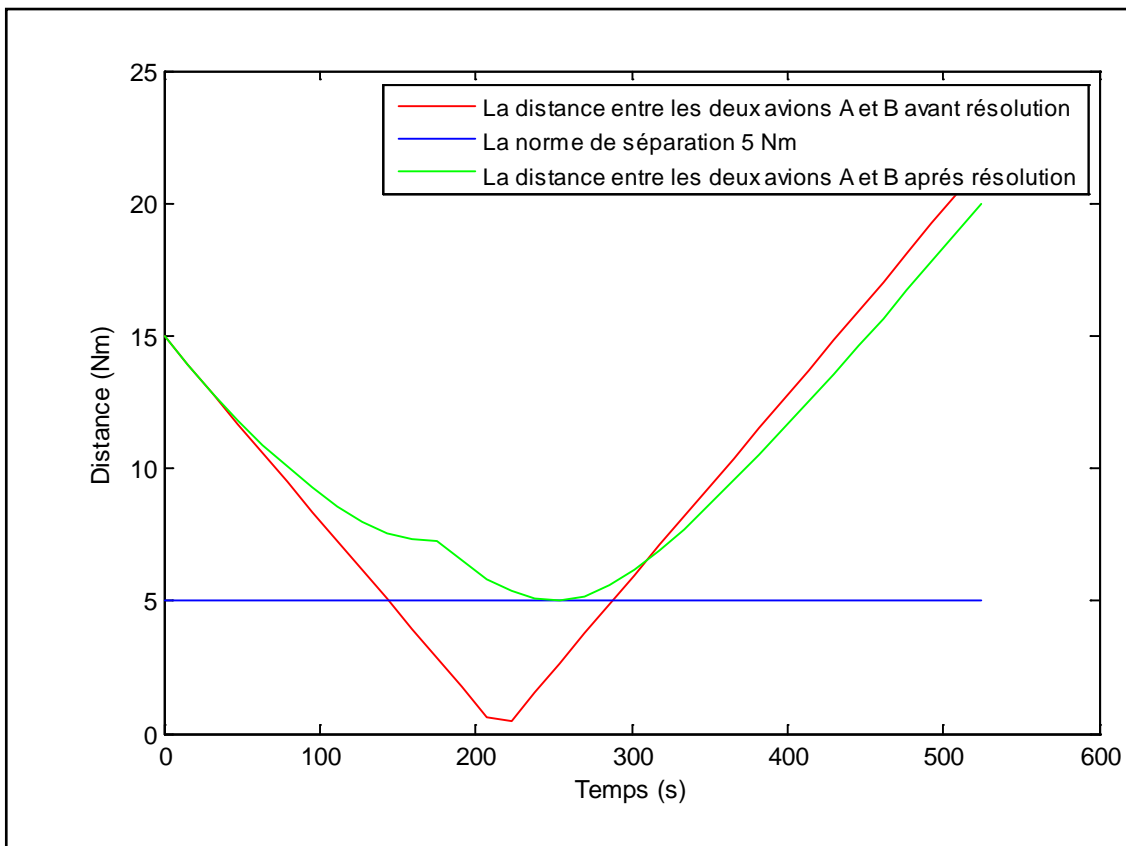


Figure 5-7 : Les distances de séparation entre les deux avions avant et après résolution par rapport à la norme de séparation « 5 Nm »

■ Deuxième cas :

➤ Tableau de données :

Tableau 5-4 : Données initiales des deux avions du premier cas

Avion	x_0 (Nm)	y_0 (Nm)	v (kt)	Cap initial(°)	Pas de temps(s)	Nbre de pas	Temps total(s)	x_{fin} (Nm)	y_{fin} (Nm)
A	0	0	390	90	15.3	35	535.5	58.01	0
B	60	0	400	270				0.5	0

➤ Tableaux des Résultats :

Tableau 5-5 : Paramètres d'évitement de l'avion central B

	Instant de débuts(s)	Durée(s)	Cap(°)
Tronçon n°1 (Etape E_0)	0	30.6	270
Tronçon n°2 (Etape E_1)	30.6	260.1	280
Tronçon n°3 (Etape E_2)	290.7	244.8	260.29

Tableau 5-6 : Temps total et temps de retard de l'avion central

Durée de trajectoire rectiligne(s)	Durée avec résolution de conflit(s)	t_{retard} (s)
535.5	543.2865	7.7865

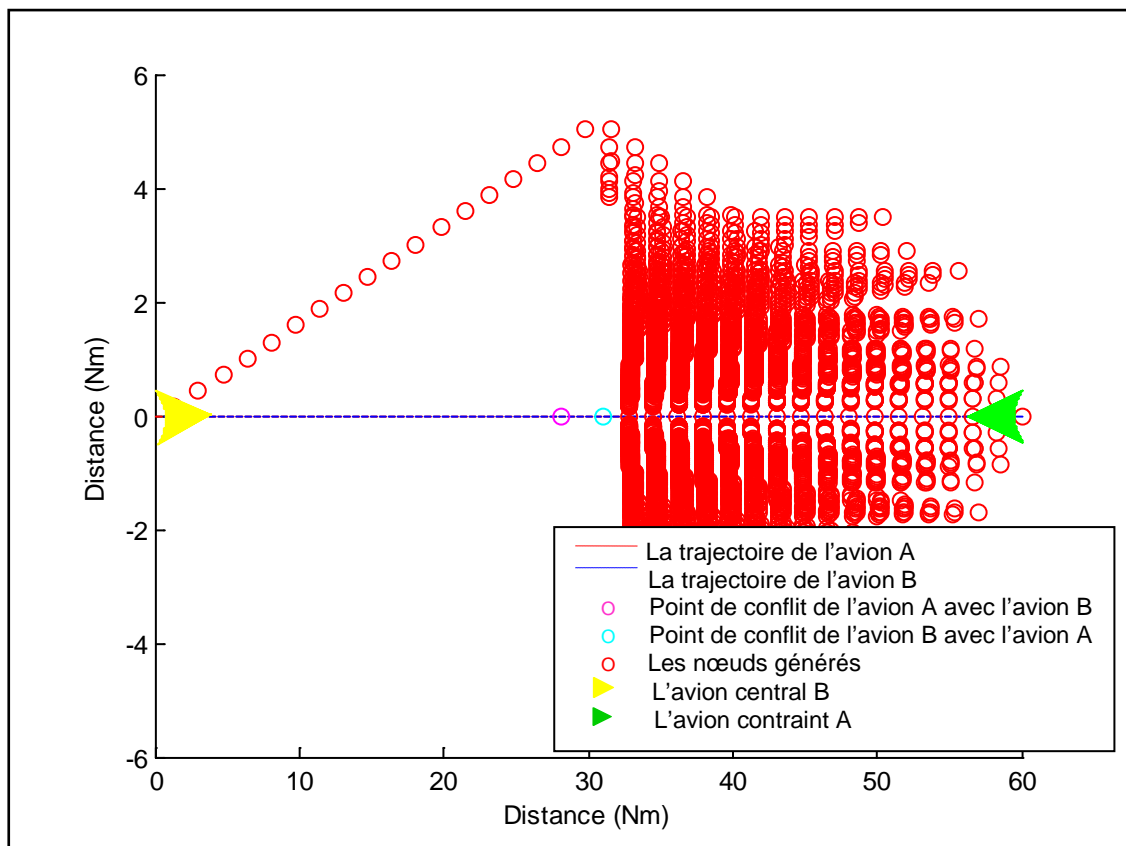


Figure 5-8 : Les nœuds de recherche de l'algorithme A*

Avion A ($x_0=0$ (Nm), $y_0=0$ (Nm), $v=390$ (kt), $cap=90$ (°))

Avion B ($x_0=60$ (Nm), $y_0=0$ (Nm), $v=400$ (kt), $cap=270$ (°))

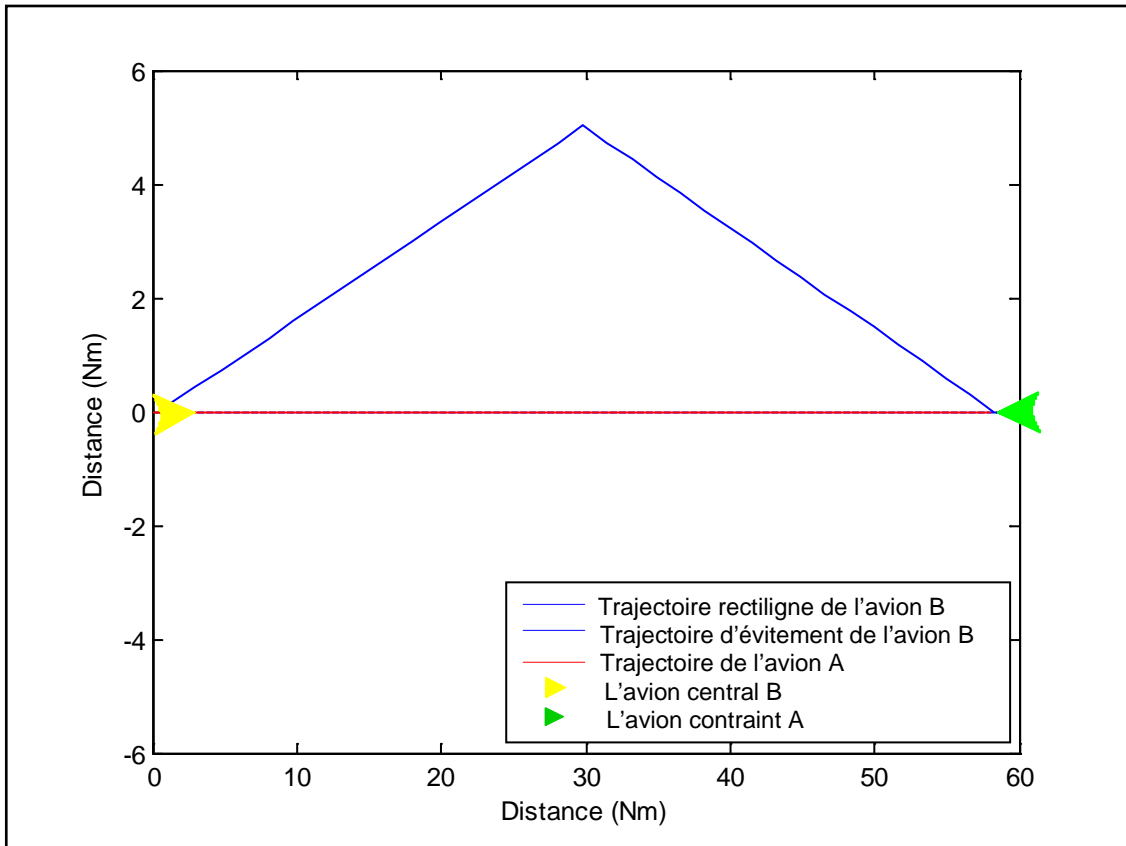


Figure 5-9 : Les trajectoires résolues par l'algorithme A*

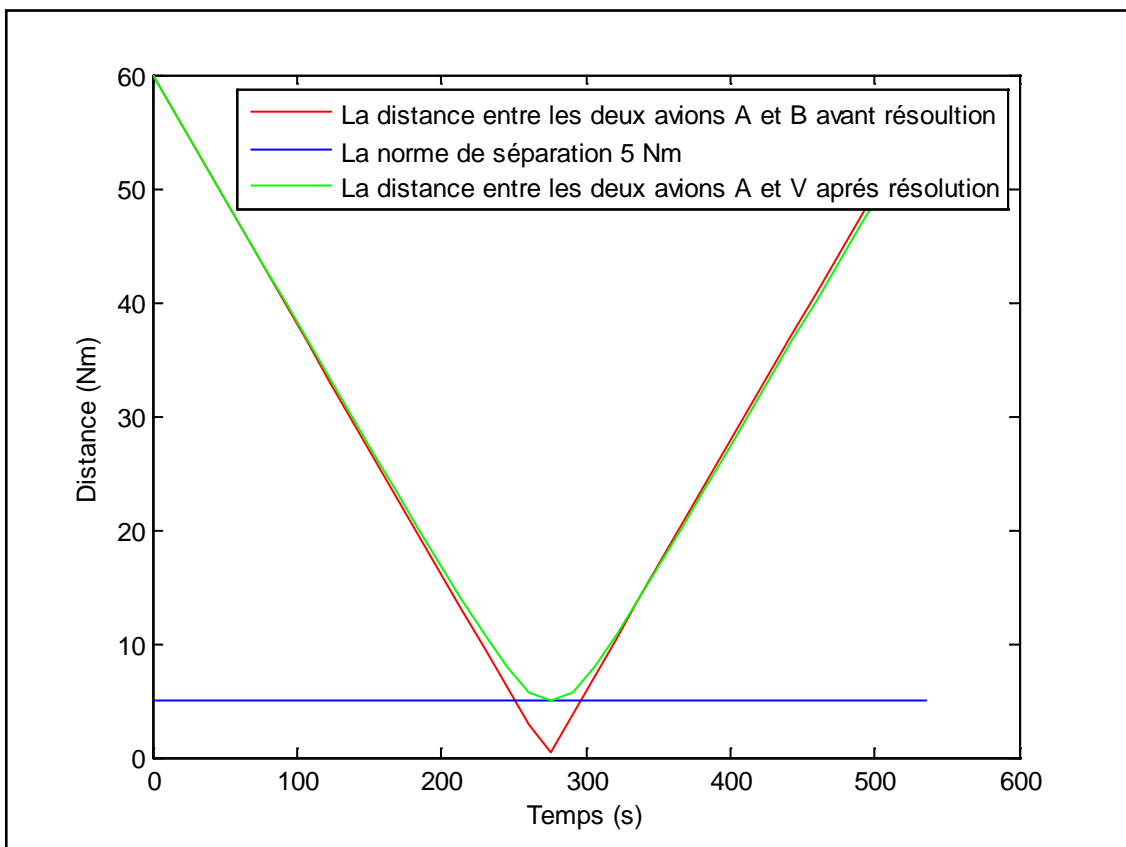


Figure 5-10 : Les distances de séparation entre les deux avions avant et après résolution par rapport à la norme de séparation « 5 Nm »

■ Troisième cas :

➤ Tableau de données :

Tableau 5-7 : Données initiales des deux avions du premier cas

Avion	x_0 (Nm)	y_0 (Nm)	v (kt)	Cap initial(°)	Pas de temps(s)	Nbre de pas	Temps total(s)	x_{fin} (Nm)	y_{fin} (Nm)
A	0	0	180	90	16.2	40	648	32.4	0
B	15	35	450	180				15	-46

➤ Tableaux des Résultats :

Tableau 5-8 : Paramètres d'évitements de l'avion central B

	Instant de débuts(s)	Durée(s)	Cap(°)
Tronçon n°1 (Etape E_0)	0	81	180
Tronçon n°2 (Etape E_1)	81	210.6	170
Tronçon n°3 (Etape E_2)	291.6	356.4	185.81

Tableau 5-9 : Temps total et temps de retard de l'avion central

Durée de trajectoire rectiligne(s)	Durée avec résolution de conflit(s)	t_{retard} (s)
648	653.05	5.05

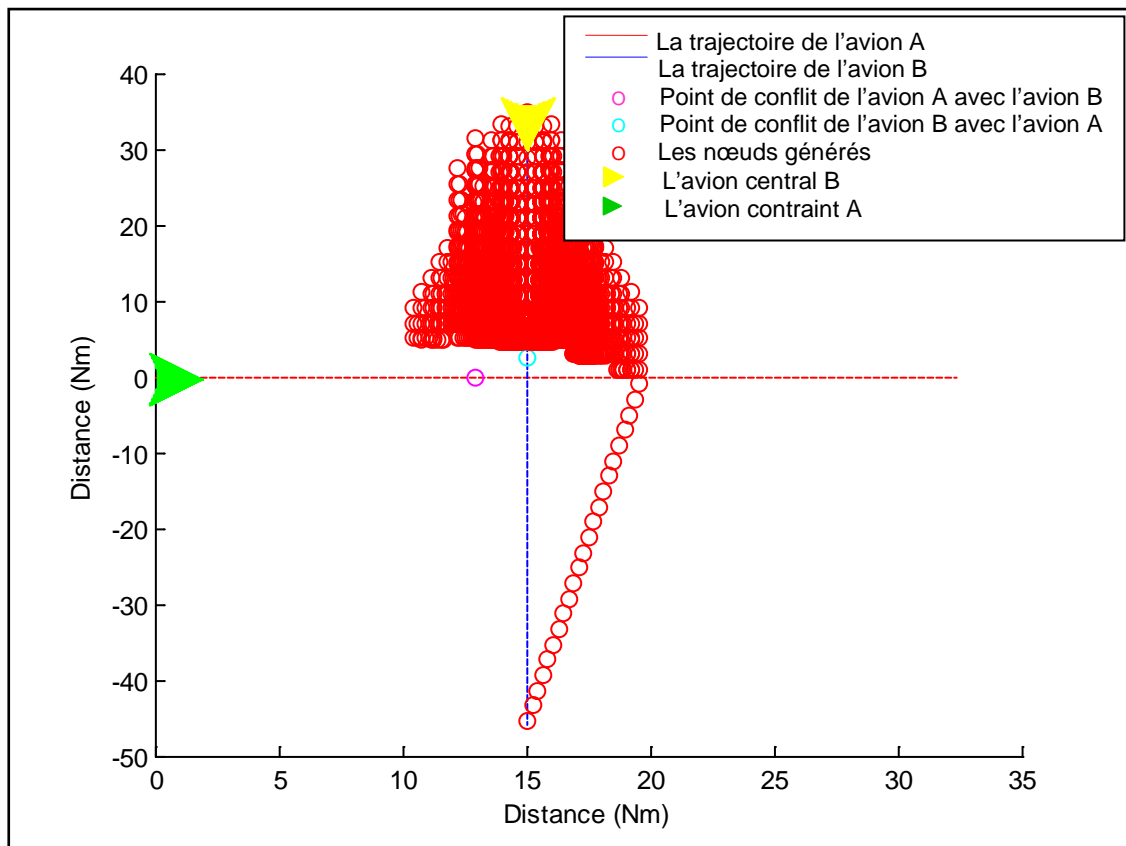


Figure 5-11 : Les nœuds de recherche de l'algorithme A*

Avion A ($x_0=0$ (Nm), $y_0=0$ (Nm), $v=180$ (kt), $cap=90$ (°))

Avion B ($x_0=15$ (Nm), $y_0=35$ (Nm), $v=450$ (kt), $cap=180$ (°))

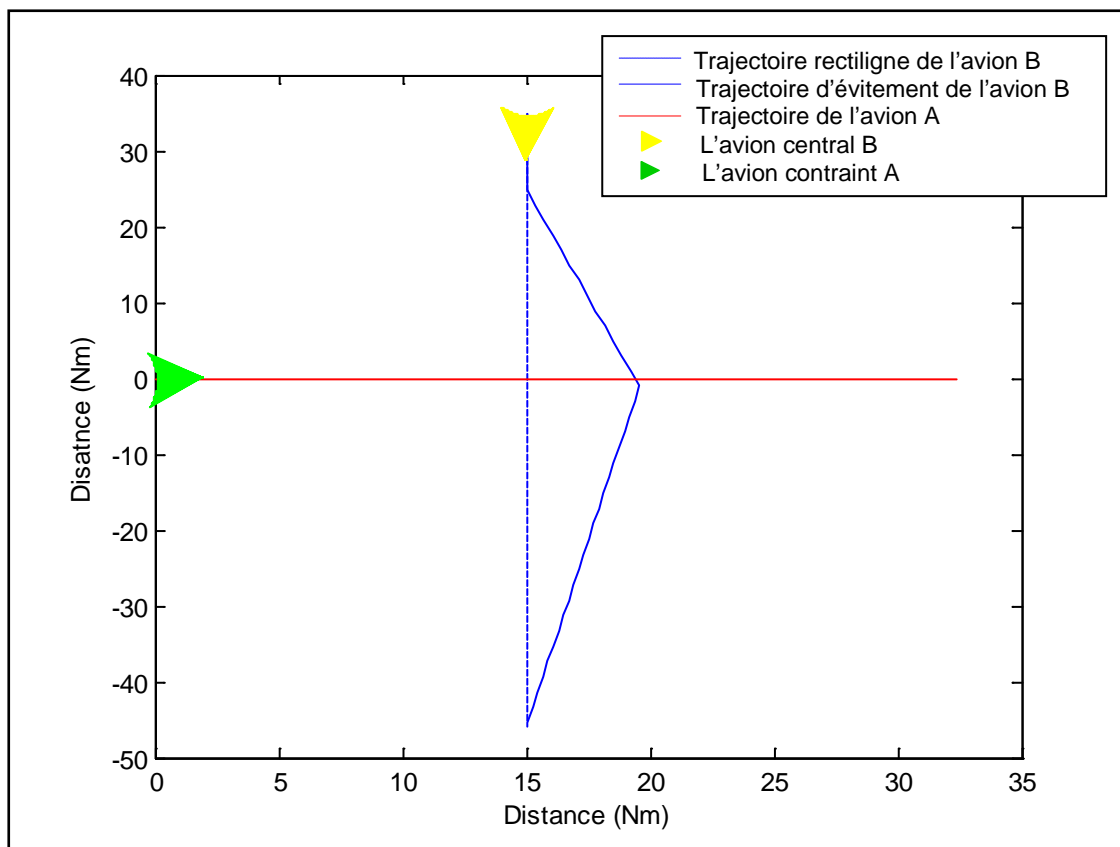


Figure 5-12 : Les trajectoires résolues par l'algorithme A*

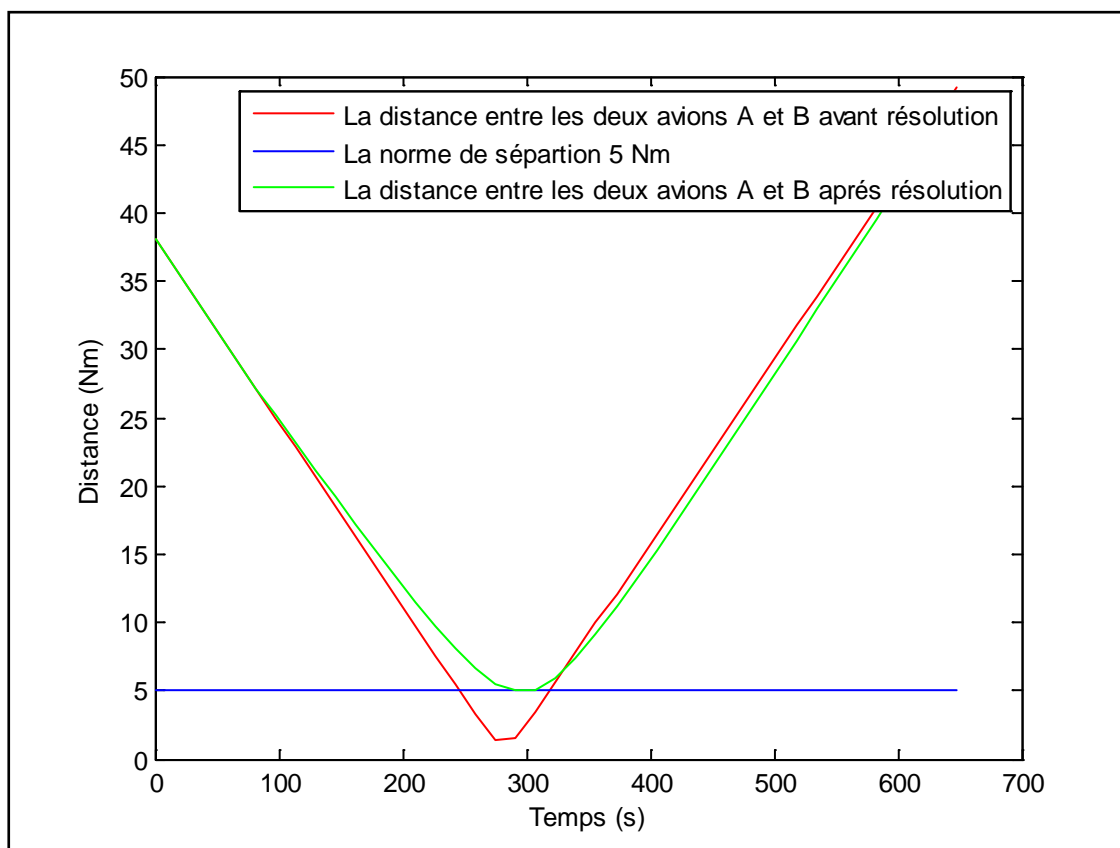


Figure 5-13 : Les distances de séparation entre les deux avions avant et après résolution par rapport à la norme de séparation « 5 Nm »

■ Quatrième cas :

➤ Tableau de données :

Tableau 5-10 : Données initiales des deux avions du premier cas

Avion	x_0 (Nm)	y_0 (Nm)	v (kt)	Cap initial(°)	Pas de temps(s)	Nbre de pas	Temps total(s)	x_{fin} (Nm)	y_{fin} (Nm)
A	0	0	190	90	13.4	40	536	28.2889	0
B	2	15	480	160				26.44	-52.16

➤ Tableaux des Résultats :

Tableau 5-11 : Paramètres d'évitements de l'avion central B

	Instant de débuts(s)	Durée(s)	Cap(°)
Tronçon n°1 (Etape E_0)	0	13.4	160
Tronçon n°2 (Etape E_1)	13.4	107.2	140
Tronçon n°3 (Etape E_2)	120.6	415.4	164.1

Tableau 5-12 : Temps total et temps de retard de l'avion central

Durée de trajectoire rectiligne(s)	Durée avec résolution de conflit(s)	t_{retard} (s)
536	544.06	8.06

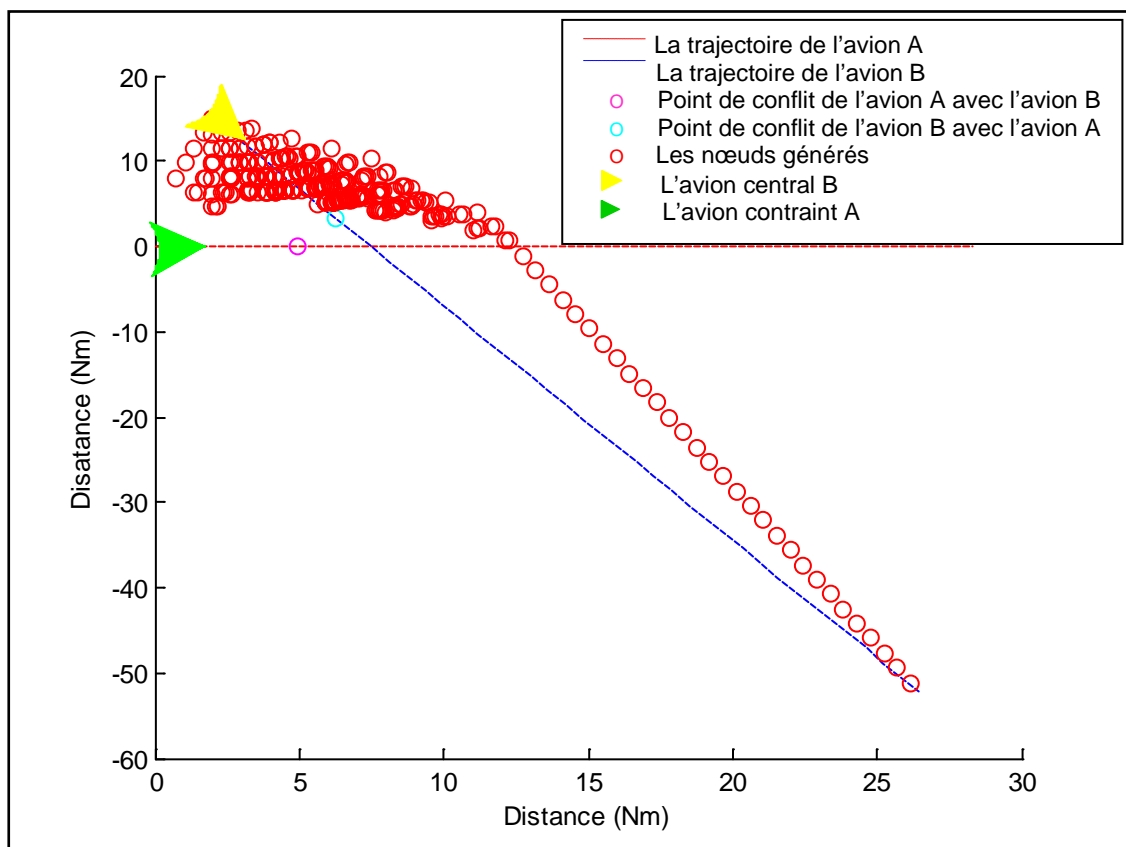


Figure 5-14 : Les nœuds de recherche de l'algorithme A*

Avion A ($x_0=0$ (Nm), $y_0=0$ (Nm), $v=190$ (kt), $cap=90$ (°))

Avion B ($x_0=2$ (Nm), $y_0=15$ (Nm), $v=480$ (kt), $cap=160$ (°))

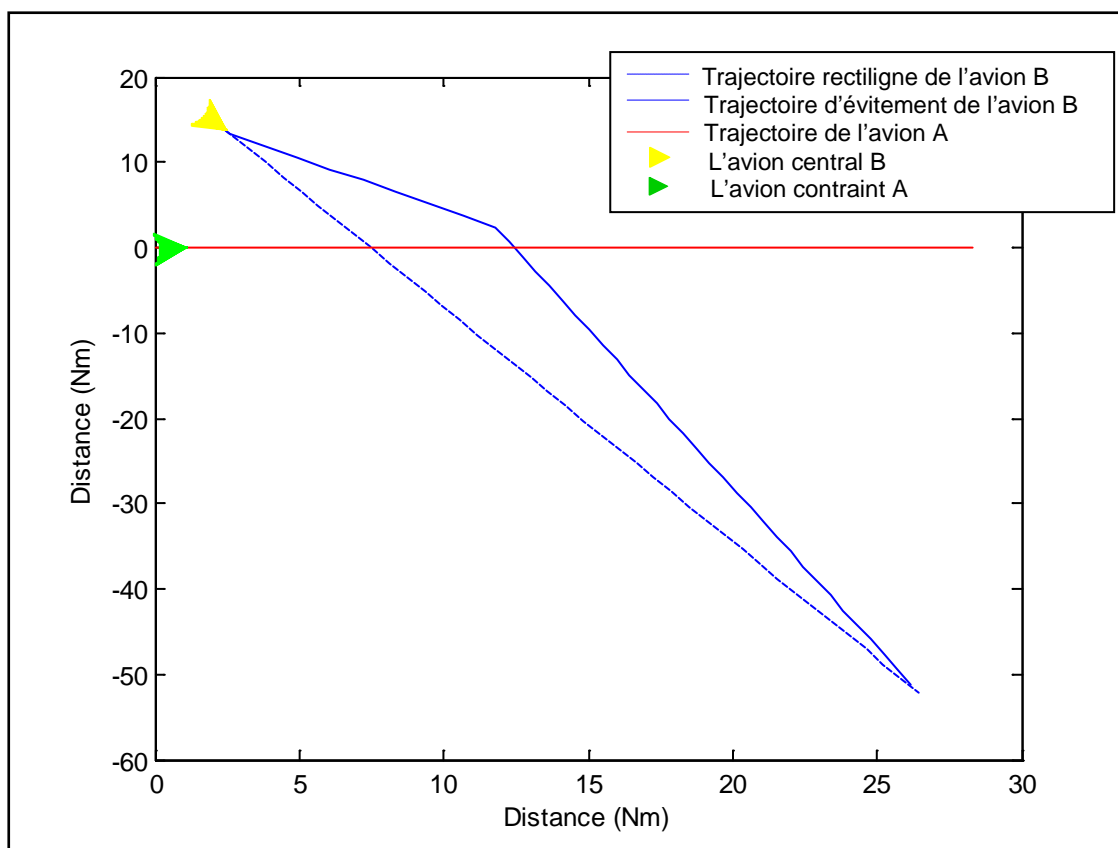


Figure 5-15 : Les trajectoires résolues par l'algorithme A*

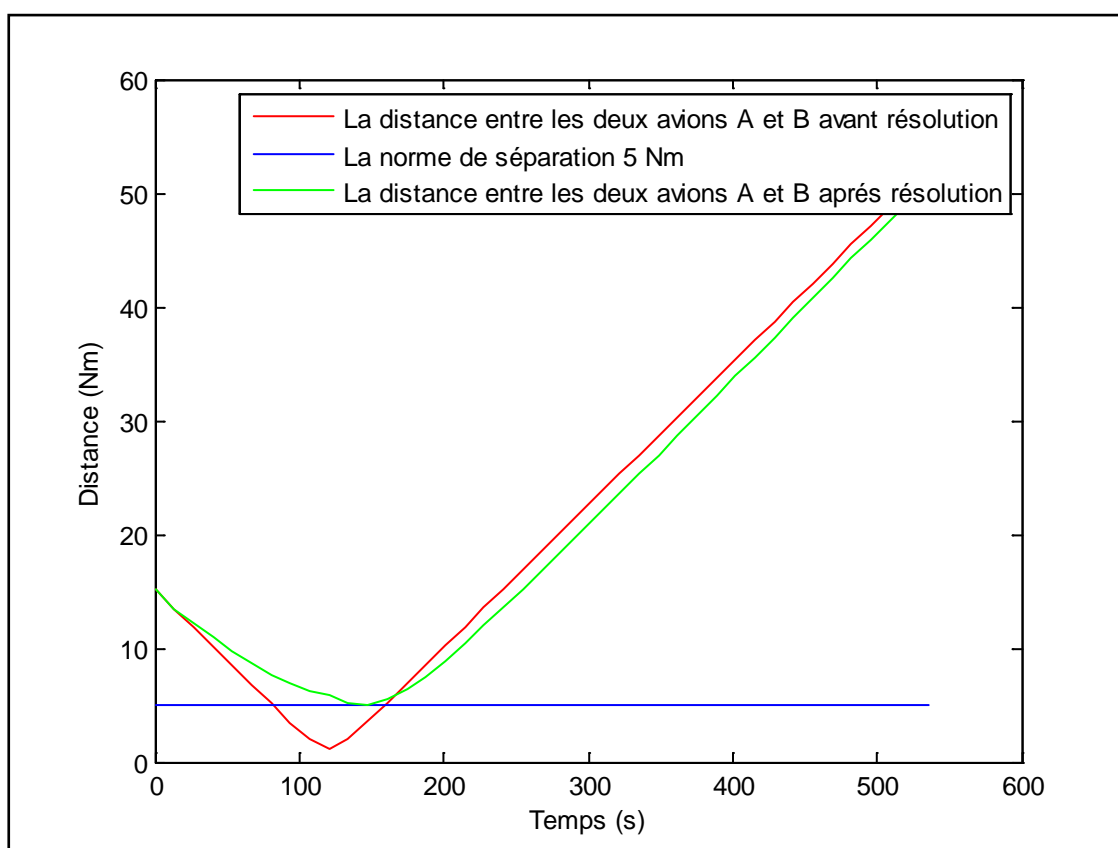


Figure 5-16 : Les distances de séparation entre les deux avions avant et après résolution par rapport à la norme de séparation « 5 Nm »

■ Cinquième cas :

➤ Tableau de données :

Tableau 5-13 : Données initiales des deux avions du premier cas

Avion	x_0 (Nm)	y_0 (Nm)	v (kt)	Cap initial(°)	Pas de temps(s)	Nbre de pas	Temps total(s)	x_{fin} (Nm)	y_{fin} (Nm)
A	0	0	350	90	15.4	40	616	59.89	0
B	30	20	470	210				-10.21	-49.65

➤ Tableaux des Résultats

Tableau 5-14 : Paramètres d'évitements de l'avion central B

	Instant de débuts(s)	Durée(s)	Cap(°)
Tronçon n°1 (Etape E_0)	0	61.6	210
Tronçon n°2 (Etape E_1)	61.6	123.2	190
Tronçon n°3 (Etape E_2)	184.8	431.2	215.49

Tableau 5-15 : Temps total et temps de retard de l'avion central

Durée de trajectoire rectiligne(s)	Durée avec résolution de conflit(s)	t_{retard} (s)
616	625.45	9.45

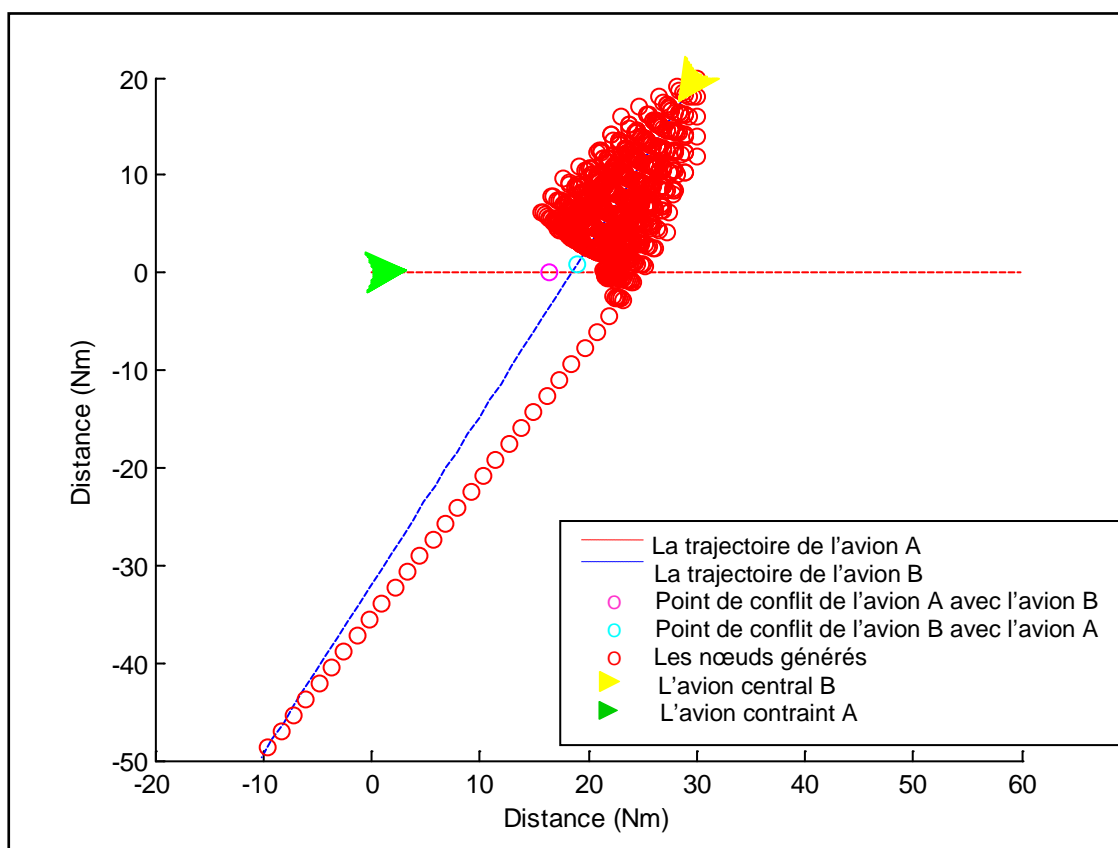


Figure 5-17 : Les nœuds de recherche de l'algorithme A*

Avion A ($x_0=0$ (Nm), $y_0=0$ (Nm), $v=350$ (kt), $cap=90$ (°))

Avion B ($x_0=30$ (Nm), $y_0=20$ (Nm), $v=470$ (kt), $cap=270$ (°))

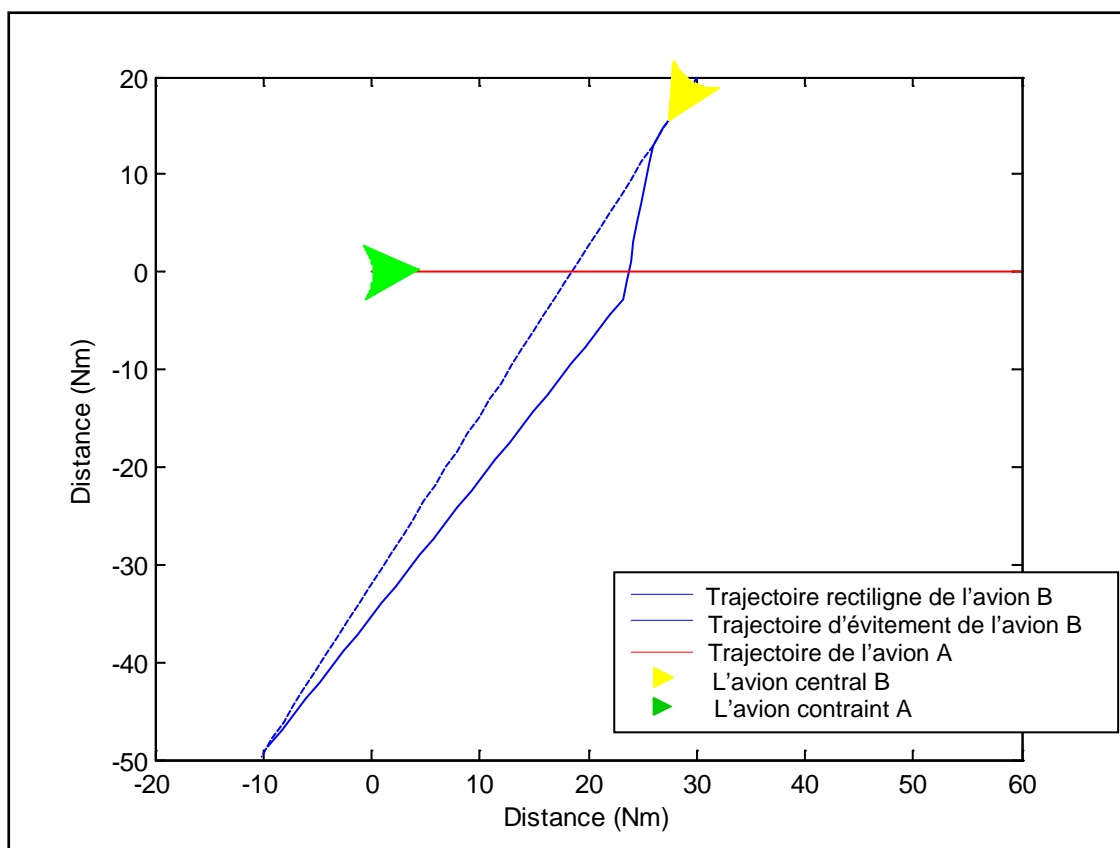


Figure 5-18 : Les trajectoires résolues par l'algorithme A*

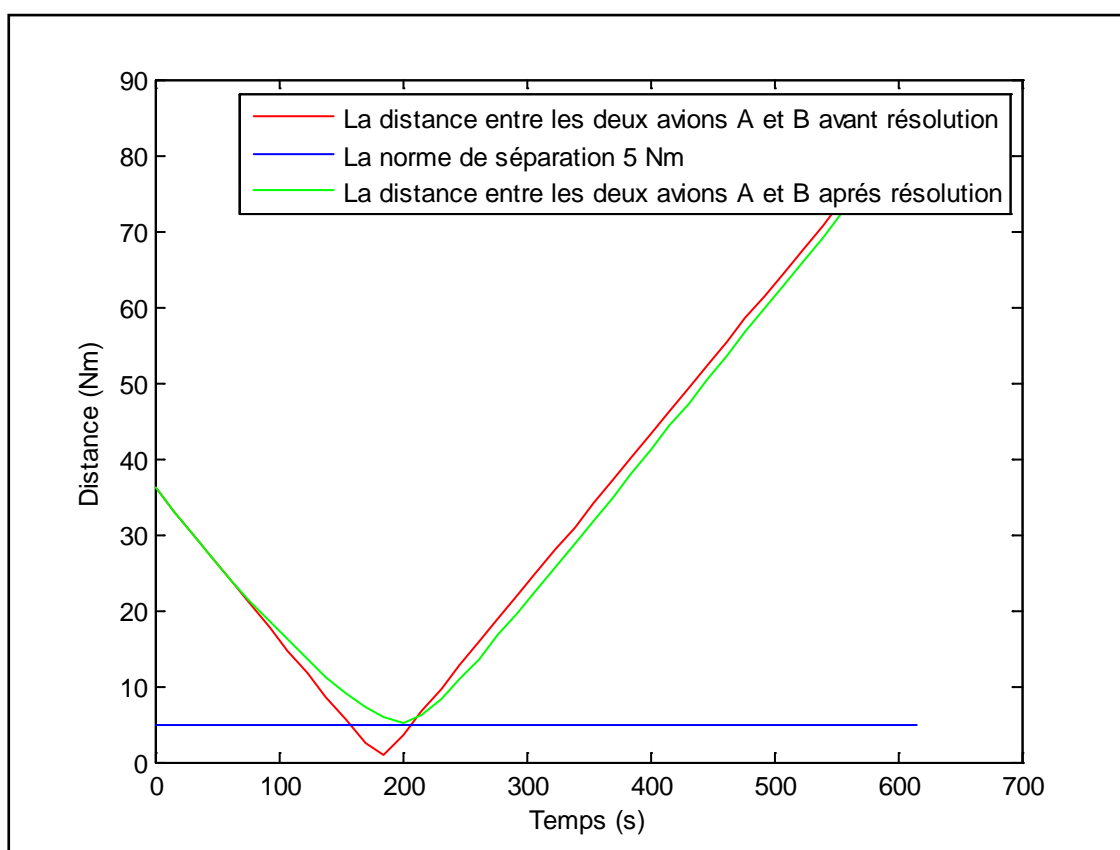


Figure 5-19 : Les distances de séparation entre les deux avions avant et après résolution par rapport à la norme de séparation « 5 Nm »

■ Sixième cas :

➤ Tableau de données :

Tableau 5-16 : Données initiales des deux avions du premier cas

Avion	x_0 (Nm)	y_0 (Nm)	v (kt)	Cap initial(°)	Pas de temps(s)	Nbre de pas	Temps total(s)	x_{fin} (Nm)	y_{fin} (Nm)
A	0	0	400	90	17.4	38	661.2	73.47	0
B	50	-10	220	300				15.01	10.20

➤ Tableaux des Résultats :

Tableau 5-17 : Paramètres d'évitements de l'avion central B

	Instant de débuts(s)	Durée(s)	Cap(°)
Tronçon n°1 (Etape E_0)	0	17.4	300
Tronçon n°2 (Etape E_1)	17.4	104.4	320
Tronçon n°3 (Etape E_2)	121.8	539.4	310.7

Tableau 5-18 : Temps total et temps de retard de l'avion central

Durée de trajectoire rectiligne(s)	Durée avec résolution de conflit(s)	t_{retard} (s)
661.2	680.48	19.28

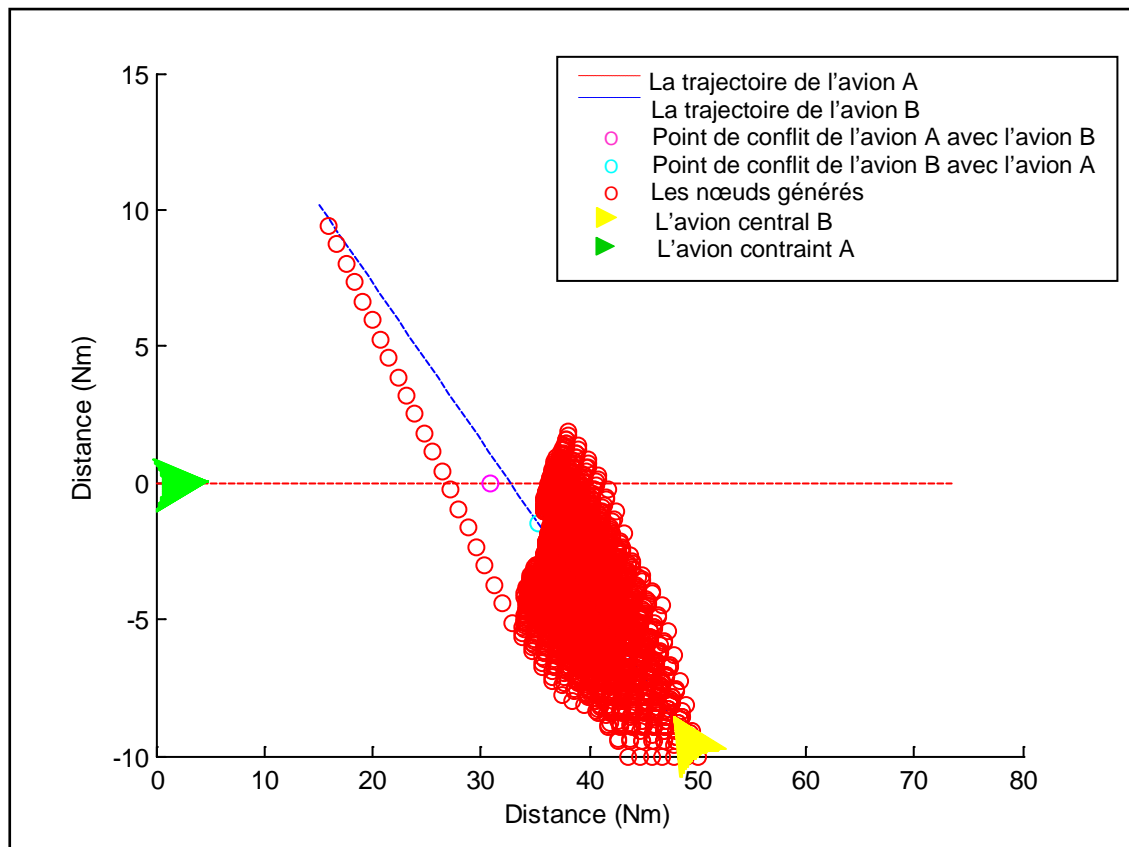


Figure 5-20 : Les nœuds de recherche de l'algorithme A*

Avion A ($x_0=0$ (Nm), $y_0=0$ (Nm), $v=400$ (kt), $cap=90$ (°))

Avion B ($x_0=50$ (Nm), $y_0=-10$ (Nm), $v=220$ (kt), $cap=300$ (°))

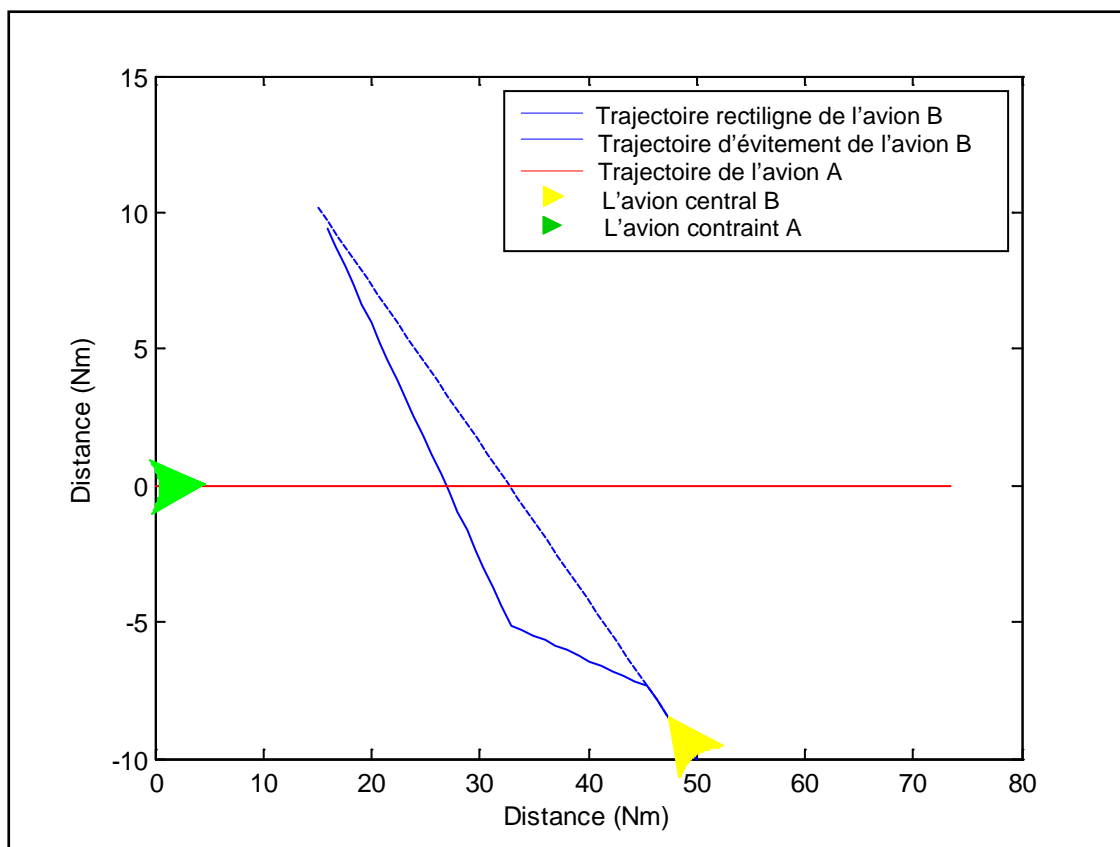


Figure 5-21 : Les trajectoires résolues par l'algorithme A*

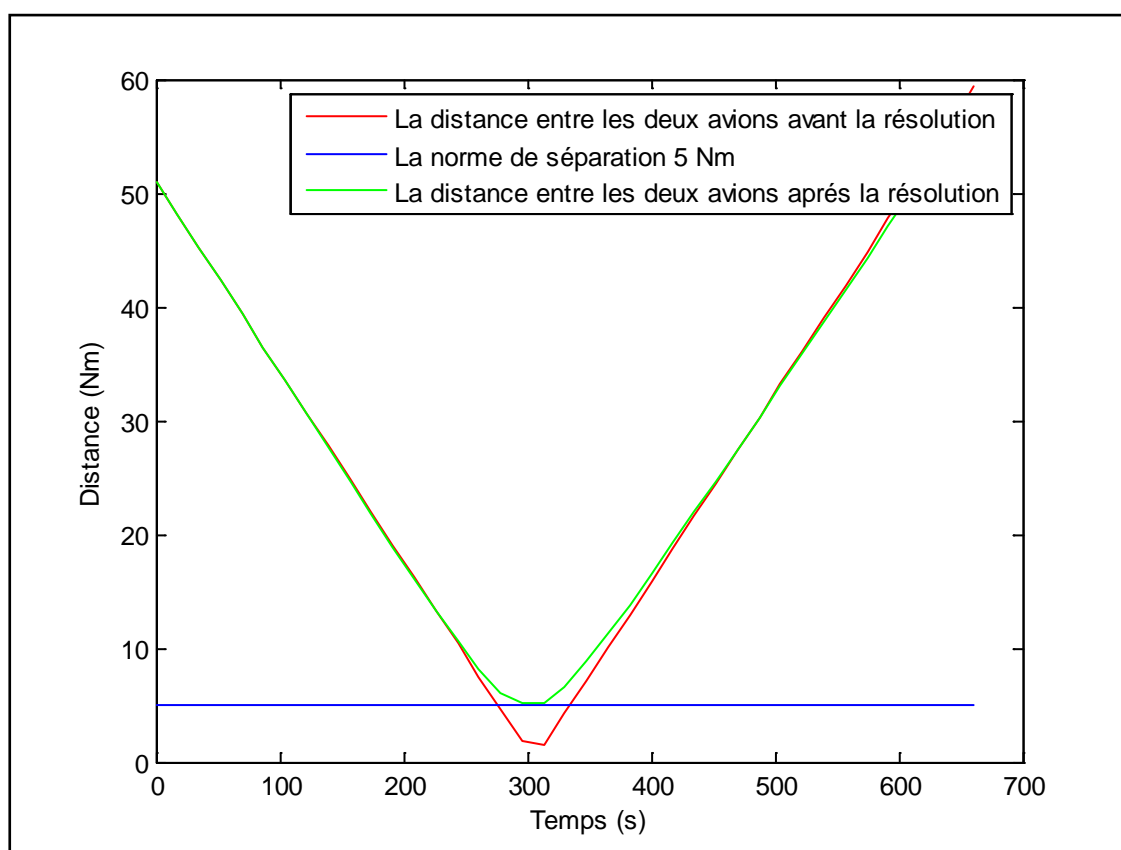


Figure 5-22 : Les distances de séparation entre les deux avions avant et après résolution par rapport à la norme de séparation « 5 Nm »

■ Septième cas :

➤ Tableau de données :

Tableau 5-19 : Données initiales des deux avions du premier cas

Avion	x_0 (Nm)	y_0 (Nm)	v (kt)	Cap initial(°)	Pas de temps(s)	Nbre de pas	Temps total(s)	x_{fin} (Nm)	y_{fin} (Nm)
A	0	0	300	90	17.3	35	605.5	50.46	0
B	5	15	270	40				34.19	19.79

➤ Tableaux des Résultats :

Tableau 5-20 : Paramètres d'évitements de l'avion central B

	Instant de débuts(s)	Durée(s)	Cap(°)
Tronçon n°1 (Etape E_0)	0	34.6	40
Tronçon n°2 (Etape E_1)	34.6	121.1	20
Tronçon n°3 (Etape E_2)	155.7	449.8	45.18

Tableau 5-21: Temps total et temps de retard de l'avion central

Durée de trajectoire rectiligne(s)	Durée avec résolution de conflit(s)	t_{retard} (s)
605.5	614.67	9.79

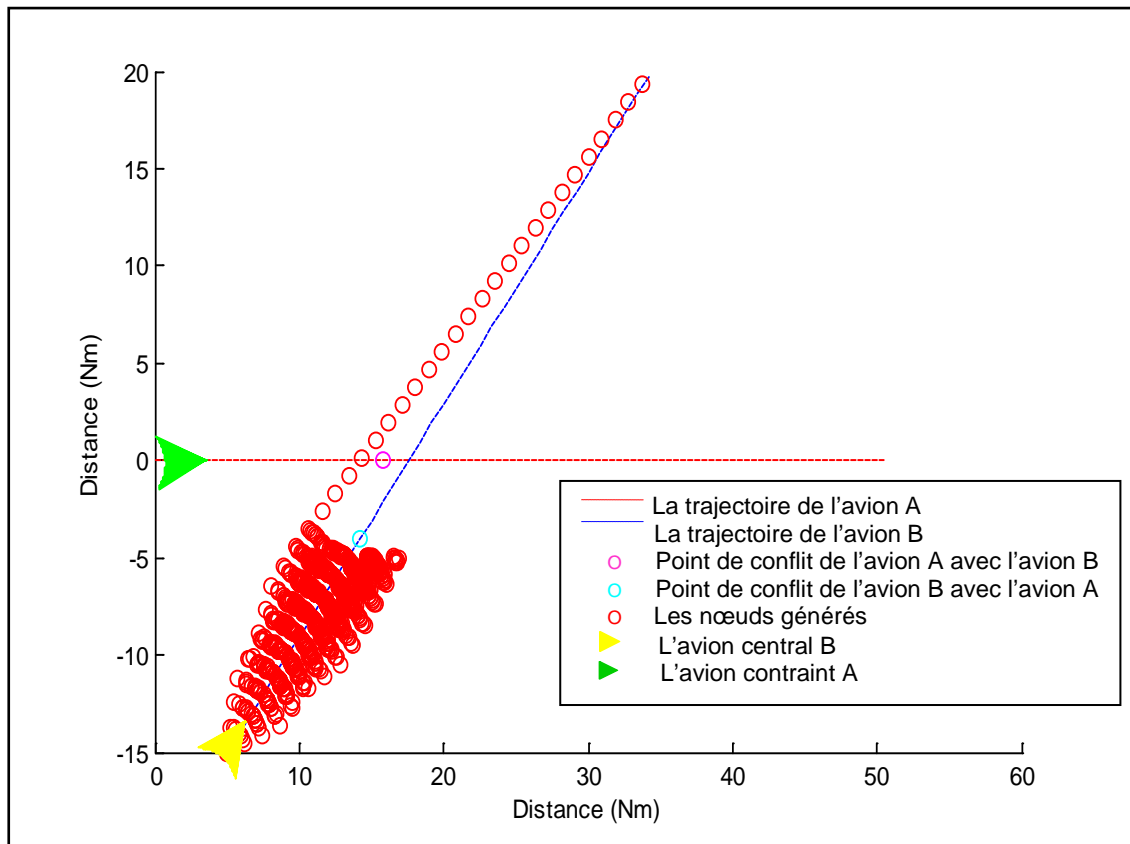


Figure 5-23 : Les nœuds de recherche de l'algorithme A*

Avion A ($x_0=0$ (Nm), $y_0=0$ (Nm), $v=300$ (kt), $cap=90$ (°))

Avion B ($x_0=5$ (Nm), $y_0=15$ (Nm), $v=270$ (kt), $cap=40$ (°))

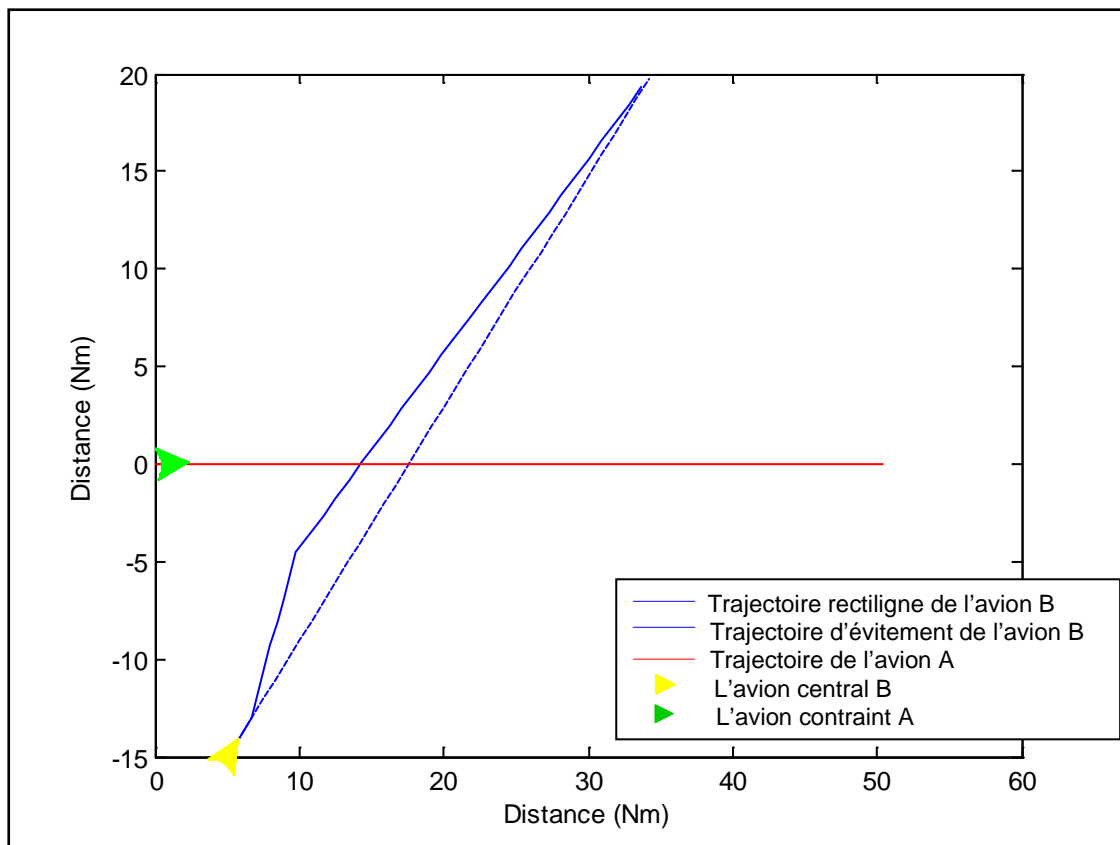


Figure 5-24 : Les trajectoires résolues par l'algorithme A*

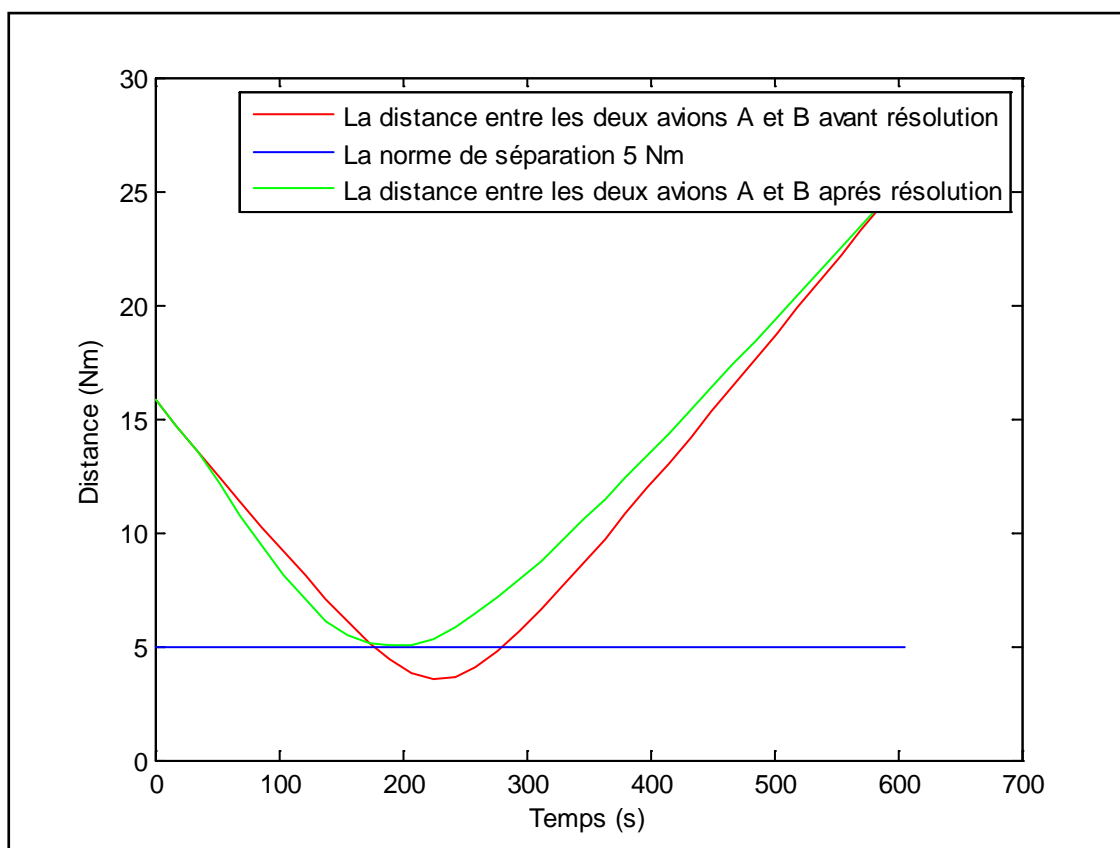


Figure 5-25 : Les distances de séparation entre les deux avions avant et après résolution par rapport à la norme de séparation « 5 Nm »

■ Huitième cas :

➤ Tableau de données :

Tableau 5-22 : Données initiales des deux avions du premier cas

Avion	x_0 (Nm)	y_0 (Nm)	v (kt)	Cap initial(°)	Pas de temps(s)	Nbre de pas	Temps total (s)	x_{fin} (Nm)	y_{fin} (Nm)
A	10	15	270	150	17.3	35	605.5	32.71	-24.33
B	30	15	335	220				-6.22	-28.16

➤ Tableaux des Résultats :

Tableau 5-23 : Paramètres d'évitements de l'avion central B

	Instant de débuts(s)	Durée(s)	Cap(°)
Tronçon n°1 (Etape E_0)	0	17.3	220
Tronçon n°2 (Etape E_1)	17.3	224.9	200
Tronçon n°3 (Etape E_2)	242.2	363.3	231.54

Tableau 5-24 : Temps total et temps de retard de l'avion central

Durée de trajectoire rectiligne(s)	Durée avec résolution de conflit(s)	t_{retard} (s)
605.5	626.83	21.33

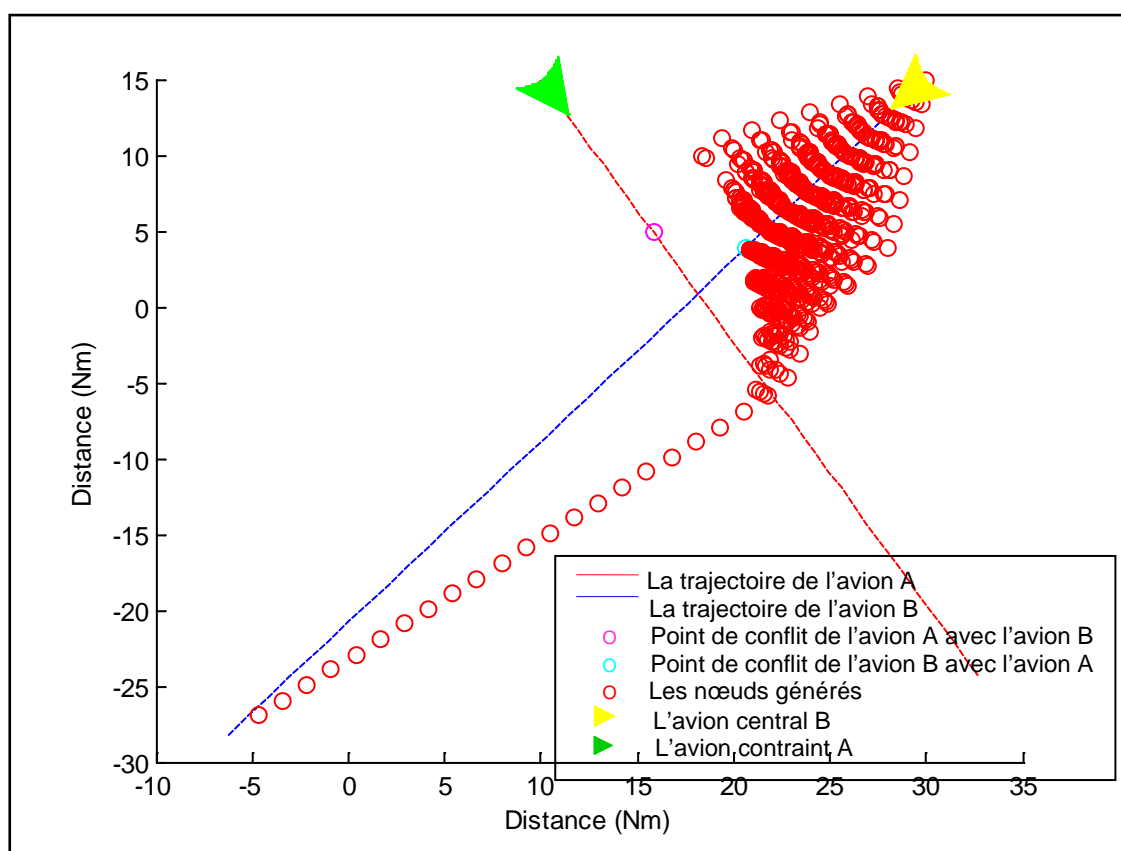


Figure 5-26 : Les nœuds de recherche de l'algorithme A*

Avion A ($x_0=10$ (Nm), $y_0=15$ (Nm), $v=270$ (kt), $cap=150$ (°))

Avion B ($x_0=30$ (Nm), $y_0=15$ (Nm), $v=335$ (kt), $cap=220$ (°))

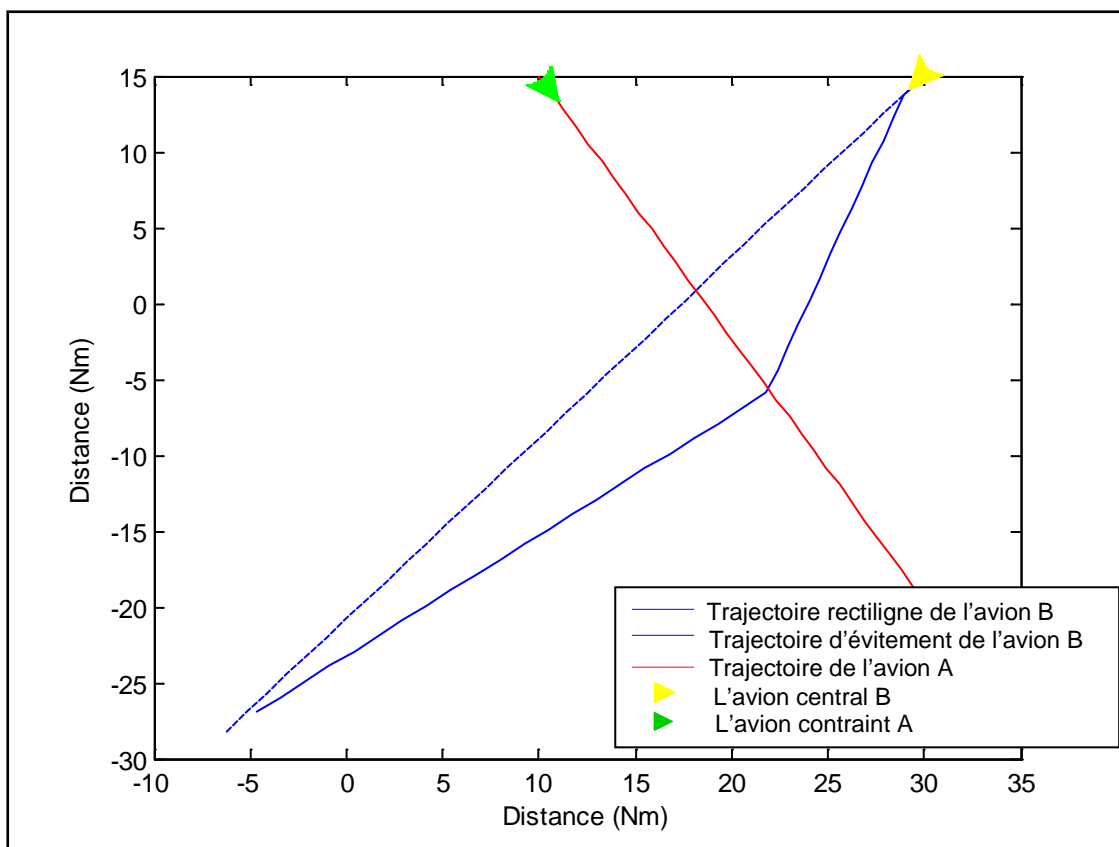


Figure 5-27 : Les trajectoires résolues par l'algorithme A*

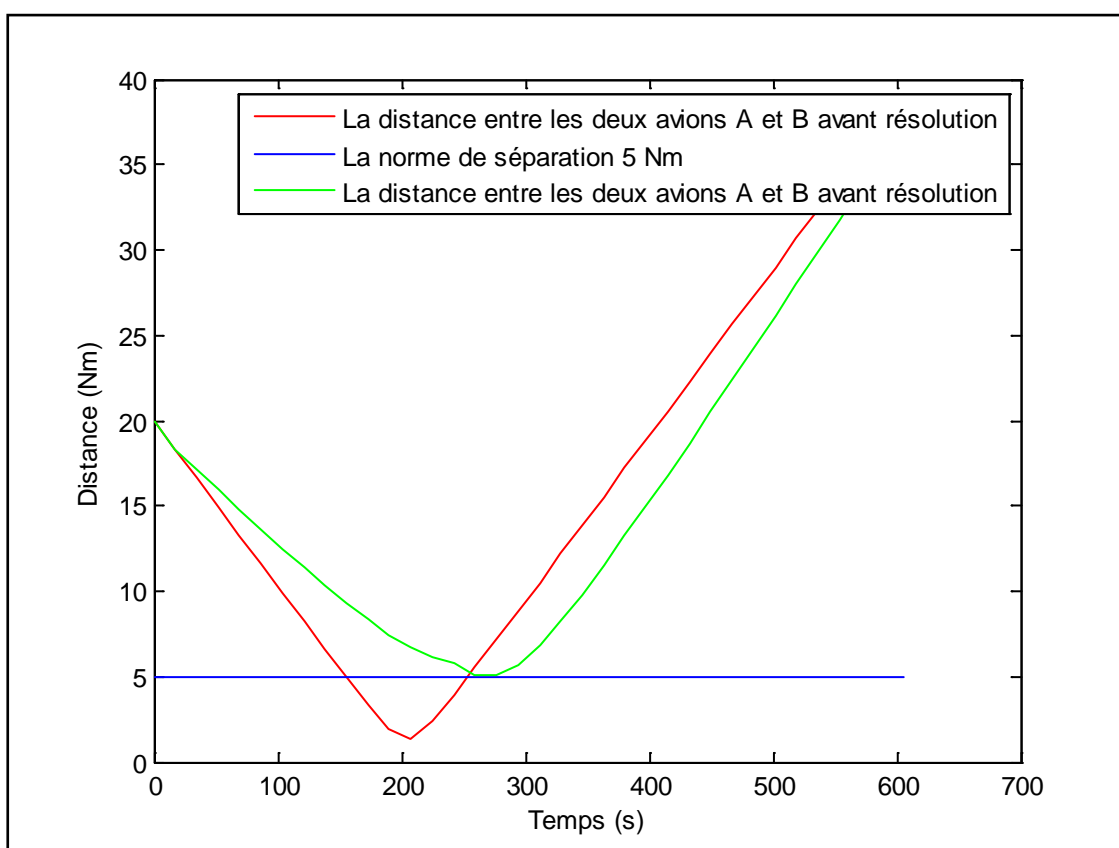


Figure 5-28 : Les distances de séparation entre les deux avions avant et après résolution par rapport à la norme de séparation « 5 Nm »

5.6.2 Cas de n avions

5.6.2.1. Exemple 1 : n=3

■ Données :

- Pas de temps=15.9
- Nombre de pas=30
- Temps total=477

Tableau 5-25 : Données initiales des trois avions

	$x_0(Nm)$	$y_0(Nm)$	v (kt)	Cap (°)	$x_{fin}(Nm)$	$y_{fin}(Nm)$
Avion A	20	0	450	90	79.63	0
Avion B	55	11	420	230	12.37	-24.77
Avion C	59	-6	420	290	6.71	13.03

■ Résultats :

➤ Premier cas :

Tableau 5-26 : Données initiales des deux avions du cas « A>B>C »

Ordre	Avion A	Avion B	Avion C
Temps Total pour atteindre le point de destination de chaque avions (s)	477	486.05	540.87
T_{retard} pour chaque avion (s)	0	9.05	63.87
Somme des retard (s)	72.92		

➤ Deuxième cas :

Tableau 5-27 : Données initiales des deux avions du cas « A>C>B »

Ordre	Avion A	Avion C	Avion B
Temps Total pour atteindre le point de destination de chaque avions (s)	477	487.43	491.07
T_{retard} pour chaque avion (s)	0	10.43	14.07
Somme des retard (s)	24.5		

➤ Troisième cas :

Tableau 5-28 : Données initiales des deux avions du cas « C>A>B »

Ordre	Avion C	Avion A	Avion B
Temps Total pour atteindre le point de destination de chaque avions (s)	477	484.39	550.40
T_{retard} pour chaque avion (s)	0	7.39	73.40
Somme des retard (s)	80.79		

➤ Quatrième cas :

Tableau 5-29 : Données initiales des deux avions du cas « C>B>A »

Ordre	Avion C	Avion B	Avion A
Temps Total pour atteindre le point de destination de chaque avions (s)	477	550.40	484.39
T_{retard} pour chaque avion (s)	0	73.40	7.39
Somme des retard (s)	80.79		

➤ Cinquième cas :

Tableau 5-30 : Données initiales des deux avions du cas « B>A>C »

Ordre	Avion B	Avion A	Avion C
Temps Total pour atteindre le point de destination de chaque avions (s)	477	484.32	507.49
T_{retard} pour chaque avion (s)	0	7.32	30.49
Somme des retard (s)	37.81		

➤ Sixième cas :

Tableau 5-31 : Données initiales des deux avions du cas « B>C>A »

Ordre	Avion B	Avion C	Avion A
Temps Total pour atteindre le point de destination de chaque avions (s)	477	492.77	493.22
T_{retard} pour chaque avion (s)	0	15.77	16.22
Somme des retard (s)	31.99		

■ Ordonner les cas précédents par ordre croissant en fonction du temps de retard :

1. Deuxième cas.
2. Sixième cas
3. Cinquième cas
4. Premier cas
5. Troisième et quatrième cas

■ Observations :

Dans cette section on va donner les deux cas (le meilleur temps de retard, le mauvais temps de retard).

➤ Le meilleur cas : le deuxième

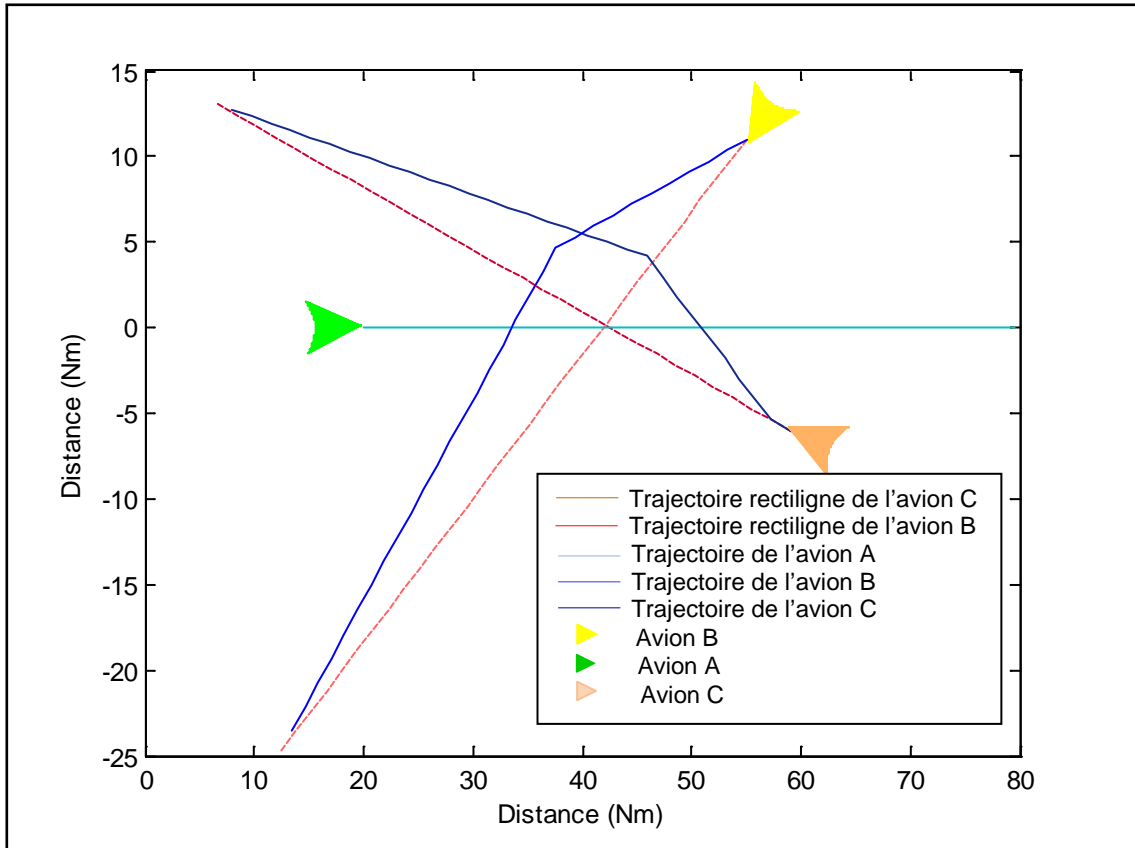


Figure 5-29 : Les trajectoires résolues par l'algorithme A*

Avion A ($x_0=20$ (Nm), $y_0=0$ (Nm), $v=450$ (kt), $\text{cap}=90$ ($^\circ$))

Avion C ($x_0=59$ (Nm), $y_0=-6$ (Nm), $v=420$ (kt), $\text{cap}=290$ ($^\circ$))

Avion B ($x_0=55$ (Nm), $y_0=11$ (Nm), $v=420$ (kt), $\text{cap}=230$ ($^\circ$))

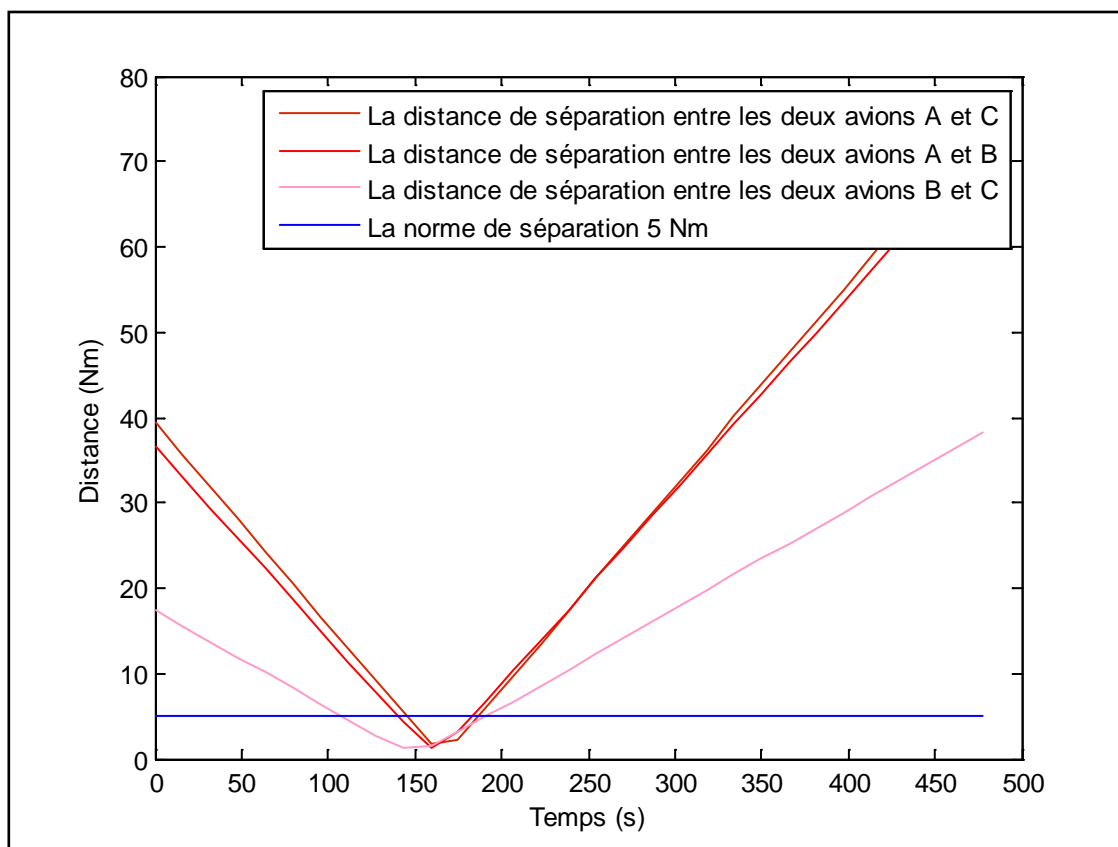


Figure 5-30 : Les distances de séparation entre les paires d'avions « (A,B), (A,C) et (B,C) » avant la résolution par rapport à la norme de séparation « 5 Nm »

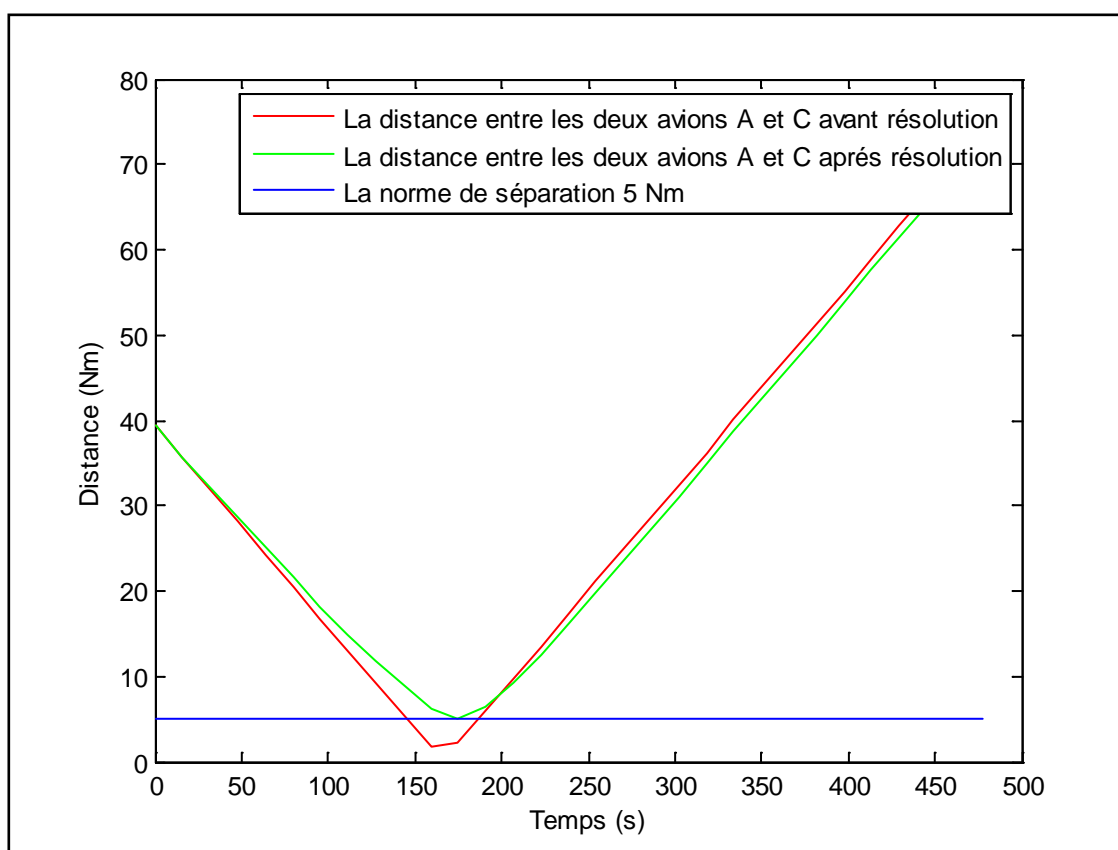


Figure 5-31 : Les distances de séparation entre les deux avions A et C avant et après la résolution par rapport à la norme de séparation « 5 Nm »

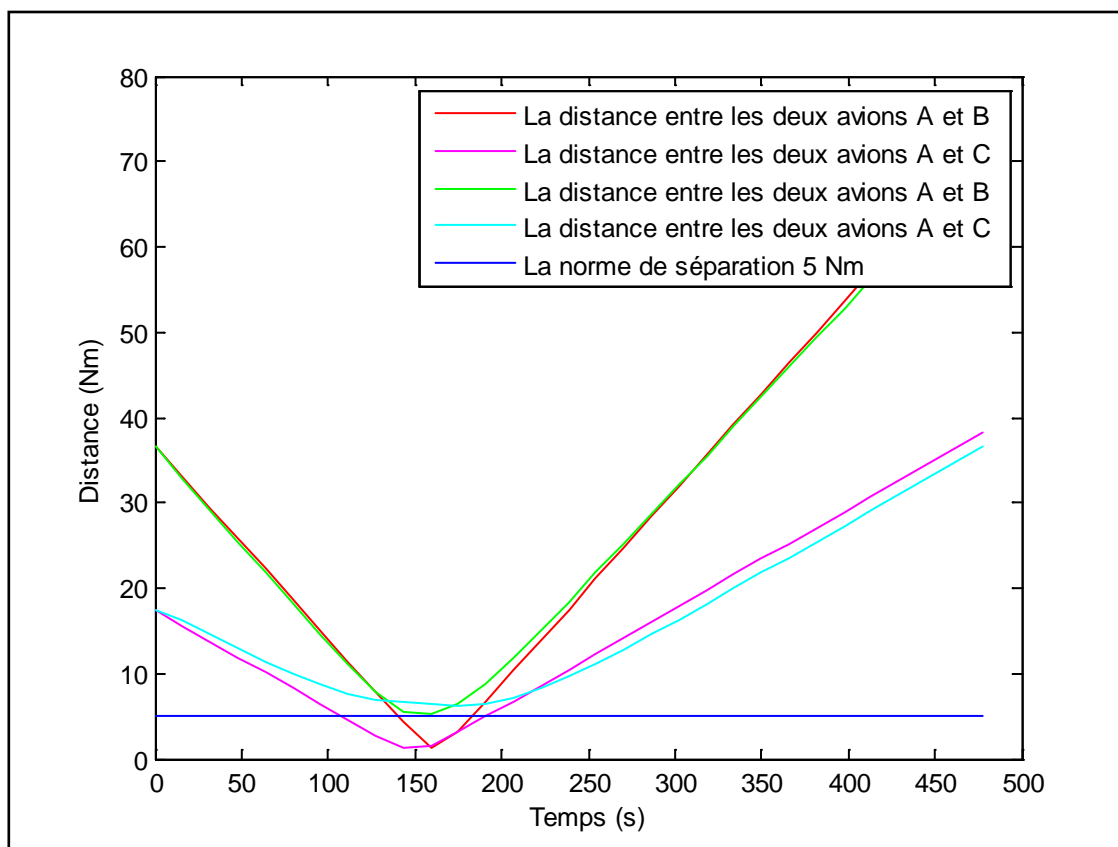


Figure 5-32 : Les distances de séparation entre les paires d'avions « (A,B) et (A,C) » avant et après la résolution par rapport à la norme de séparation « 5 Nm »

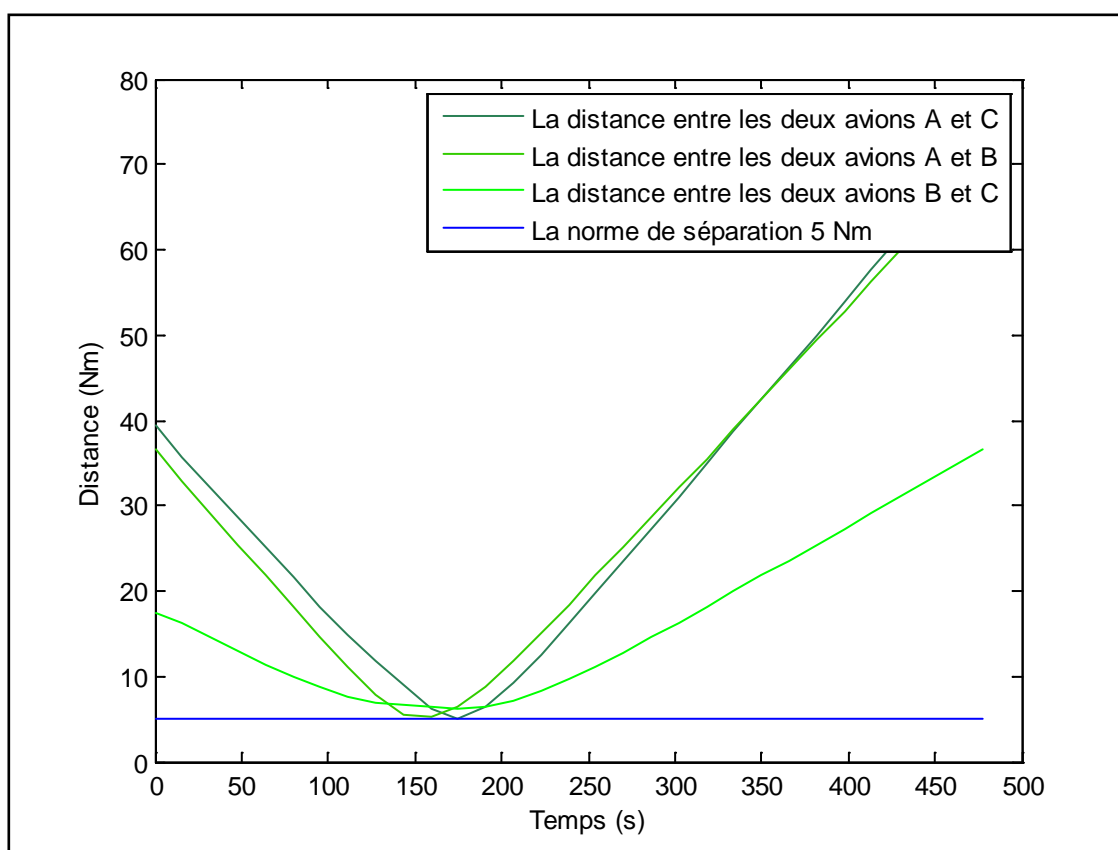


Figure 5-33 : Les distances de séparation entre les paires d'avions « (A,B), (A,C) et (B,C) » après la résolution par rapport à la norme de séparation « 5 Nm »

➤ Le mauvais cas : le quatrième

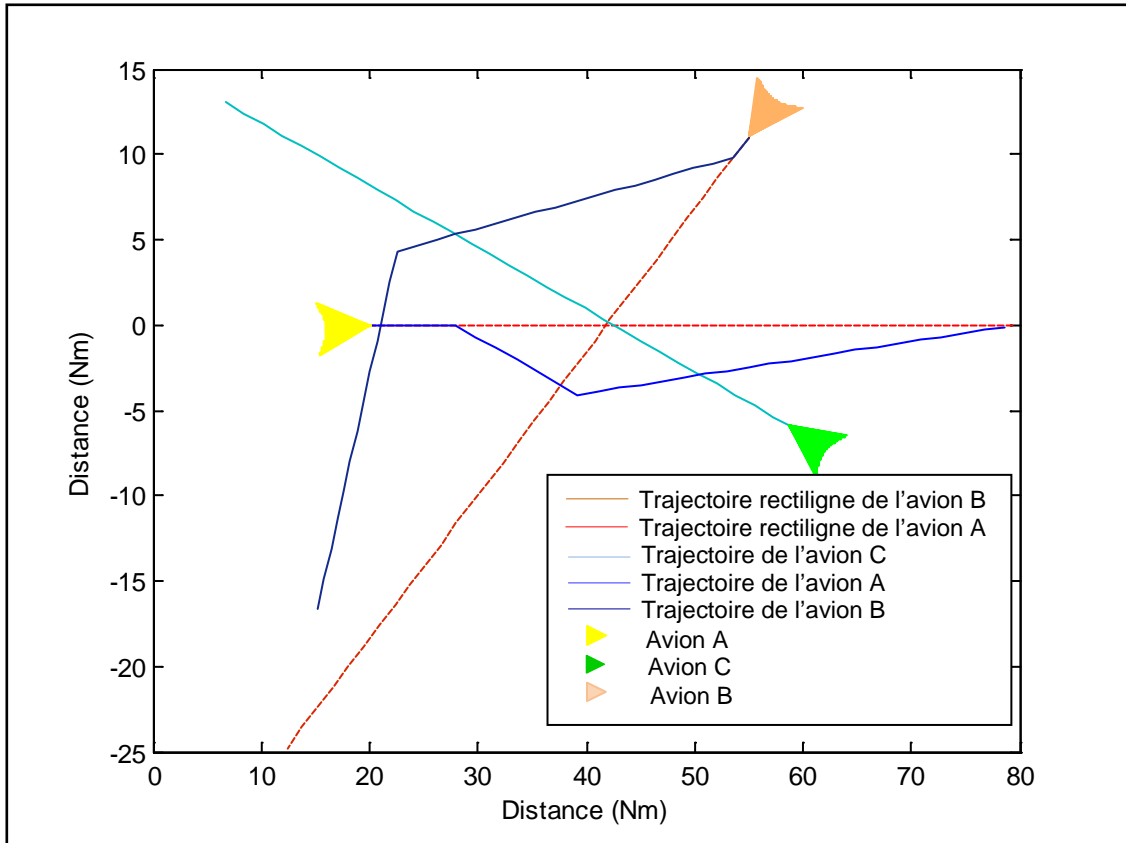


Figure 5-34 : Les trajectoires résolues par l'algorithme A*

Avion C ($x_0=59$ (Nm), $y_0=-6$ (Nm), $v=420$ (kt), $cap=290$ (°))

Avion B ($x_0=55$ (Nm), $y_0=11$ (Nm), $v=420$ (kt), $cap=230$ (°))

Avion A ($x_0=20$ (Nm), $y_0=0$ (Nm), $v=450$ (kt), $cap=90$ (°))

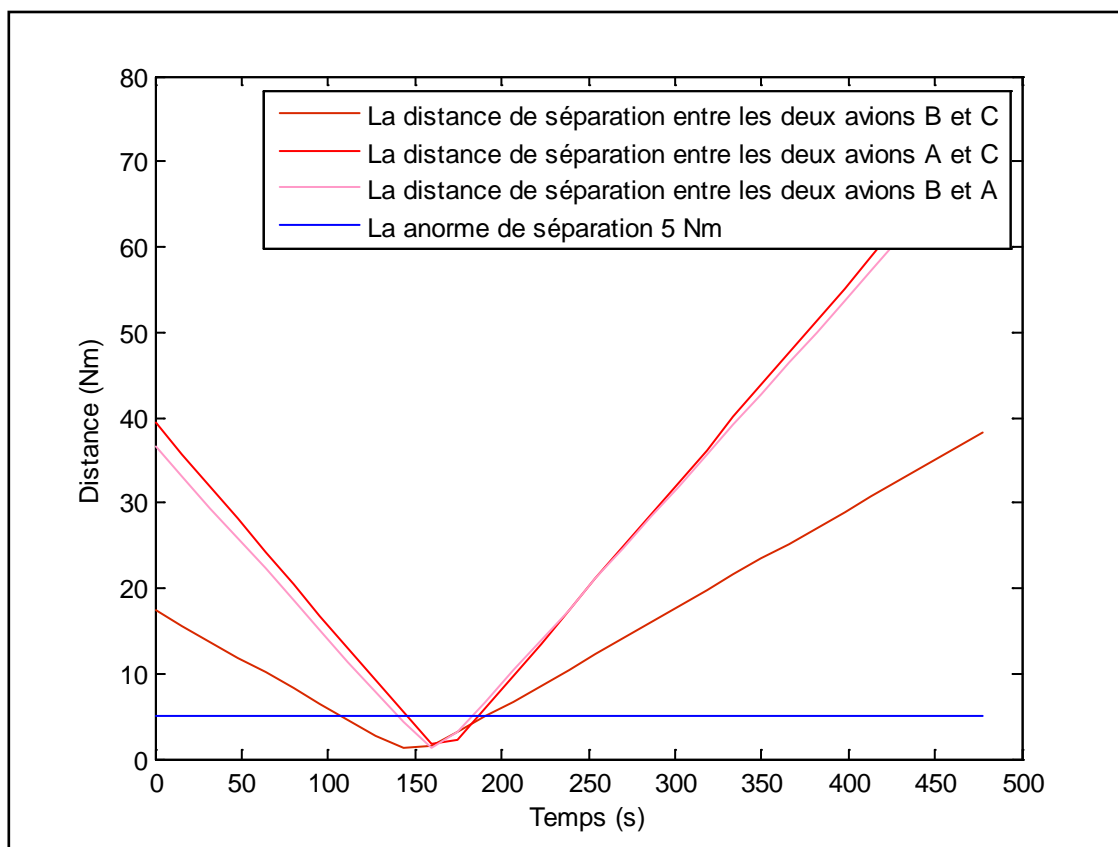


Figure 5-35 : Les distances de séparation entre les paires d'avions « (A,B), (A,C) et (B,C) » avant la résolution par rapport à la norme de séparation « 5 Nm »

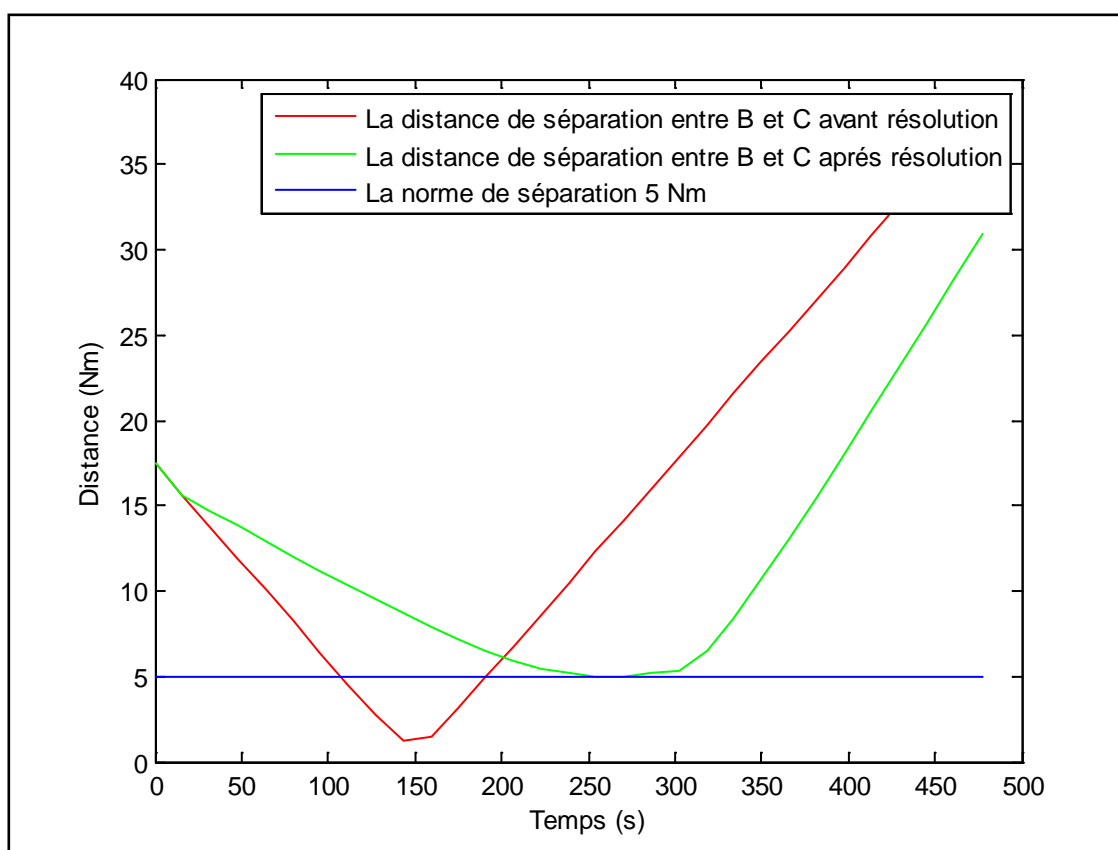


Figure 5-36 : Les distances de séparation entre les deux avions A et C avant et après la résolution par rapport à la norme de séparation « 5 Nm »

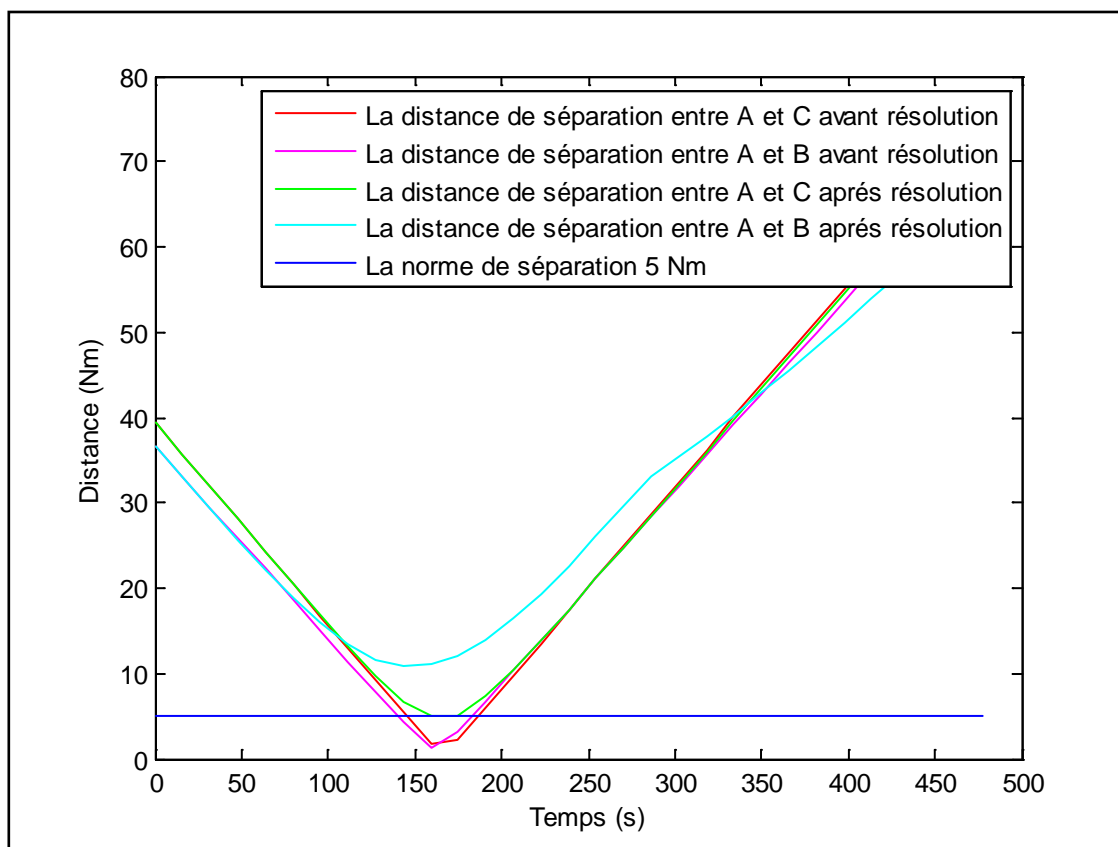


Figure 5-37 : Les distances de séparation entre les paires d'avions « (A,B) et (A,C) » avant et après la résolution par rapport à la norme de séparation « 5 Nm »

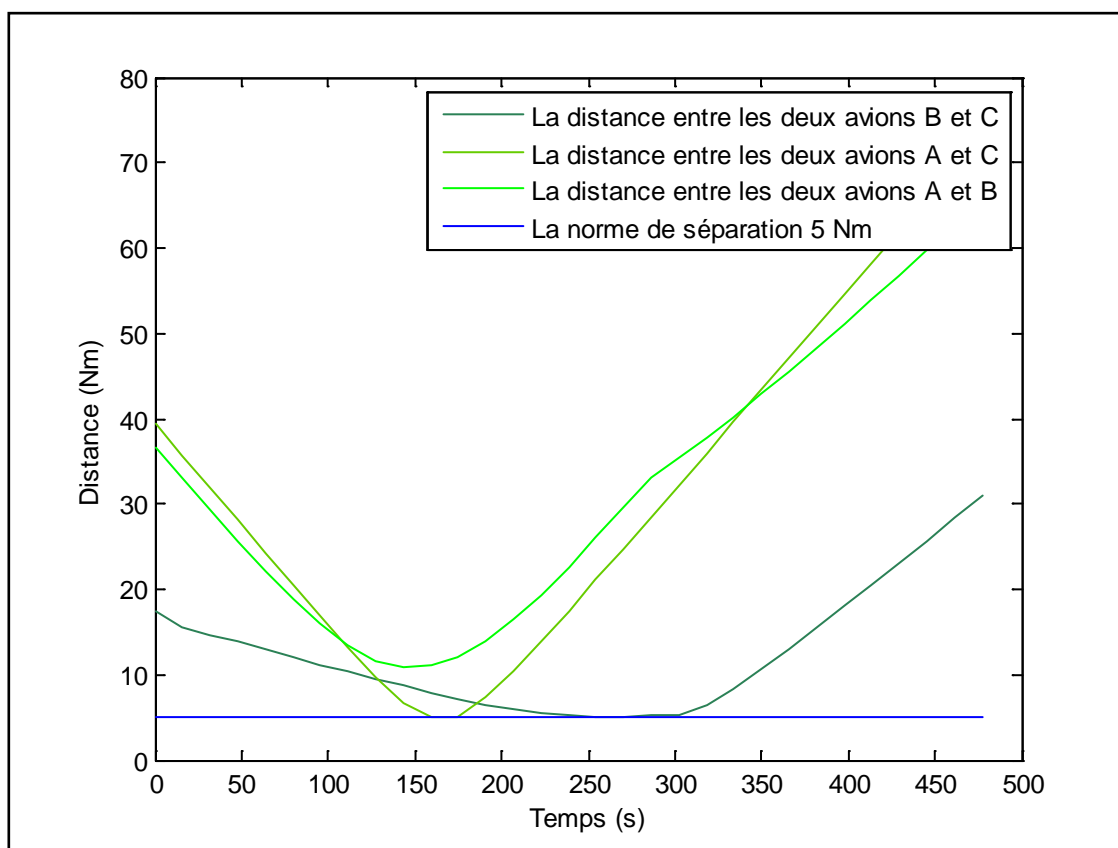


Figure 5-38 : Les distances de séparation entre les paires d'avions « (A,B), (A,C) et (B,C) » après la résolution par rapport à la norme de séparation « 5 Nm »

D'après ces résultats, on constate que le temps de retard varie suivant l'ordre de priorité de la résolution qui a été choisi pour les avions, ainsi une suite naturel de ce travail est d'étudier une technique d'ordonnement automatique pour les avions et faire sa modélisation. La technique d'ordonnement par jetons peut être proposé pour cela.

5.6.2.2. Exemple 1 : $n=4$

■ Données :

- Pas de temps=15.9
- Nombre de pas=30
- Temps total=477

Tableau 5-32 : Données initiales de cas de quatre avions

	x_0 (Nm)	y_0 (Nm)	v (kt)	Cap (°)	x_{fin} (Nm)	y_{fin} (Nm)
Avion A	30	0	400	90	83	0
Avion B	32	11	420	140	67.77	-31.63
Avion C	48	11	420	210	20.17	-37.19
Avion D	59	-6	420	290	6.71	13.03

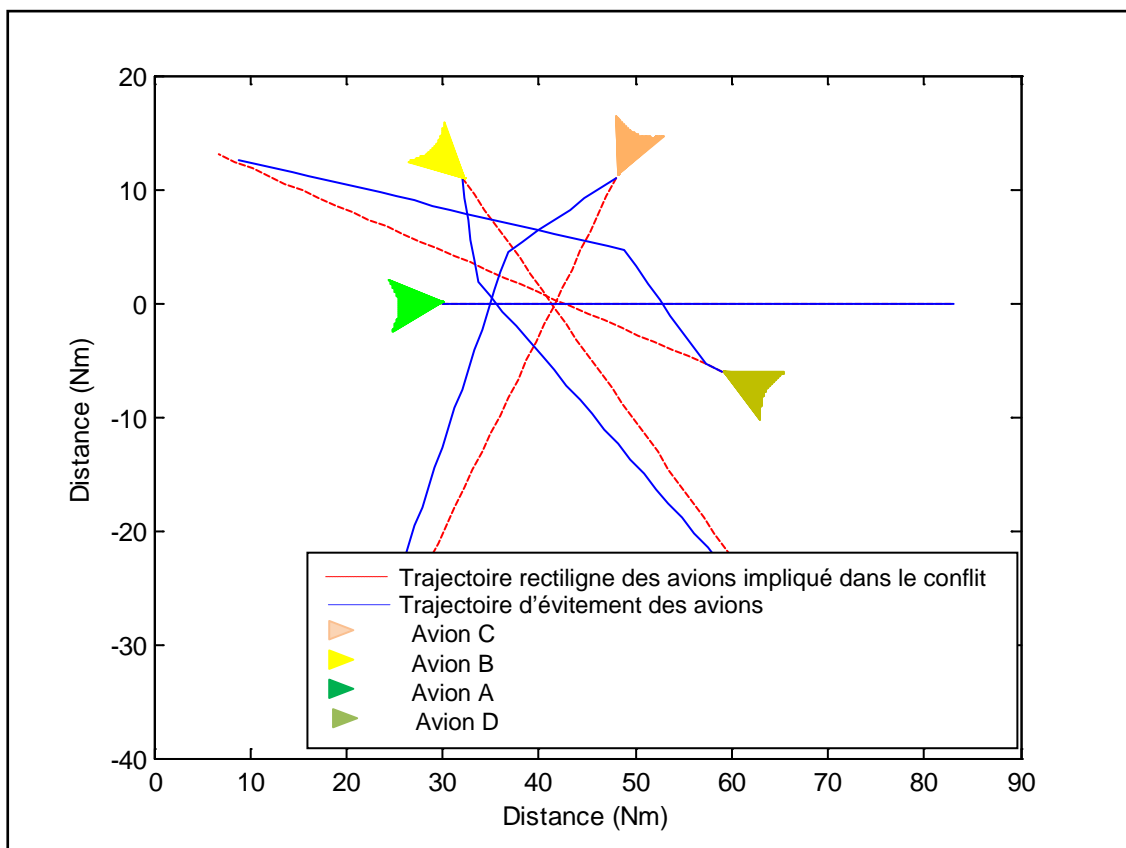


Figure 5-39 : Les trajectoires résolues par l'algorithme A*

Avion A ($x_0=30$ (Nm), $y_0=0$ (Nm), $v=400$ (kt), $cap=90$ (°))

Avion B ($x_0=32$ (Nm), $y_0=11$ (Nm), $v=420$ (kt), $cap=140$ (°))

Avion C ($x_0=48$ (Nm), $y_0=11$ (Nm), $v=420$ (kt), $cap=210$ (°))

Avion D ($x_0=59$ (Nm), $y_0=-6$ (Nm), $v=420$ (kt), $cap=290$ (°))

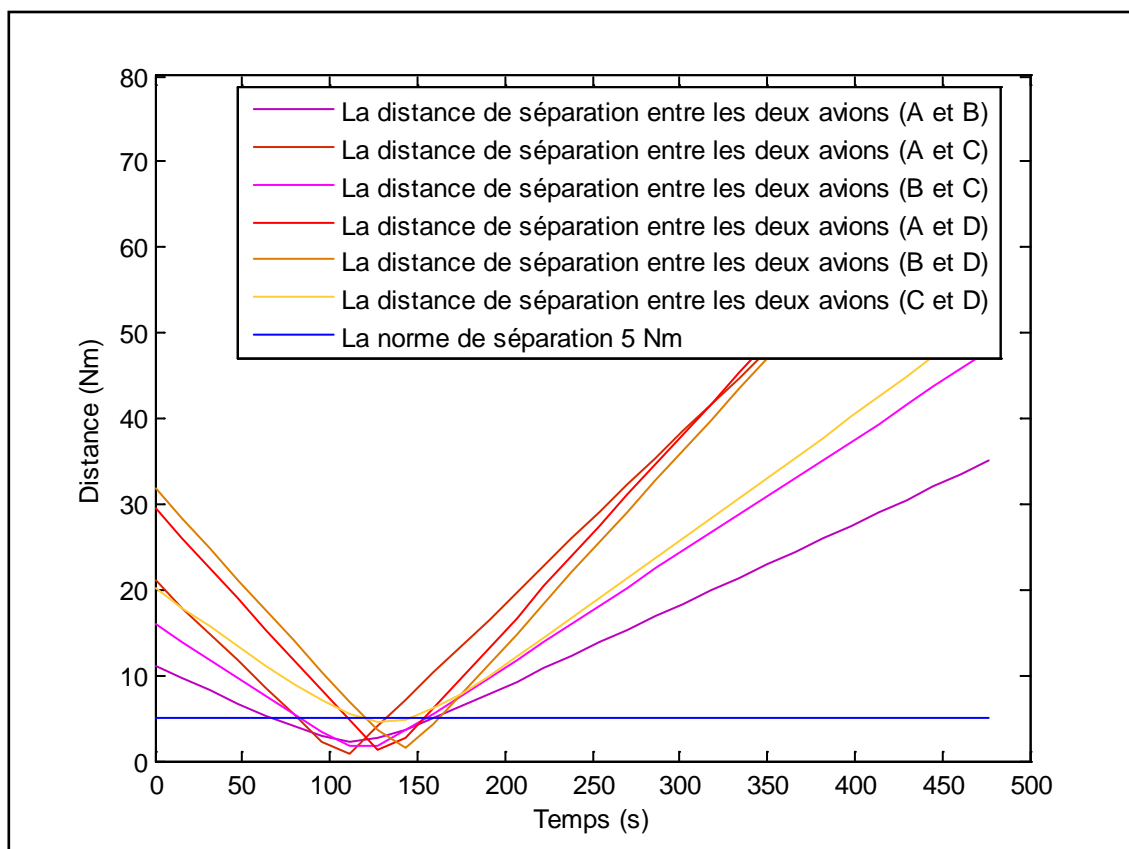


Figure 5-40 : Les distances de séparation entre les paires d'avions « (A,B), (A,C), (B,C), (A,D), (B,D) et (C,D) » avant la résolution par rapport à la norme de séparation « 5 Nm »

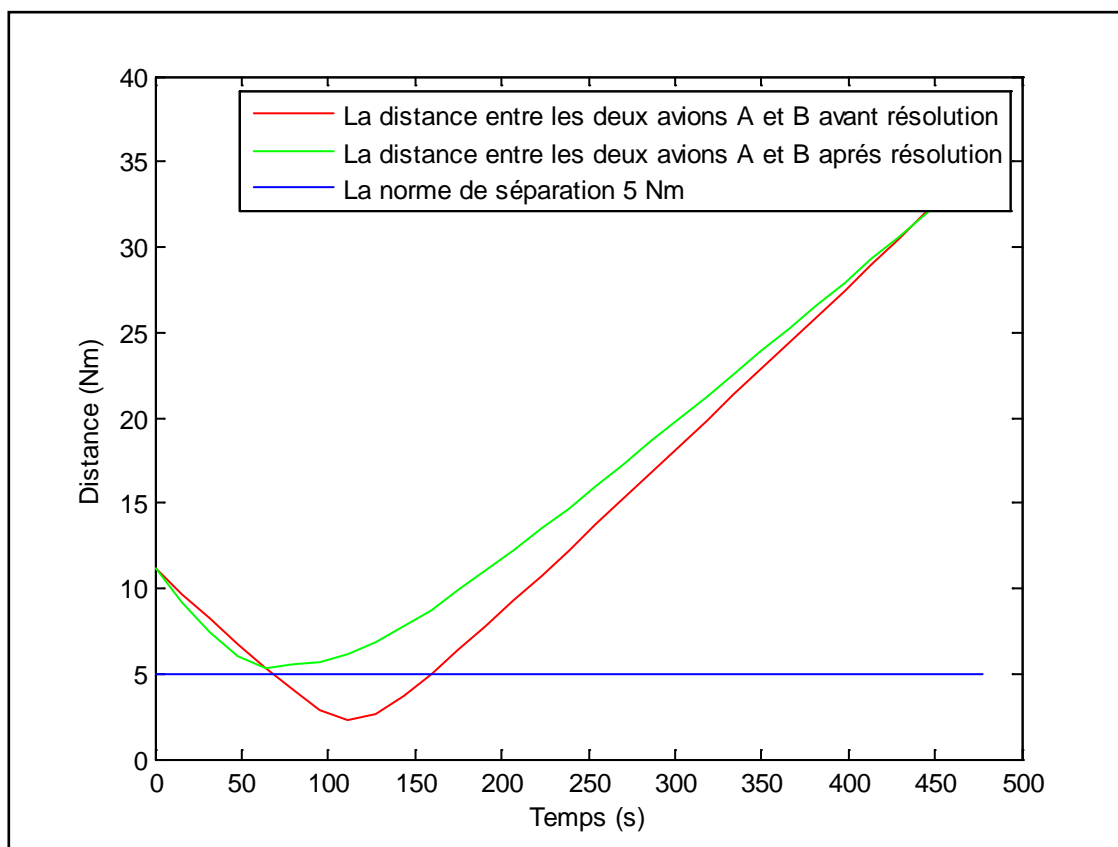


Figure 5-41 : Les distances de séparation entre les deux avions A et C avant et après la résolution par rapport à la norme de séparation « 5 Nm »

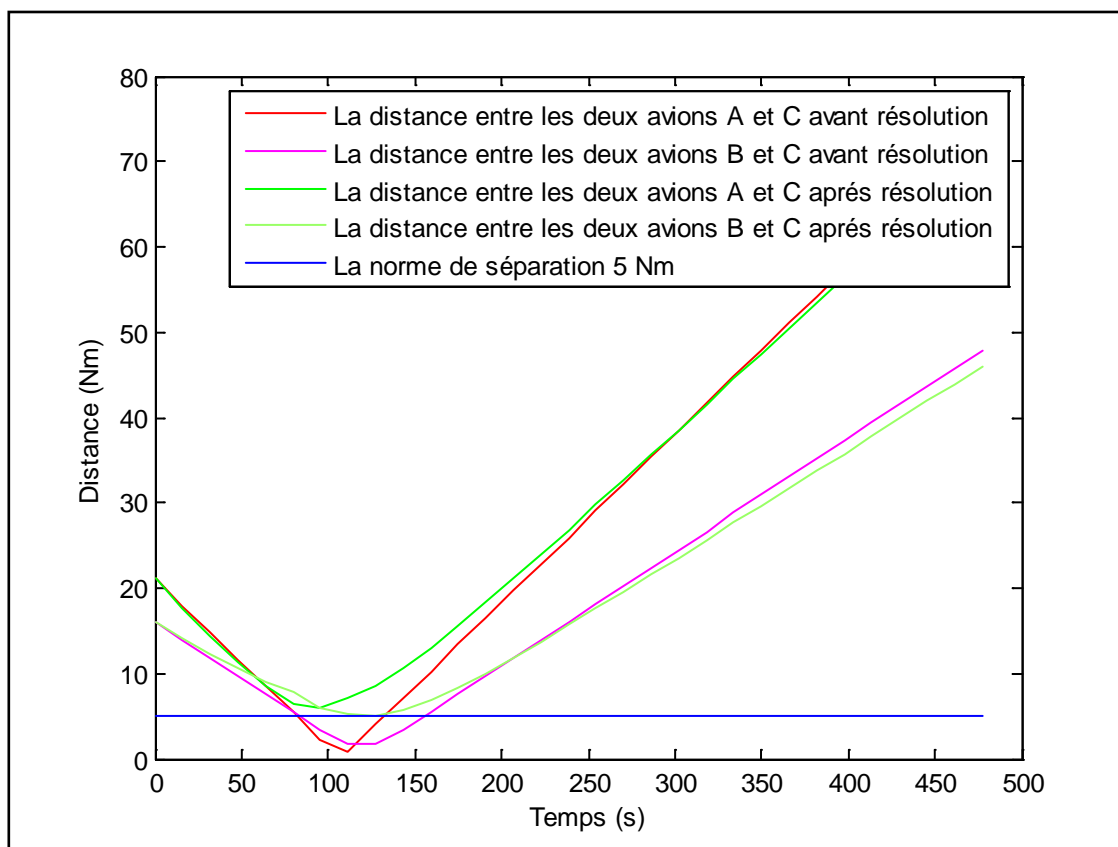


Figure 5-42 : Les distances de séparation entre les paires d'avions « (A,B) et (A,C) » avant et après la résolution par rapport à la norme de séparation « 5 Nm »

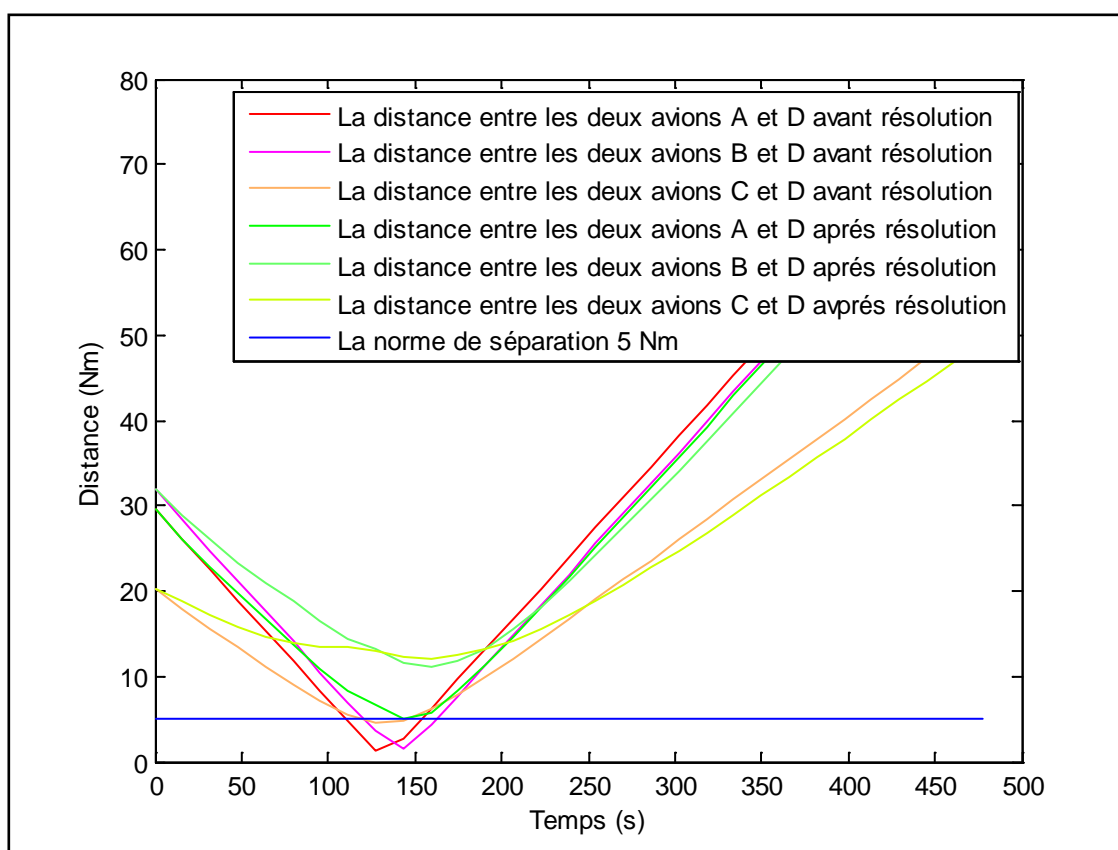


Figure 5-43 : Les distances de séparation entre les paires d'avions « (A,D), (B,D) et (C,D) » avant et après la résolution par rapport à la norme de séparation « 5 Nm »

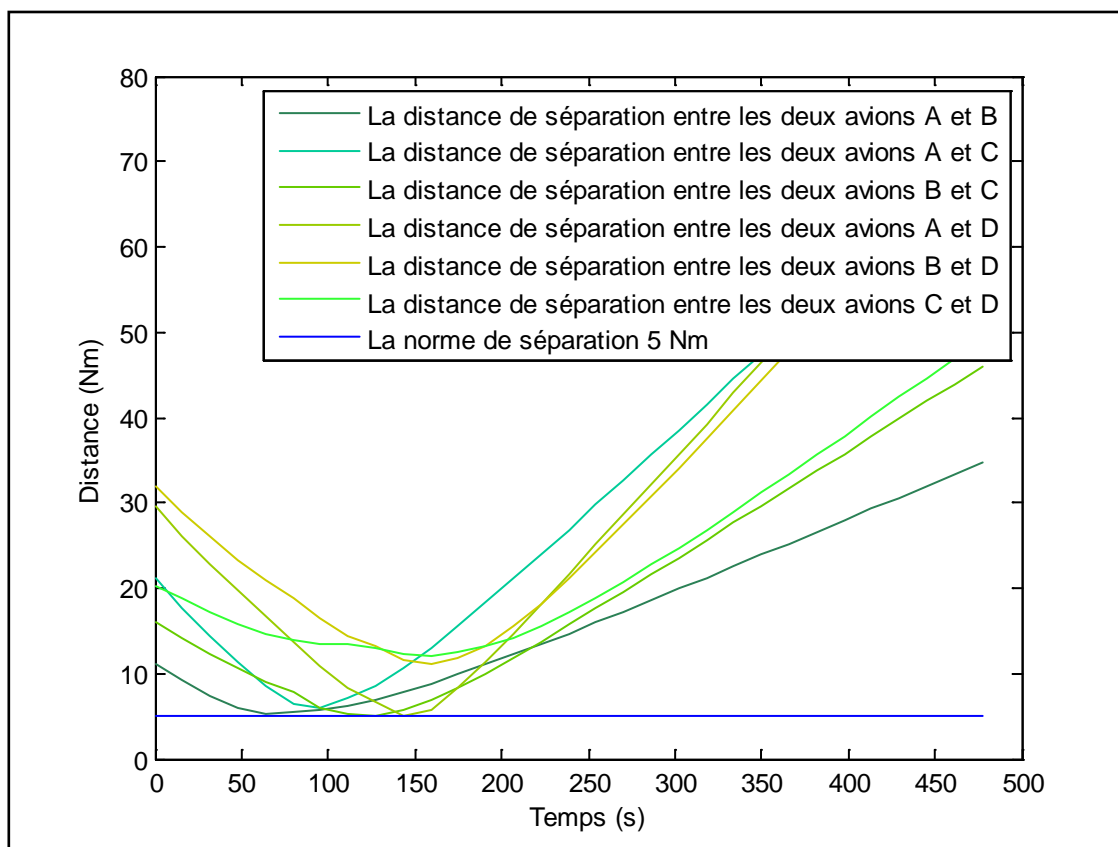


Figure 5-44 : Les distances de séparation entre les paires d'avions « (A,B), (A,C), (B,C), (A,D), (B,D) et (C,D) » après la résolution par rapport à la norme de séparation « 5 Nm »

CONCLUSION

L'étude qui a été réalisée dans ce travail tend à montrer qu'une automatisation est aujourd'hui possible sur un plan purement technique. Les outils disponibles (GPS, FMS de dernière génération, Data-Link, puissance de calculetc), permettraient certainement d'aider à réaliser une automatisation en s'affranchissant à peut près totalement de toute intervention humaine.

En revanche, le problème de la transition vers un tel système est loin d'être résolue. Car ce problème a de multiples facettes : psychologique (un passager accepterait-il de monter dans un avion automatisé), sociale, et également technique (basculement d'un système à un autre qui est de nature radicalement différente).

L'algorithme A* (AStar), peut être utilisé pour optimiser n'importe quelle trajectoire entre un point de départ et un point de destination, en évitant tout obstacle de n'importe quel type (mobile ou fixe), en minimisant (la distance, le temps ou/et la consommation).

D'après les différentes études et simulations qu'on a fait dans ce travail, l'algorithme A* (AStar) peut représenter une technique fiable pour la résolution de conflits, tout en respectant la contrainte du coût minimum (le plus court chemin). Une technique d'ordonnement est toujours nécessaire et indispensable pour la fiabilité de la résolution avec un coût minimum.

L'intégration de ce code dans le système d'avion, peut être donc applicable en ajoutant d'autres contraintes (comme l'ordonnement), et rendra ce dernier autonome pour prendre en charge la détection et la résolution de conflits, avec une consommation minimum de carburant.

APPENDICE A

LISTE DES SYMBOLES ET ABREVIATIONS

ATC	: Contrôle de trafic aérien
ATM	: Gestion de trafic aérien
ATFM	: Gestion de flux de trafic aérien
AWY	: Voie aérienne
ACAS	: Système anticollision embarqué
AIP	: Publication des informations aéronautiques
ASF	: Fonction automatique de séparation
AMPF	: Fonction de gestion de planning de l'espace aérien
AG	: Algorithme génétique
β	: Beta
CENA	: Centre d'étude de la navigation aérienne
CTA	: Région de contrôle
CTR	: Zone de contrôle
cm	: Centimètre
CCR	: Centre de contrôle régional
DEA	: Diplôme d'études approfondis
Eurocontrol	: European Organisation for the Safety of Air Navigation
EFR	: Extended flying rules
E_{opt}	: Espace à optimiser
ft	: Pied

FMS	: Système de gestion de vol
FL	: Niveau de vol
Free-R	: Free route experimental encounter resolution
FAA	: Administration fédéral de l'aviation américaine
FIR	: Région d'information de vol inférieure
f_{opt}	: Fonction à optimiser
GPS	: Système global de positionnement par satellite
GS	: Gentle-strict
IFR	: Règles de vol aux instruments
J	: Jour
kt	: Nœuds
LOG	: Laboratoire global d'optimisation
MOM	: Gestion des options de manœuvres
mn	: Minute
m	: Mètre
Nm	: Mille nautique
OACI	: Organisation de l'aviation civile internationale
p_m	: Probabilité de mutation
p_c	: Probabilité de croisement
RNAV	: Navigation de surface
RVSM	: Réduction des minimums de séparation verticale
TACAN	: Système de navigation aérienne tactique
TMA	: Région de contrôle terminale
TCAS	: Système d'avertissement de trafic et d'évitement d'abordage
TWR	: Tour de contrôle
t_i	: Compteur du temps / $i=1,n$
t_{fin}	: Temps de simulation
UIR	: Région d'information de vol supérieur
UTA	: Région supérieure de contrôle

VFR : Règles de vol à vue
VOR : Radiophare omnidirectionnel

APPENDICE B

On a utilisé aussi le langage Maple, avec un algorithme différent que celui utilisé avec le Matlab, pour comparé les résultats et confirmé l'efficacité de l'algorithme A* (AStar)

En Maple :

- Sans existence d'un obstacle

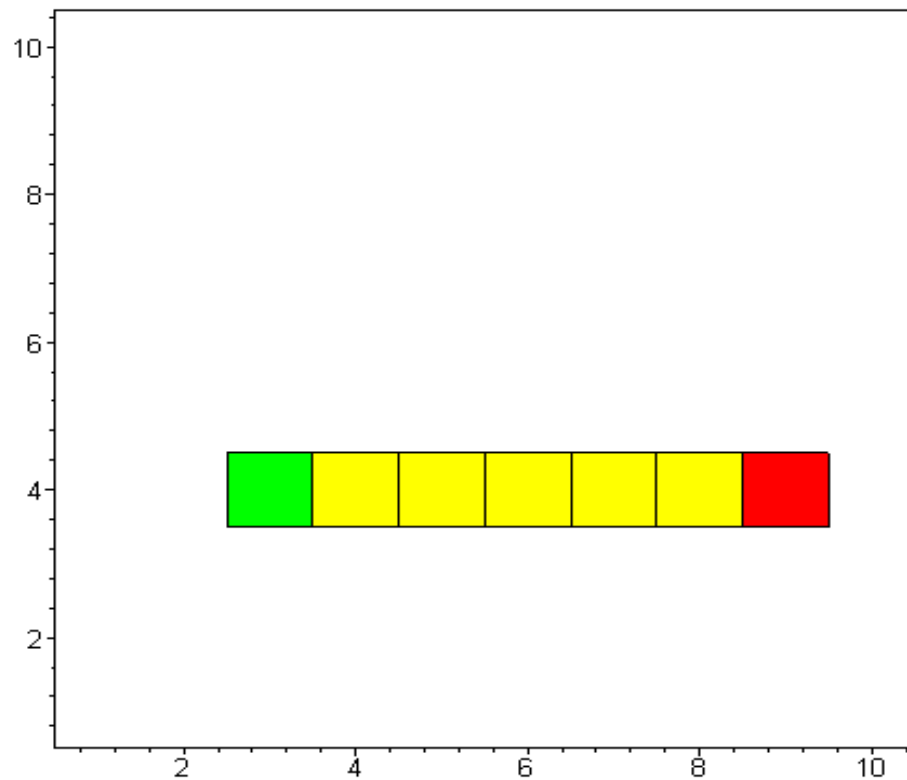


Figure 1: Plus court Chemin trouver en absence d'obstacle

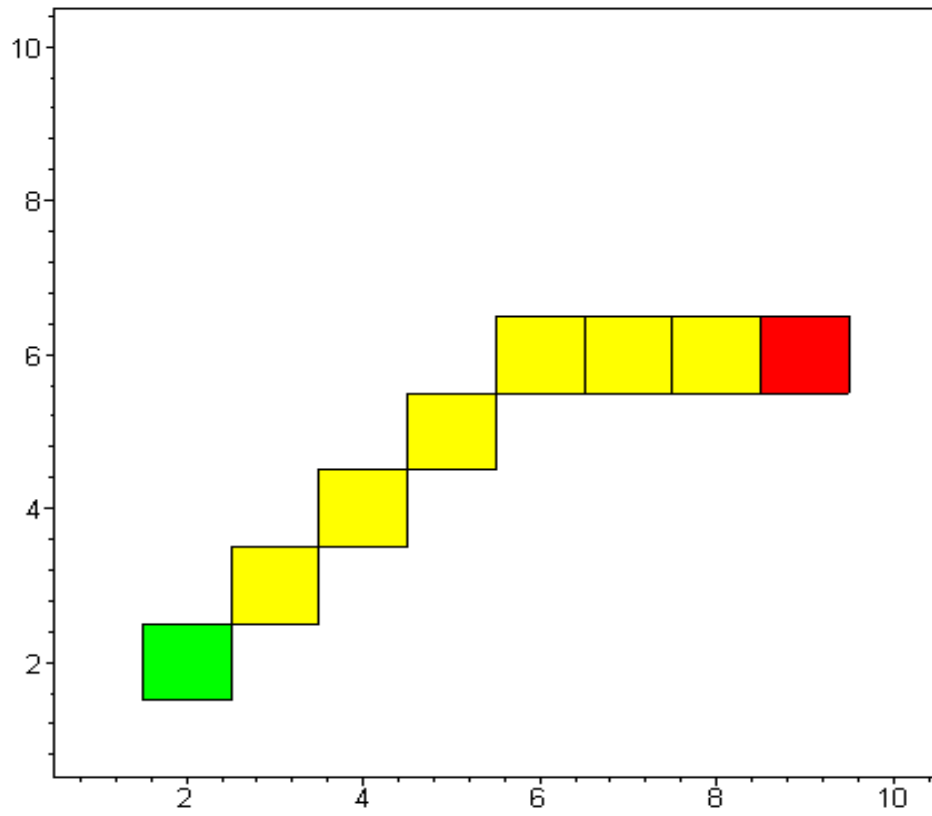


Figure 2 : Plus court Chemin trouver en absence d'obstacle

- Dans l'existence d'un obstacle infranchissable (mur fermé)

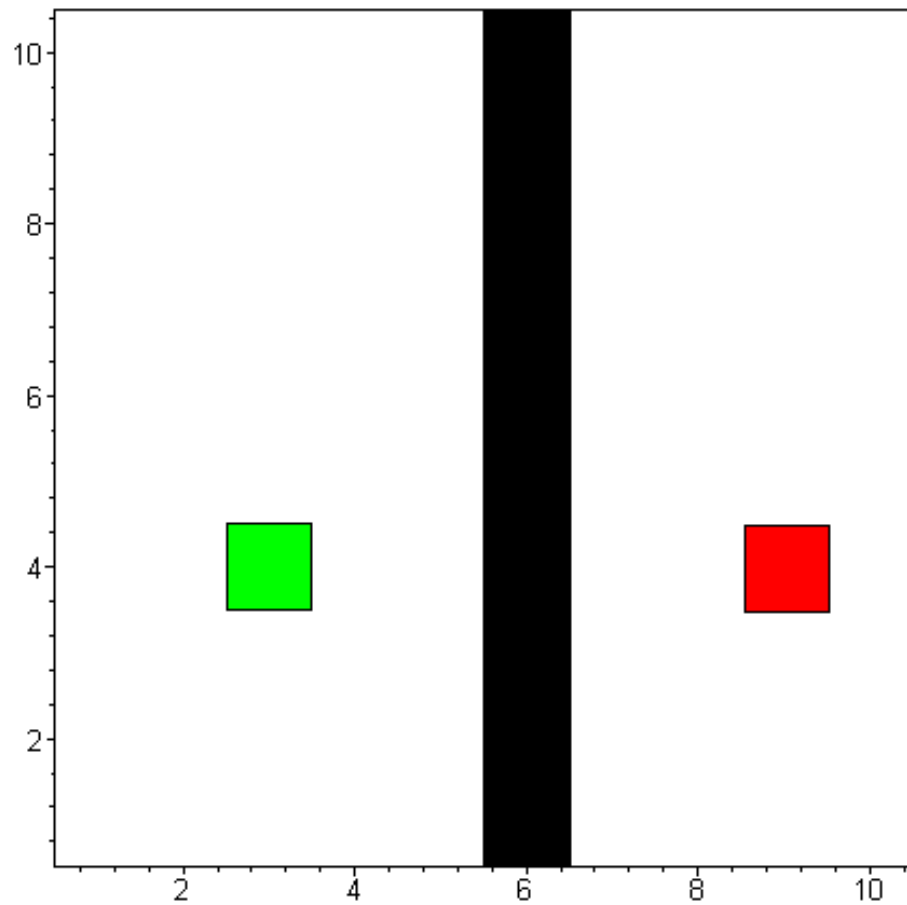


Figure 3 : inexistence du chemin dans le cas d'un mur infranchissable

```

*****", 5, [4, 3], 38, 50, 88
"retour arriere"
"élément fermé", 4, [5, 4]
*****", 4, [5, 4], 52, 30, 82
"retour arriere"
"élément fermé", 3, [4, 4]
*****", 3, [4, 4], 42, 40, 82
"retour arriere"
"élément fermé", 2, [3, 3]
*****", 2, [3, 3], 28, 60, 88
"retour arriere"
"élément fermé", 1, [2, 2]
*****", 1, [2, 2], 14, 80, 94
"retour arriere"
"élément fermé", 0, [1, 1]
*****", 0, [1, 1], 0, 100, 100
"----- NO SOLUTION -----"

```

Figure 4 : pas de solution dans ce cas

- Dans l'existence d'un obstacle quelconque mais pas fermé (mur)

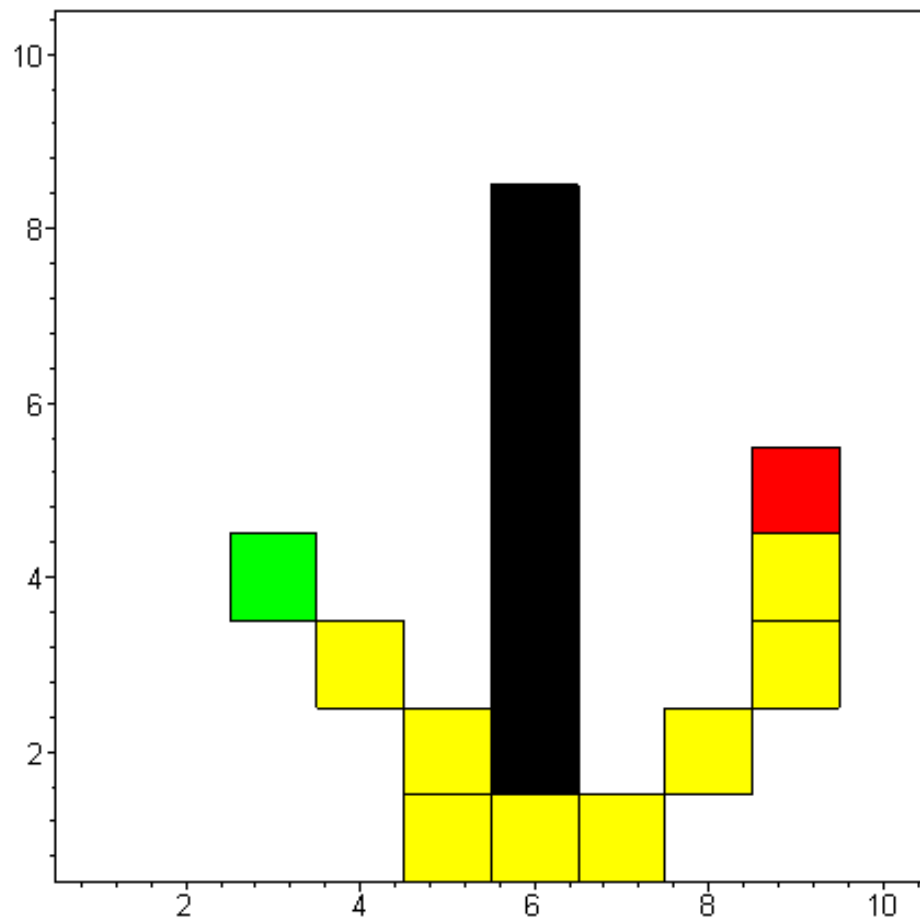


Figure 5 : Plus court chemin trouver dans la présence d'un obstacle

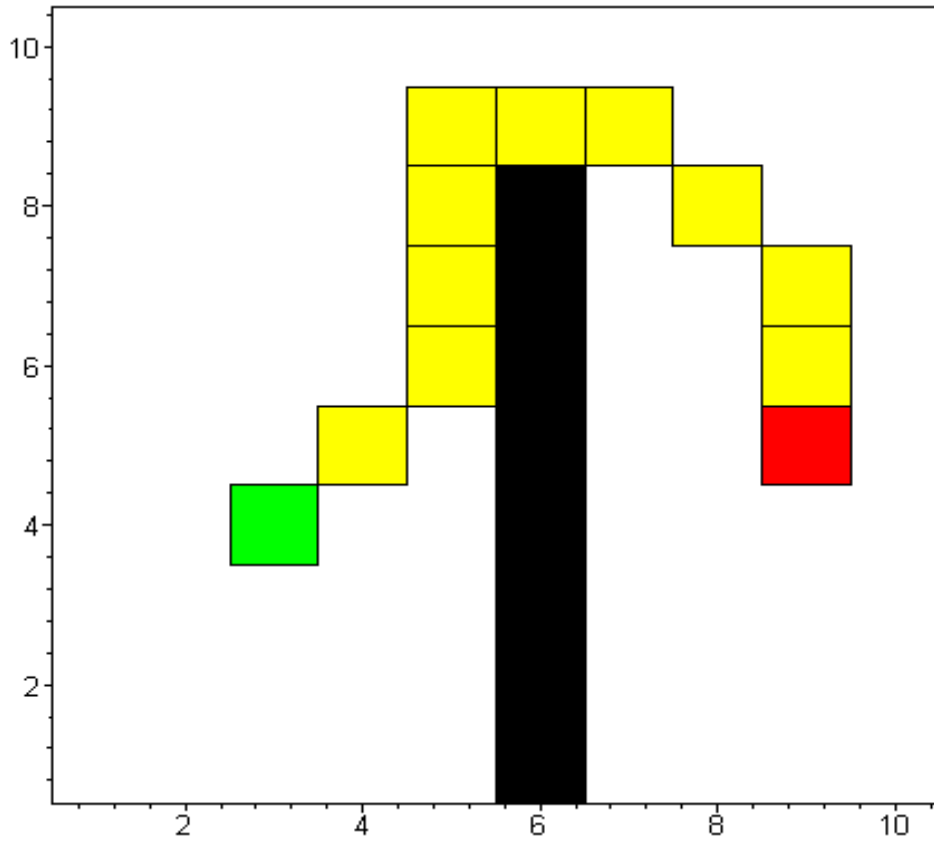


Figure 6 : Plus court chemin trouver dans la présence d'un obstacle

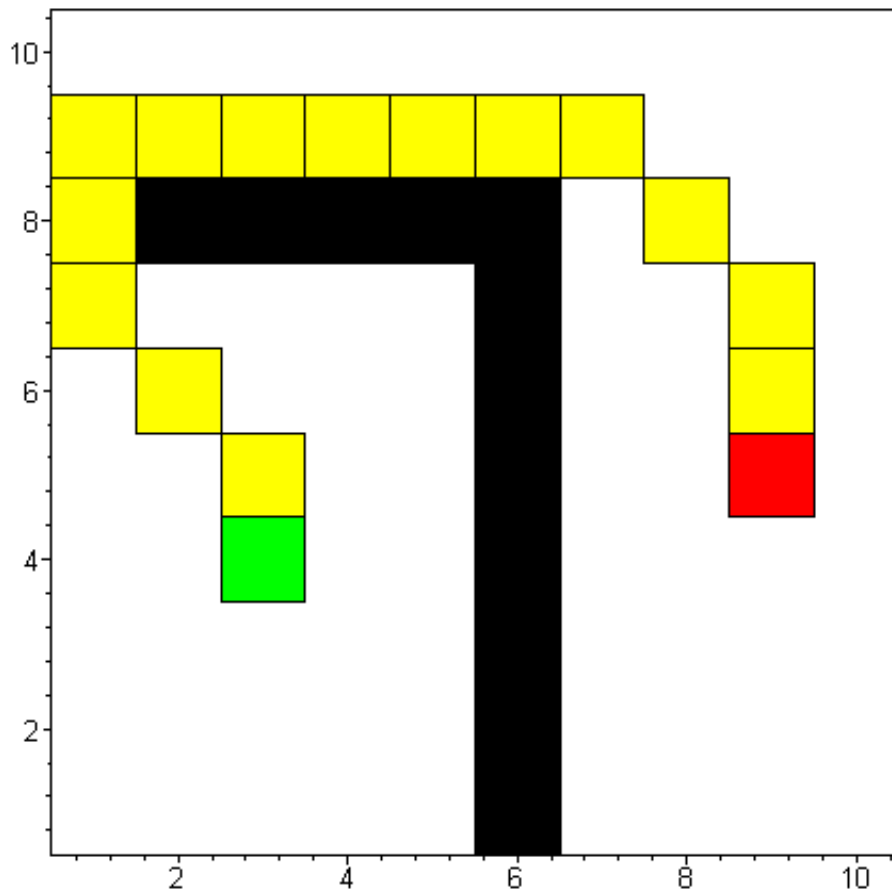


Figure 7 : Plus court chemin trouver dans la présence d'un obstacle

Programme :

```

> restart:
>
>
#####
> # Definition du domaine u0 uf et mur
> MM:=10:NN:=10:
> u0:=[3,4];
> uf:=[9,5];
>
>
>
mur:={ [2,8],[3,8],[4,8],[5,8],[6,1],[6,2],[6,3],[6,4],[6,5],[6,6],
[6,7],[6,8],[6,8],[6,8] };
>
#####
> notinmur:=proc(u)
local rr,v:
rr:=true:
for v in mur do
rr:=rr and is(not(u=v)):
od:
return rr:
end proc:
> indom := proc(u)
is(op(1,u)>=1) and is(op(1,u)<=MM) and is (op(2,u)>=1) and
is(op(2,u)<=NN)
and notinmur(u)
and is(not(u=u0)):
end proc:
>
#####
> # Definition de fonctions
>
> heur := proc(u)
10*(abs(op(1,u)-op(1,uf))+abs(op(2,u)-op(2,uf))):
#sqrt((op(1,u)-op(1,uf))^2+(op(2,u)-op(2,uf))^2):
end proc:
>
> gs := proc(i,j)
if(abs(i)=abs(j)) then 14: else 10: fi:
end proc:
>
#####
> # Point de depart
> OUV:={}:FER:={}:
> k:=0:
> u(k):=u0;
> gg(u(0)):=0:
> ff(u(0)):=heur(u(0)):
>
> k;u(0);gg(u(0));ff(u(0));

```



```

> FER:=FER union {u(k)};
> OUV:=OUV minus {u(k)};
>
> while( not(is(u(k)=uf)) ) do
read "test_.txt":
print("*****",OUV,rr,k,u(k),gg(u(k)),heur(u(k)),ff(u(k)));
if(is(u(k)=u0)) then print("no solution"): break:
fi:
od:

> ##### Dessin
> with(plottools):
with(plots):

vv[0]:=uf:
print(vv[0]):
for i from 1 do vv[i]:=pere(vv[i-1]):
print(vv[i]):
if ( is(vv[i]=u0) ) then break: fi:
od:

l1:=
polygon([u0+[-1/2,-1/2],u0+[1/2,-1/2],u0+[1/2,1/2],u0+[-
1/2,1/2]],color=green),
polygon([uf+[-1/2,-1/2],uf+[1/2,-1/2],uf+[1/2,1/2],uf+[-
1/2,1/2]],color=red):

for v in mur do
l1:=l1,polygon([v+[-1/2,-1/2],v+[1/2,-1/2],v+[1/2,1/2],v+[-
1/2,1/2]],color=black):
od:

for ii from 0 to i do
l1:=l1,polygon([vv[ii]+[-1/2,-1/2],vv[ii]+[1/2,-
1/2],vv[ii]+[1/2,1/2],vv[ii]+[-1/2,1/2]],color=yellow):
od:
display(l1, axes=boxed, view=[1/2..MM+1/2, 1/2..NN+1/2]);

```

Sous programme :

```

FER:=FER union {u(k)}:
print("FER",FER);
print("OUV",k,OUV);
OUV:=OUV minus {u(k)}:
UV:={}:
  for i from -1 to 1 do
    for j from -1 to 1 do
      if ( (not(i=0 and j=0)) and notinmur(u(k)+[i,0]) and notinmur(u(k)+[0,j]) ) then
        gss([i,j]):=gs(i,j):
        UV:=UV union {[i,j]}:
        print("uv",UV);
      fi:
    od:od:
kb:=1:
  for uu in UV do
    v(kb):=u(k)+uu:
    print("v(kb)",v(kb));
    if( indom(v(kb)) and is( not( v(kb) in FER ) ) ) then
      if( is( v(kb) in OUV ) ) then
        re:=gg(u(k))+gss(uu):
        if( re < gg(v(kb)) ) then
          gg(v(kb)):=re:
          ff(v(kb)):=heur(v(kb))+gg(v(kb)):
          pere(v(kb)):=u(k):
        fi:
      else
        gg(v(kb)):=gg(u(k))+gss(uu):
        ff(v(kb)):=heur(v(kb))+gg(v(kb)):
        OUV:= OUV union {v(kb)}:
        pere(v(kb)):=u(k):
      fi:
    print("v(kb),gg(v(kb)),ff(v(kb))",v(kb),gg(v(kb)),ff(v(kb)));
    kb:=kb+1:
    else continue: fi:
  od:
n:=nops(OUV):
print("OUV",OUV,n):
rr:=ff(OUV[1]):
k:=k+1:
u(k):=OUV[1]:

  for bb from 1 to n do
    if ( is( ff(OUV[bb]) < rr ) ) then
      rr:=ff(OUV[bb]):
      u(k):=OUV[bb]:
    fi:
  print("OUV bb",OUV,bb,(heur(OUV[bb]),(ff(OUV[bb]),rr):
od:
OUV:=OUV minus {u(k)}:

```

BIBLIOGRAPHIE

1. J.-M Alliot, T.Schiex, P.Brisset, F.Garcia : « Intelligence artificielle et informatique théorique ». 2ème édition Cépadues mars 2002. ISBN 2.85428.578.6.
2. Frédéric Médioni, « Méthodes d'optimisation pour l'évitement aérien : systèmes centralisés, systèmes embarqués ». Thèse doctorat informatique de l'école polytechnique. 11 décembre 1998.
3. Xavier Olive, « Résolution de conflits par algorithmes stochastiques parallèles ». Mastère à l'école National supérieur de l'aeronautique et de l'espace, Février à juillet 2006.
4. Frédéric Médioni, « Optimisation de la résolution de conflits par algorithmes stochastiques et programmation linéaire ». DEA ENAC, 6 juillet 1994
5. Delahaye Daniel, « Optimisation de la sectorisation de l'espace aérien par algorithmes génétiques ». Thèse de Doctorat, ENSAE, 1995.
6. David Gianazza, « Algorithme évolutionnaire et A* pour la séparation en 3D des flux de trafic aérien ». Journal européen des systèmes automatisés, volume 38 10/2004.
7. Géraud Granger, « Détection et résolution de conflits aériens : modélisations et analyse ». Thèse doctorat informatique de l'Ecole Polytechnique 28 octobre 2002
8. Nicolas Barnier, « Automatisation de la gestion du trafic aérien ». Article laboratoire global d'optimisation Europe, 10 mai 2006.
9. Dodin Pierre, « Résolution de conflits via la programmation semie-définie ». DEA ENAC, Septembre 1999
10. Pierre-Selim Huard, « Résolution de conflits par régulation en vitesse et ordonnancement ». Master sciences de la modélisation, et de l'information et des systèmes, ENAC Février 2006.

11. EUROCONTROL « Stratégie de gestion de la circulation aérienne pour les années 2000 ». Volume 1, édition 2003.
12. Safia Kedad-Sidhoum, « Algorithmique avancée, plus courts chemins : Algorithme A* ». présentation LIP6, 2005-2006.
13. Khayyam, « Recherche de chemin par l'algorithme A* ». (<http://khayyam.developpez.com/articles/algo/astar/>) 23 Aout 2006.
14. Alech Nicolas, « Rapport de stage de 3^{ème} année ». ENSEEIHT, 15 Juin 1994
15. Manuel d'information aéronautique d'Alger, Circulaire d'information aéronautique, Série A n° 01 du 28 Février 2002.
16. Nicolas Barnier, « Application de la programmation par contraintes à des problèmes de gestion du trafic aérien ». Thèse de doctorat de l'INPT soutenue le 6 décembre 2002.
17. Nicolas Durand, « Algorithmes génétiques et autres outils d'optimisation appliqués à la gestion de trafic aérien ». Thèse doctorat informatique INPT, 5 octobre 2004.
18. Nicolas Archambault, Géraud Granger and Nicolas Durand, « Heuristiques d'ordonnement pour une résolution embarquée de conflits aériens par une méthode séquentielle ». publication ENAC 2009.
19. David Gianazza, « Algorithme évolutionnaire et A* pour la séparation en 3D des flux de trafic aérien ». Journal européen des systèmes automatisés année 2005.
20. N. Durand et J.-B. Gotteland, « Algorithmes génétiques appliqués à la gestion du trafic aérien ». Journal sur l'enseignement des sciences et technologies de l'information et des systèmes, Volume 2, Hors-série (7 mars 2003).
21. Géraud Granger, « Résolution de conflits embarquée dans les espaces de faible densité ». DEA ENAC Septembre 1998.
22. Frédéric Koriche, « Colonies de fourmis fourrageuses », DEA ENAC 2007.
23. Hernane Soumeiya et Belkadi Khaled, « Deux stratégies parallèles de l'optimisation par colonie de fourmis ». 4^{ème} conférence international : science de l'électronique, technologie de l'information et de télécommunication a Tunis, 25-29, 2007
24. La lettre de l'académie nationale de l'air et de l'espace, automatisation du transport aérien, n° 48 2006.

25. BENAÏSSA Rachid, « Détection et résolution de conflits aérien », mémoire de magistère, Institut aéronautique, Blida, Septembre 2004.
26. OTMANI Abouelkacem, « Etude et modélisation de la resectorisation de l'espace aérien », PFE Ingénieur, Institut aéronautique, Blida 2005.
27. Nicolas Monmarché, « Algorithmes de fourmis artificielles : applications à la classification et à l'optimisation ». Thèse de doctorat soutenue le 20 décembre 2000.
28. Alliot (Jean-Marc), Bosc (Jean-François), Durand (Nicolas) et Maugis (Lionel). « Cats : a complete air traffic simulator ». Rapport technique, CENA, 1996.
29. Angerand (Luc) et LeJeannic (Hervé), « Bilan du projet SAINTEX ». Rapport technique, CENA, 1992.
30. Alliot (Jean-Marc), « Techniques d'optimisation stochastique appliquées aux problèmes du trafic aérien ». Thèse d'habilitation à diriger des recherches, INPT, mai 1996.
31. Bosc (Jean-François), « Techniques d'évitement réactif et simulation du trafic aérien ». Thèse de Doctorat, INPT, 1997.
32. O. Catoni « Large deviations for Annealing ». Thèse de Doctorat, Université de Paris XI, 1990.
33. Fadda Alessandro. « Etude de problèmes inverses par algorithmes d'évolution et réseaux de neurones ». Thèse de Doctorat, Ecole Polytechnique, 1998.
34. Fron Xavier, Maudry Bernard et Tumelin Jean-Claude « Arc 2000 : Automatic Radar Control ». Rapport technique, Eurocontrol, 1993.
35. Gaudière (Gervais), « Résolutions de conflits aériens par réseaux de neurones ». DEA ENAC, 1995.
36. Y. LeCun « Modèles connexionnistes de l'apprentissage ». Thèse de Doctorat, Université Pierre et Marie Curie, Paris VI, 1987.
37. Niedringhaus W.P, « A mathematical formulation for planning automated aircraft separation for AERA3 ». Rapport technique, FAA, 1989.
38. Zeghal (Karim), « Techniques réactives pour l'évitement ». Rapport technique, ONERA, Juin 1993.
39. Zeghal (Karim), « Vers une théorie de la coordination d'actions. Application à la navigation aérienne ». Thèse de Doctorat, Université Paris VI, 1994.

- 40.L. Angerand and H. LeJeannic, « Bilan du projet SAINTEX ». rapport technique, CENA, 1992.
- 41.N. Archambault, « Optimisation du solveur de conflits du simulateur de trafic cats ». Master Paris VI, 2003.
- 42.N. Durand and J.M. Alliot, « Existing algorithms for collision avoidance, and what the future might hold ». Rapport technique, CENA, 1993.
- 43.V.N. Duong, E. Hoffman, J.P. Nicolaon, L. Flochic, and A. Bossu. « Extended flight rules to apply to the resolution of encounters in autonomous airborne separation ». Rapport technique, Eurocontrol, 1996.
- 44.J. Dréo, A. Pétrowski, P. Siarry, and E. Taillard. « Métaheuristiques pour l'Optimisation Difficile ». Eyrolles, 2003. ISBN : 2-212-11368-4.
- 45.Doc OACI 4444 « procédures pour les services de navigation aérienne ». Quatorzième édition (2001).
- 46.DOC OACI 9426 « Manuel de planification des services de la circulation aérienne ». decembre 1992.
- 47.David GIANAZZA, « Optimisation des flux de trafic aérien ». Thèse d'état, Novembre 2004.
- 48.Thomas Rivière. « Robustesse des solution du problème d'allocation de créneaux ». DEA , juillet 2001.
- 49.<http://www.policymanac.org/aStarTutorial.html>
- 50.<http://theory.Stanford.edu/AstarComparaison.html>
- 51.www.recherche.enac.fr/log
- 52.<http://pom.tls.cena.fr>
- 53.www.recherche.enac.fr/opti/papers/chesis/HABIT