

N° d'ordre : 2151

THÈSE

présentée pour obtenir le titre de

DOCTEUR DE L'INSTITUT NATIONAL POLYTECHNIQUE DE TOULOUSE

Discipline : Informatique

École doctorale : Informatique et Télécommunications

Spécialité : Programmation et Systèmes

par

David GIANAZZA

OPTIMISATION DES FLUX DE TRAFIC AÉRIEN

Soutenance le 2 novembre 2004 devant le jury composé de :

M.	Yves DUTHEN	IRIT	(Président du jury)
M.	Marc SCHOENAUER	INRIA	(Rapporteur)
M.	Patrick SIARRY	Université Paris 12	(Rapporteur)
M.	Jean-Marc ALLIOT	CENA/LOG	(Directeur de thèse)
M.	Arnaud DEDRYVERE	DNA	(Membre du jury)
M.	Alain PRINTEMPS	CENA	(Membre du jury)

Remerciements

Merci tout d'abord à Jean-Marc Alliot et Nicolas Durand, qui ont encadré ma thèse et avant cela mon DEA. C'est grâce à eux que j'ai pris goût à la recherche et à l'informatique. Je les remercie de m'avoir fait une place au sein du laboratoire dans des circonstances particulièrement difficiles de ma vie. Sans leur soutien et leur amitié, rien n'aurait été possible.

Merci aussi à Pascal Brisset, pas seulement pour ses conseils éclairés en Ocaml, mais également pour sa gentillesse, sa disponibilité, sa curiosité d'esprit, et pour les tiramizu, carrot cakes, et autres pâtisseries étranges faites maison (merci à Sylvie également...). Merci à tous les autres membres du LOG ainsi qu'à ceux du LEEA pour l'ambiance conviviale qu'ils savent si bien entretenir au quotidien. Je pense en particulier à Géraud Granger, Nicolas Barnier, Thomas Rivière, Jean-Baptiste Gotteland, Nathalie Lenoir, Christian Bontemps, et Kevin Guittet.

Je remercie Joseph Noailles pour sa gentillesse et son humanité. C'est aussi grâce à ses cours d'optimisation, suivis durant mon DEA, que j'ai repris pied dans la voie scientifique.

Ma gratitude va également à Alain Printemps, chef du CENA, et à Dominique Colin de Verdière, chef-adjoint du CENA, pour l'intérêt qu'ils ont porté à mes travaux et pour m'avoir permis de réaliser cette thèse dans le cadre de ma vie professionnelle. Mon affectation au CENA/LOG a été grandement facilitée par le soutien actif de Frantz Dissler et de Nicolas Dubois. Je ne les en remercierais jamais assez.

Je tiens à remercier vivement MM. Marc Schoenauer et Patrick Siarry d'avoir bien voulu être rapporteurs de cette thèse, et également M. Yves Duthen, président du jury, ainsi que M. Arnaud Dedryvere qui a bien voulu participer au jury, aux côtés de Jean-Marc Alliot et d'Alain Printemps.

J'ai voulu limiter mes remerciements aux personnes rencontrées dans le cadre professionnel. Mais, toute règle ayant ses exceptions, j'adresse un grand merci à Marie-Pierre et à mes filles, Typhaine et Auriane, ainsi qu'à ma famille et à mes amis, notamment pour avoir supporté avec philosophie la période de rédaction de ma thèse.

Je n'ose remercier le hasard, qui fait parfois si mal ou si bien les choses.

Tout ce qui existe dans l'univers est le fruit du hasard et de la nécessité.

(Démocrite / vers 460-370 avant JC)

La vie de l'homme dépend de sa volonté ; sans volonté, elle serait abandonnée au hasard.

(Confucius / 551 à 479 avant JC)

Table des matières

1	Introduction	7
1.1	Problématique	7
1.2	Démarche de recherche	8
1.3	Organisation du document	9
2	Le contexte de la gestion du trafic aérien	11
2.1	Présentation du système existant	11
2.1.1	Généralités	11
2.1.2	La conception actuelle	11
2.1.3	Un système saturé	12
2.2	Les évolutions possibles	12
2.2.1	Les changements conceptuels	12
2.2.2	L'automatisation dans l'avion	13
2.2.3	L'automatisation des systèmes de contrôle aérien	14
2.2.4	Conclusion sur les évolutions du trafic aérien	15
2.3	Choix du contexte et des critères d'optimisation	15
2.3.1	Une question de contexte (s)...	16
2.3.2	... et aussi une question de critères	18
3	Description des algorithmes	19
3.1	Introduction	19
3.2	Justification du choix des méthodes employées	20
3.3	Méthode de classification par nuées dynamiques	21
3.3.1	Généralités sur les méthodes de classification	21
3.3.2	Partitions d'un ensemble à n éléments	21
3.3.3	Les méthodes de partitionnement	22
3.3.4	Description de l'algorithme utilisé	23
3.4	Diagrammes de Voronoï et triangulation de Delaunay	25
3.4.1	Définition d'un diagramme de Voronoï	25
3.4.2	Dualité entre la triangulation de Delaunay et le diagramme de Voronoï	25
3.4.3	L'algorithme de Fortune	26
3.4.4	Implantation	29
3.5	Algorithmes d'exploration d'arbres	32
3.5.1	Séparation & évaluation (<i>Branch & bound</i>)	32
3.5.2	L'algorithme A^*	33
3.6	Algorithmes génétiques	36

3.6.1	Principe et généralités	36
3.6.2	L'algorithme génétique classique	37
3.6.3	Améliorations du principe de sélection	39
3.6.4	Amélioration du principe de croisement	42
3.6.5	Amélioration du principe de mutation	43
3.6.6	Optimisation sous contraintes avec des algorithmes évolutionnaires	43
3.6.7	Une explication intuitive du fonctionnement des AG	44
3.7	Conclusion sur la description des algorithmes	45
4	Optimisation des regroupements de secteurs	49
4.1	Introduction	49
4.2	Description du problème	49
4.3	Modélisation	50
4.4	Difficulté du problème	51
4.5	Algorithmes déterministes	53
4.6	Algorithme génétique	54
4.7	Résultats	55
4.7.1	Comparaison des algorithmes	57
4.7.2	Comparaison entre schéma optimal et schéma déposé	58
4.8	Etude statistique sommaire des dégroupements de secteurs	60
4.8.1	Démarche	60
4.8.2	Valeurs des variables au moment du dégroupement	61
4.8.3	Comparaison des valeurs avant et après dégroupement	64
4.9	Conclusion sur l'optimisation des regroupements de secteurs	67
5	Construction d'un réseau de routes	69
5.1	Introduction	69
5.2	Agrégation des points de croisement	70
5.3	Construction d'un réseau par triangulation de Delaunay	74
5.4	Considérations sur la sectorisation associée au réseau de routes	75
5.5	Plus court chemin à travers le réseau de routes	76
5.6	Conclusion sur la construction du réseau de routes	80
6	Allocation de niveaux de croisière, et séparation des segments évolutifs	85
6.1	Introduction	85
6.2	Allocation de niveaux de croisière, par une méthode de Coloration de Graphe	86
6.2.1	Description succincte de l'algorithme de N. Barnier et P. Brisset	86
6.2.2	Application aux flux dans l'espace aérien français	88
6.2.3	Validation par simulation de trafic	89
6.3	Séparation des trajectoires 3D, avec un algorithme A^*	92
6.3.1	Hypothèses de départ	92
6.3.2	Détection des interférences entre trajectoires	94
6.3.3	Description de l'algorithme	94
6.3.4	Résultats	97
6.4	Conclusion sur l'allocation des niveaux et la séparation des segments évolutifs	98

7	Étude d'un problème simplifié de séparation de flux	101
7.1	Introduction	101
7.2	Modélisation des trajectoires dans l'espace	101
7.2.1	Trajectoire préférentielle associée à un flux	101
7.2.2	Déviations latérales et verticales	102
7.2.3	Modélisation d'une trajectoire	103
7.2.4	Coût associé à une trajectoire	103
7.3	Modélisation des interférences entre trajectoires	104
7.4	Description du problème, stratégies possibles	105
7.5	Difficulté des problèmes, choix des algorithmes	106
7.6	Résolution du problème <i>1 contre n</i> par un algorithme A^*	107
7.6.1	Mise en oeuvre de l'algorithme A^*	107
7.6.2	Description du parcours de l'arbre	107
7.6.3	Coût de transition entre états	108
7.6.4	Choix de l'heuristique	109
7.6.5	Résultats de l' A^*	110
7.7	Résolution du problème global par un algorithme génétique	114
7.7.1	Représentation du problème	114
7.7.2	Choix de la population initiale	114
7.7.3	Le critère d'adaptation	115
7.7.4	L'opérateur de croisement	115
7.7.5	L'opérateur de mutation	116
7.7.6	Résultats de l'algorithme génétique	116
7.8	Conclusion sur l'étude du problème simplifié	119
8	Définition d'un réseau statique de trajectoires 3D séparées, en France et en Europe	121
8.1	Introduction	121
8.2	Définition des flux origine-destination	122
8.3	Analyse des flux origine-destination	124
8.3.1	Analyse des flux en Europe	124
8.3.2	Analyse des flux en France	130
8.3.3	Conséquences sur la problématique	132
8.4	Modélisation des flux selon les évolutions verticales	134
8.5	Modélisation des trajectoires	138
8.5.1	Adaptation du modèle aux cas des flux entrants et sortants	138
8.5.2	Options dans le choix des routes	139
8.6	Prise en compte des performances réelles des avions	139
8.7	Détection des interférences entre trajectoires	142
8.7.1	Inconvénients de la détection avec la distance d	142
8.7.2	Détection par intersection de tubes	142
8.7.3	Interférences autour des aéroports ou dans les phases évolutives	144
8.7.4	Résumé des différents modes de détection	145
8.8	Adaptation des algorithmes au problème réel	145
8.8.1	Adaptation de l' A^*	146
8.8.2	Adaptation de l'algorithme génétique	148
8.9	Commentaires préalables à la présentation des résultats	150
8.9.1	La démarche adoptée	150

8.9.2	Choix des paramètres	151
8.9.3	Les configurations de référence	151
8.9.4	Description des configurations de test	152
8.10	Résultats avec les données France, au-dessus du FL145	152
8.10.1	FR-JUNE21-UNIC-A*-20-PAR	152
8.10.2	FR-JUNE21-UNIC-A*-10-PAR	159
8.10.3	FR-JUNE21-UNIC-GA-20-PAR	160
8.10.4	FR-JUNE21-UNIC-A*-20-STD	161
8.10.5	FR-JUNE21-PROX-A*-15-PAR	162
8.10.6	FR-JUNE21-MULTI-A*-10-PAR	165
8.11	Résultats sur l'Europe, au-dessus du FL145	166
8.11.1	EU-JUNE21-UNIC-A*-20-PAR	166
8.11.2	EU-JUNE21-UNIC-GA-20-PAR	169
8.11.3	EU-JUNE21-MULTI-A*-15-PAR	170
8.12	Conclusion sur l'allocation de trajectoires 3D	170
9	Planification de trajectoires 4D non conflictuelles pour des trains d'avions.	173
9.1	Introduction	173
9.2	Définition des trains d'avions	173
9.3	Prise en compte de la dimension temporelle dans le modèle	174
9.3.1	Trajectoires 4D	174
9.3.2	Détection d'interférences en 4D	174
9.4	Choix de l'algorithme	175
9.5	Résultats pour les données France	175
9.6	Résultats pour les données Europe	178
9.7	Conclusion sur la planification des trajectoires 4D	178
10	Conclusion et perspectives	183
10.1	Contributions	183
10.2	Perspectives de recherche	185
10.3	Perspectives d'utilisation opérationnelle	186
	Glossaire	187
	Bibliographie	191

Chapitre 1

Introduction

1.1 Problématique

Le trafic aérien est depuis longtemps un secteur globalement en forte croissance. Face à un trafic qui double tous les dix ans, les services chargés du contrôle et de la régulation de la circulation aérienne ont dû trouver des solutions pour faire face à cette augmentation : recrutement de personnels supplémentaires, restructurations de l'espace aérien, innovations techniques accompagnées de nouvelles méthodes de travail augmentant la productivité des contrôleurs. Ces mesures et ces innovations ont permis d'absorber toujours plus de trafic tout en améliorant ou en maintenant un niveau de sécurité élevé. Citons pour illustration de ces évolutions l'utilisation du radar, l'automatisation des coordinations inter-centres, l'introduction d'interfaces homme-machines de plus en plus évoluées, le découpage de l'espace en secteurs plus petits, la définition «d'autoroutes aériennes» (réseau ARN).

On constate toutefois depuis quelques années des difficultés grandissantes à accroître de façon significative la capacité des systèmes de gestion du trafic aérien. De faibles variations du volume de trafic induisent dorénavant de très fortes variations dans les retards au décollage (cf rapports Eurocontrol [Com00], [Com01]). Ces dernières années, l'accroissement exponentiel des retards au décollage montre que nous sommes face à un «mur de la capacité» dont les causes ne sont pas conjoncturelles mais structurelles. Il faut de plus en plus de moyens pour gagner des marges de capacité de plus en plus faibles.

Ces constatations ont amené à imaginer et explorer, avec plus ou moins de succès, plusieurs voies de recherche pour sortir de cette impasse :

- améliorer le processus de régulation du trafic,
- décharger à des degrés divers le contrôleur aérien de certaines de ses tâches en les automatisant,
- automatiser totalement le système,
- déléguer aux avions la détection et la résolution des conflits de trajectoire (*free flight*),
- tenter d'imaginer de nouvelles façons de contrôler (*super-sector, sector-less*), ou plus généralement de gérer le trafic (*free flight, free route*),
- optimiser la structure et la gestion de l'espace aérien européen, ainsi que l'organisation des flux de trafic à travers cet espace.

Ces tentatives visent deux objectifs fondamentaux, qui ne sont pas nécessairement exclusifs : le premier consiste à adapter au mieux la demande de trafic aux limitations du système existant, le second consiste à l'inverse à modifier le système afin d'augmenter sa capacité à traiter du trafic, tout en restant dans des conditions de sécurité équivalentes.

Le travail présenté dans cette thèse est centré sur ce dernier point : on cherchera à améliorer

l'écoulement des flux aériens sans réguler la demande de trafic. Au-delà de cet objectif principal, nous serons amenés à nous poser plusieurs questions, et notamment comment optimiser la structure de l'espace aérien, comment organiser les flux de trafic dans cet espace, et peut-on éventuellement optimiser la gestion de ce système sur certains points ?

1.2 Démarche de recherche

Commençons d'abord par placer notre démarche dans son contexte, avant d'en présenter les étapes concrètes. Un certain nombre de travaux ont déjà été effectués, sur la sectorisation de l'espace aérien (citons par exemple [DASF95], au LOG, [VM01], [TBD02]), la construction de réseaux de routes aériennes ([Meh00]), l'allocation de niveaux de vol en croisière ([Let98], et plus récemment au LOG [BB02]), ou encore la définition de routes directes séparées ([MGZK98]). Bien que très intéressants, ces travaux restent malgré tout parcellaires, n'envisageant le problème que sous un seul angle : par exemple uniquement dans le plan, pour le réseau de routes, ou en considérant seulement les phases de croisière, pour l'allocation de niveaux. Et, hormis l'étude TOSCA [MGZK98], dont nous discuterons au chapitre 8, et qui fait un réel effort de validation, les résultats obtenus ne sont pas validés en termes de capacité à traiter plus de trafic en toute sécurité.

Par ailleurs, on voit régulièrement émerger de nouveaux concepts de gestion du trafic – *free flight* ([Fre95b], [fre95a]), *free route* ([Fre01], [Fre02]), *sector-less* ([DGNS01]), *super-sector* ([GNG⁺03]) – que nous présenterons plus en détail dans le chapitre suivant, et dont les promoteurs espèrent qu'ils seront plus performants que le système actuel. Ces concepts sont encore embryonnaires, pour les plus récents, et également plus ou moins bien validés quant au gain attendu.

Les précédents travaux du Laboratoire d'Optimisation Globale (LOG) portaient sur la prévision de trajectoires ([Fab99]) par réseaux de neurones, et sur la résolution automatisée de conflits aériens ([Dur96], [Bos97], [Méd98], [Gra02]), dans le contexte d'un système centralisé ou distribué, par diverses méthodes, déterministes ou stochastiques. La résolution de conflits au roulage a également été abordée ([GD03], [Got04]). Parmi les autres travaux du laboratoire, on trouve également la sectorisation de l'espace aérien par algorithmes génétiques ([Del95], [DASF95]), ainsi que l'application de la programmation par contraintes à quelques problèmes de gestion du trafic aérien ([Bar02]), dont principalement la régulation de la demande, mais aussi l'allocation de niveaux de croisière, plus proche de notre problématique.

L'optimisation des flux de trafic aérien, telle que nous l'envisageons dans cette thèse, et en particulier sans réguler la demande, est donc une voie de recherche encore peu explorée. Telle que nous venons de l'exprimer, notre problématique reste très générale. Concrètement, une démarche en plusieurs étapes a été adoptée.

Dans un premier temps, nous tenterons d'améliorer l'écoulement du trafic sans modifier la structure de l'espace aérien (routes et sectorisation). Comme on s'interdit également de modifier le trafic, et en particulier les heures de décollage, les routes suivies, ou les niveaux de vol, il ne reste plus qu'une seule possibilité : regrouper les secteurs élémentaires sur les positions de contrôle de manière à équilibrer au mieux la charge de trafic.

Dans un deuxième temps, nous chercherons à optimiser les flux origine-destination en jouant sur le réseau de routes.

Comme cette approche bidimensionnelle n'exploite pas la possibilité de séparer les flux aériens dans le plan vertical, nous évaluerons dans un troisième temps l'intérêt d'allouer des niveaux de croisière séparés aux principaux flux.

L'allocation de niveaux ne tenant pas compte des phases évolutives des vols (montée ou descente),

nous chercherons dans un quatrième temps à définir des trajectoires 3D complètes, séparées les unes des autres, pour les principaux flux de trafic. Ceci en tenant compte de la disparité des performances des avions sur chaque flux.

Enfin, en dernier lieu, nous tiendrons compte de la dimension temporelle et chercherons à planifier des trajectoires 4D non-conflictuelles pour des trains d'avions.

1.3 Organisation du document

La thèse est découpée en dix chapitres, le premier étant cette introduction. Le chapitre 2 traite de la gestion du trafic aérien, en présentant tout d'abord le système existant, puis les évolutions possibles, avant de terminer en précisant quelques choix fondamentaux faits dans cette thèse, et par une discussion sur les critères possibles d'optimisation.

Le chapitre 3 décrit les algorithmes utilisés :

- La section 3.3 présente de manière générale les techniques de classification et de partitionnement de nuages de points, et détaille l'algorithme de nuées dynamiques utilisé à plusieurs reprises.
- Dans la section 3.4, on présente l'algorithme de S. Fortune permettant de construire un diagramme de Voronoï, puis d'obtenir une triangulation de Delaunay, à partir d'un nuage de points.
- L'algorithme de Séparation & évaluation (*Branch & bound*), ainsi que l'algorithme A^* sont présentés et illustrés sur un exemple dans la section 3.5.2.
- Enfin, la section 3.6 traite des algorithmes génétiques.

Les trois chapitres suivants explorent différentes pistes possibles pour l'optimisation des flux de trafic aérien :

- Le chapitre 4 traite de l'optimisation des regroupements de secteurs sur les positions de contrôle, par des méthodes déterministes classiques (*Branch & bound*, A^*) ou stochastiques, en l'occurrence un algorithme génétique. L'objectif est d'obtenir des schémas prévisionnels d'ouverture des secteurs qui soient les plus équilibrés possibles tout en respectant les contraintes de capacité des secteurs. Ces schémas prévisionnels sont ceux fournis par les centres de contrôle à l'organisme en charge de la régulation pré-tactique du trafic. Les résultats obtenus sont évalués en comparant sur un cas réel les délais générés avec le schéma optimisé et ceux obtenus avec le schéma réellement déposé le jour choisi.
- Dans le chapitre 5, l'objectif est de construire un réseau de routes aériennes, intelligible et exploitable par des êtres humains, en partant d'une situation initiale vierge : des routes directes entre origine et destination. Le critère retenu est ici la densité des points de croisement du réseau. Une méthode d'agrégation par nuées dynamiques est utilisée pour obtenir un nuage de points (les barycentres des agrégats) respectant certains critères de densité. On construit ensuite un maillage entre les points ainsi obtenus, en appliquant la triangulation de Delaunay. Un mini-secteur est associé à chaque point de croisement du nouveau réseau en utilisant le diagramme de Voronoï associé à ces points. Enfin, un algorithme A^* permet de trouver la route la plus directe entre deux points donnés de ce réseau, tout en respectant des contraintes d'angles entre segments successifs de route. A ce stade, le résultat est un réseau planaire de routes. On ne tient pas compte des évolutions verticales des avions circulant sur le réseau.
- Le chapitre 6 s'intéresse de plus près à la dimension verticale. Dans un premier temps, on y évalue au moyen de simulations de trafic l'intérêt pratique de l'algorithme de coloration de graphes développé par N. Barnier et P. Brisset ([BB02]). Celui-ci permet d'allouer des niveaux de croisière séparés pour les flux d'avions qui se croisent. Cependant, les segments de vol

évolutif (montée, descente) ne sont pas pris en compte avec cette méthode. Dans un deuxième temps, on cherchera donc à séparer également ces phases évolutives, en adaptant l'algorithme A^* utilisé dans le chapitre 5.

Les trois derniers chapitres approfondissent le sujet de l'optimisation stratégique des flux de trafic en 3D, déjà évoqué au chapitre 6, avant d'aborder la planification des trajectoires 4D de trains d'avions, en introduisant la dimension temporelle.

- Le chapitre 7 se focalise sur un problème simplifié de séparation de trajectoires dans l'espace. Après avoir estimé la complexité du problème pour deux stratégies possibles (1 contre n , ou optimisation globale), on y propose deux méthodes de résolution : un algorithme A^* pour la stratégie 1 contre n , et un algorithme génétique pour l'optimisation globale. Ces algorithmes sont testés sur des problèmes jouets, avec une modélisation très simplifiée des profils de vol et de la détection des interférences entre trajectoires.
- Dans le chapitre 8, ces algorithmes sont adaptés aux données réelles afin de produire un réseau statique de trajectoires 3D séparées, pour les principaux flux de trafic en France et en Europe.
- Enfin, on introduit la dimension temporelle dans le chapitre 9, pour d'abord agréger les avions de chaque flux en *trains*, puis planifier une trajectoire 4D non conflictuelle pour chacun de ces trains d'avions.

Le chapitre 10 viendra conclure la thèse et présentera quelques perspectives pour la poursuite des travaux présentés. Le lecteur trouvera en fin de document les références bibliographiques, ainsi qu'un glossaire des termes, acronymes, et abréviations employés dans cette thèse.

Le travail effectué durant la thèse a donné lieu à quatre communications lors de conférences internationales ([GA02], [GAG02], [GD04], [GDA04]) et à deux communications aux colloques de l'Ecole Doctorale Informatique et Télécommunications de Toulouse ([Gia02] et [Gia03])

Chapitre 2

Le contexte de la gestion du trafic aérien

2.1 Présentation du système existant

2.1.1 Généralités

Avant toute chose, précisons ici pour les lecteurs non familiers du monde de la Navigation Aérienne quelques notions utiles à la compréhension du reste de ce document.

On distingue principalement deux types de circulation aérienne civile : l'*approche*, qui désigne la circulation autour des aérodromes, et l'*en-route*, c'est-à-dire la circulation des avions entre la zone d'approche de départ et celle d'arrivée.

Dans les zones d'approche, la circulation est généralement dense. Le travail des contrôleurs aériens consiste à amener les avions entrants de leur point d'entrée dans la zone d'approche jusqu'à la piste, et à guider les avions au décollage jusqu'à leur point de sortie, tout en respectant les cadences maximales d'utilisation de la (ou des) piste(s). La circulation est généralement organisée selon des itinéraires de départ ou d'arrivée, la difficulté du contrôle consistant alors à faire manoeuvrer des avions horizontalement, verticalement, ou en ajustant leur vitesse, dans un espace restreint et encombré, de façon à les mettre en séquence sur leur itinéraire de départ ou d'arrivée.

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 de balises radio-électriques. Le trafic est généralement 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.

2.1.2 La conception actuelle

L'objectif des organismes chargés de gérer la circulation aérienne est d'assurer la sûreté et l'efficacité de l'écoulement du trafic aérien. La gestion du trafic aérien est souvent décrite comme une imbrication de filtres dont le but est d'éviter la collision des avions circulant dans l'espace aérien :

Stratégique : c'est l'organisation à long terme de la structure de l'espace aérien (routes, secteurs, zones militaires, etc) et de l'affectation des flux de trafic sur le réseau de routes.

Pré-régulation : au jour J-2 est défini un schéma de pré-régulation du trafic pour le jour J, en fonction du trafic prévu et d'un schéma prévisionnel d'ouverture des secteurs aériens par les centres de contrôle.

Régulation en temps réel : le jour J, le schéma de régulation est réajusté en fonction du trafic réel et des impondérables du moment.

Contrôle tactique : c'est la phase clé consistant à assurer la séparation entre les avions traversant l'espace. L'horizon temporel du contrôle va de la trentaine de minutes pour la pré-détection d'un conflit présumé, jusqu'à quelques minutes avant le conflit pour la résolution proprement dite.

Anti-collision d'urgence : il s'agit de systèmes embarqués sur l'avion, dont le but est de détecter la présence du trafic environnant et d'en informer le pilote, et en dernier recours de lui fournir des avis d'évitement lorsqu'une collision est anticipée. L'horizon temporel de l'anticollision est d'environ 45 secondes avant la collision présumée.

2.1.3 Un système saturé

Depuis sa création, le secteur du transport aérien est un domaine globalement en forte croissance. Le trafic aérien ne cesse d'augmenter, causant des saturations de plus en plus fréquentes des systèmes de contrôle aérien. Pour parer à ces surcharges, de nombreuses solutions ont été envisagées, dont principalement : augmenter la capacité du système, ou adapter le trafic à la capacité existante.

Augmenter la capacité, c'est mettre en œuvre des moyens permettant de prendre en charge plus d'avions dans l'espace contrôlé. Ce qui veut dire améliorer les méthodes de travail, tirer parti au maximum des technologies de traitement de l'information, mais aussi éventuellement engager plus de personnel, ou encore faire travailler plus les personnels disponibles.

Réguler le trafic en fonction de la capacité disponible, c'est imposer des contraintes aux opérateurs aériens, sur l'heure de départ, et éventuellement sur l'itinéraire et l'altitude de croisière des vols. On peut aussi tenter d'adapter le trafic aux ressources disponibles par des contraintes tarifaires, en augmentant les taxes d'atterrissage aux heures de pointe sur les aéroports saturés, comme à Londres par exemple. Depuis longtemps déjà, les organismes de la circulation aérienne ont mis en place des mesures de régulation du trafic pour éviter les surcharges du système de contrôle. Celles-ci sont plus rapides et plus faciles à mettre en œuvre qu'une augmentation de la capacité, mais atteignent également leur limites.

Les rapports de la *Performance Review Commission* d'Eurocontrol ([Com00], [Com01]) l'indiquent clairement : le système de gestion du trafic, dans sa conception actuelle, est structurellement saturé. A titre d'exemple, une augmentation d'un peu plus de six pour cent du trafic en 1999 s'est traduite par soixante-huit pour cent d'augmentation des retards dus à l'ATFM en-route.

Bien sûr, on peut toujours prévoir à plus ou moins long terme un scénario d'effondrement du trafic aérien mondial, suite à l'épuisement des réserves pétrolières. Mais, en attendant, le développement du trafic mondial, en particulier dans les pays émergents du continent asiatique, ne laisse pas prévoir de fléchissement immédiat. Le système actuel doit donc évoluer en permanence pour pouvoir gérer un trafic en croissance.

Plusieurs évolutions sont envisageables, sur le plan conceptuel comme sur le plan technologique, que nous allons évoquer succinctement.

2.2 Les évolutions possibles

2.2.1 Les changements conceptuels

Un certain nombre de concepts émergents proposent des changements radicaux dans la façon de gérer le trafic aérien.

Dans une conception toute libérale qui nous vient d'outre-atlantique, le *free flight* ([Fre95b], [fre95a]) permettrait aux avions de suivre des routes directes de leur choix dans les espaces à relativement faible densité de trafic. Ce concept peut être envisagé de différentes manières, dont les suivantes :

- en supposant des possibilités de navigation 4D pour le FMS¹ et des liaisons de données sol-bord, un système de contrôle au sol donnerait des instructions de navigation au FMS de l'avion, avec d'éventuelles négociations de trajectoire entre l'avion et le contrôle, afin d'assurer l'anticollision (projet ARC2000 d'Eurocontrol),
- en supposant un équipement embarqué d'anticollision (TCAS/ACAS) sophistiqué et des liaisons de données entre aéronefs, les avions assureraient eux-mêmes leur anticollision en négociant entre eux les trajectoires, de façon automatique.

Ces deux hypothèses sont le reflet de deux "philosophies" qui s'affrontent en matière de sécurité aérienne : celle où les vols sont entièrement sous la responsabilité d'organismes au sol, et celle de l'avion entièrement autonome. N'importe quelle option entre ces deux extrêmes est bien sûr envisageable, et sans doute plus réaliste. Le concept *free-flight* proprement dit, développé initialement aux Etats-Unis, favorise plutôt l'approche distribuée, alors que le concept *free-route* ([Fre01], [Fre02]), proposée par huit états européens, s'oriente vers une approche plus centralisée.

Deux autres concepts plus récents, le *super-sector* et le *sector-less* proposés par le département *Innovative Research* d'Eurocontrol ([DGNS01], [GNG⁺03]), envisagent un contrôle au sol dont les missions seraient complètement redéfinies. Le contrôleur ne gérerait plus des avions à l'intérieur d'un secteur, mais assurerait le suivi d'un groupe d'avions du départ à l'arrivée, ou pendant la traversée d'un secteur de grande taille, en ayant pour mission d'éviter les conflits avec le reste du trafic. Ces concepts, encore à l'étude, impliqueraient une redéfinition complète du réseau de routes aériennes, et il n'est pas encore prouvé qu'ils apportent un gain, dans la capacité à traiter plus de trafic en toute sécurité, par rapport au système actuel.

2.2.2 L'automatisation dans l'avion

En parallèle du débat sur le choix entre un système centralisé, au sol, et un système réparti entre tous les avions, se pose la question fondamentale de l'automatisation. Le train à grande vitesse sans conducteur serait possible sans la résistance psychologique des passagers à l'idée de se déplacer aussi vite dans un engin commandé par une machine. Notons d'ailleurs que le rôle du conducteur se réduit à maintenir les mains sur le volant et à surveiller que tout se passe bien. Le métro sans conducteur existe déjà, ce qui prouve que l'attitude du passager à l'égard de l'automatisme est susceptible d'évoluer. Pourquoi pas, d'ici quelques temps, envisager l'avion sans pilote ?

Outre les impératifs de productivité qui poussent à l'automatisation, les statistiques montrent que l'homme est une des principales causes directes des accidents d'avions. Le débat ne porte donc plus actuellement sur la question de savoir s'il faut ou non automatiser, mais plutôt de quelle manière il faut le faire.

Les progrès technologiques en matière de circulation de l'information permettent d'envisager dans un avenir proche des liaisons de données permanentes entre le sol et l'avion. De plus les automatismes de l'avion permettent maintenant d'atterrir sans que le pilote touche les commandes, dans des conditions météorologiques très dégradées. Il reste encore quelques progrès à faire dans le domaine pour arriver au tout automatique jusqu'à l'arrêt devant l'aérogare. Mais pilotage automatique et liaisons sol-bord performantes autorisent à penser que l'on pourrait voir un jour un système de transport aé-

¹*Flight Management System* : c'est le système de gestion des automatismes de l'avion, dont le pilotage automatique

rien entièrement commandé par les organismes au sol, qui déterminerait les trajectoires à suivre par les pilotes automatiques. D'éventuels pilotes humains ne seraient là que pour surveiller le bon fonctionnement des automatismes, comme c'est déjà souvent le cas actuellement.

Cependant, les équipements embarqués offrent de plus en plus de facilités dans la navigation et le positionnement de l'avion, d'informations sur la météo et sur le trafic environnant, et vont même jusqu'à proposer au pilote des manœuvres d'évitement des collisions. Associés à des systèmes dits "tête haute"² d'aide au pilotage, ces systèmes pourraient permettre à un pilote humain de s'affranchir des informations, voire des instructions, provenant des organismes au sol, pour assurer lui-même et complètement la sécurité de son avion et de ses passagers. Mais la même alternative avec un pilote automatique et un système performant d'anticollision n'est pas non plus impensable.

Concernant l'avion, il semblerait que les tenants du tout-automatique aient aujourd'hui l'avantage. Avec les cockpits des dernières générations d'avions Airbus A320, A330, A340 et Boeing B777, le rôle des pilotes consiste de plus en plus à donner des ordres au pilote automatique à travers un clavier, et à surveiller que les automatismes fonctionnent correctement. Les liaisons de données air-sol, ou les liaisons de données air-air associées aux systèmes anticollision risquent même à courte échéance de dispenser le pilote humain d'avoir à donner ces ordres. Dans le cas de l'avion, la transition entre le pilotage "au manche" et le pilotage via FMS s'est faite brutalement. Si ce nouveau type de pilotage "au clavier" est intéressant, en particulier pour les possibilités de suivre automatiquement des procédures standardisées, il présente néanmoins quelques inconvénients, dont la difficulté pour un pilote humain de "reprenre la main" en cas de besoin. Plusieurs possibilités se présentent pour l'avenir :

- le statu quo, en partant de l'hypothèse que les pilotes humains soient de mieux en mieux formés au pilotage au clavier. Le risque étant que lesdits pilotes ne sachent plus réagir correctement en cas de défaillance des automatismes,
- une évolution vers plus d'automatisation, de façon à supprimer totalement les cas où l'intervention de l'homme s'avère nécessaire, et à limiter au maximum les risques de défaillance des systèmes. Le principal argument militant pour cette évolution est que la cause principale des accidents d'avions est le facteur humain.
- l'intégration, en plus des fonctions existantes ou à venir, d'automatismes qui aident réellement au pilotage humain, en particulier dans les phases délicates de décollage et d'atterrissage, celles où il est le plus crucial qu'un pilote humain puisse intervenir rapidement et à bon escient si besoin est. Les dispositifs "tête haute" semblent à cet égard plus qu'utiles, et permettent de plus aux pilotes de pratiquer un pilotage actif, en bref de ne pas perdre la main. L'argument selon lequel, pour éviter que l'homme ne soit le maillon faible du pilotage, il faut construire des avions qui soient conçus pour être pilotés par des humains, ne manque pas d'intérêt.

2.2.3 L'automatisation des systèmes de contrôle aérien

Concernant les systèmes au sol de gestion du trafic aérien, la problématique est sensiblement la même. L'évolution du trafic laisse à penser qu'il ne sera plus possible d'y faire face par la seule augmentation du nombre de contrôleurs et un découpage de l'espace en plus petits secteurs. Une telle méthode, qui augmente considérablement les coordinations inter-secteurs et les tâches associées, a ses

²Les systèmes HUD (*Head Up Display*) ou "tête haute" sont des aides au pilotage où les informations nécessaires sont projetées directement sur la vitre du cockpit par un système optique. La principale innovation est cependant de proposer un pilotage "à l'énergie totale" d'une précision et d'une simplicité remarquables. Ces systèmes sont principalement développés sur les avions d'armes, et font leur timide apparition dans le civil, surtout outre-atlantique. L'inventeur du système est un ingénieur français, Mr Klopstein, qui dès les années soixante expérimentait ce principe de pilotage avec un collimateur de vol électronique

limites. En conséquence il faudra automatiser certaines tâches du contrôle, de façon à alléger la charge de travail du contrôleur, et éventuellement changer les méthodes de contrôle. Reste à savoir quelles tâches automatiser et comment.

Les tâches du contrôleur “radar”, celui qui donne les instructions aux avions, peuvent grossièrement se répartir de la manière suivante :

- les tâches de surveillance du trafic et de détection des conflits à venir,
- les tâches de résolution des conflits en cours,
- les tâches de coordination, qui consistent à prendre en compte le trafic appelant sur la fréquence radio à l'entrée dans le secteur, ou à donner des instructions de changement de fréquence aux avions sortant du secteur.

Par ailleurs, un ou plusieurs contrôleurs “organiques” prennent en charge les autres tâches liées à la coordination, dont notamment celles de planification, et celles de pré-détection des conflits.

La résolution de conflits est la tâche la moins coûteuse en temps pour le contrôleur et la plus difficile à automatiser. Les autres tâches seront donc sans doute automatisées d'abord, ce qui requiert de disposer des liaisons de données sol-bord.

Quelle que soit l'évolution à venir, la manière la plus sûre d'automatiser les tâches du contrôle semble être d'offrir d'abord au contrôleur humain la possibilité de se décharger de certaines d'entre elles sur la machine. La démarche actuelle est donc de proposer des aides automatiques “intelligentes”, pour la détection et la résolution de conflits, pour la surveillance et la coordination du trafic.

2.2.4 Conclusion sur les évolutions du trafic aérien

Face à ces nombreuses évolutions possibles, tant sur le plan des méthodes de travail que des technologies, il est bien difficile de prévoir à quoi ressemblera la gestion du trafic aérien d'ici une quinzaine d'années.

On ne distingue pas *a priori* de voie qui soit objectivement meilleure qu'une autre, et on ne peut prévoir les choix conceptuels, voire idéologiques, qui influenceront sur les évolutions futures du système.

On peut cependant affirmer que le système actuel dispose en lui-même d'une certaine inertie : il existe et fonctionne malgré tout, avec plus ou moins d'efficacité. La mise en place d'un système alternatif introduisant de nouvelles méthodes de contrôle et/ou une automatisation de certaines tâches devra d'une part savoir gérer la transition à partir du système actuel, et d'autre part démontrer un gain effectif dans la capacité à traiter plus de trafic en toute sécurité.

2.3 Choix du contexte et des critères d'optimisation

La problématique évoquée dans l'introduction de cette thèse peut être abordée selon plusieurs angles, en fonction à la fois du contexte dans lequel on choisit de se placer, et aussi des critères selon lesquels on cherche à améliorer le système. Nous venons de voir dans la section précédente que l'on ne peut pas prévoir avec certitude les évolutions futures des systèmes de gestion du trafic aérien. Nous allons ici, d'une part, préciser quelques hypothèses de base adoptées dans le cadre de cette thèse, et, d'autre part, d'évoquer les différents critères que l'on pourra choisir d'optimiser.

2.3.1 Une question de contexte (s)...

Systeme centralisé ou distribué ?

Imaginons une situation hypothétique où chaque avion vole au niveau qu'il a demandé, en suivant une route directe entre départ et arrivée. Cette situation minimise les coûts individuels de chaque vol et est optimale de ce point de vue, bien qu'elle soit inacceptable sur le plan de la sécurité des vols.

Les projets basés sur le paradigme du *free flight* tablent sur l'utilisation de nouveaux équipements de bord pour, à partir de cette situation, améliorer la sécurité. L'objectif est de permettre à chaque avion d'éviter le reste du trafic en fournissant au pilote une visualisation du trafic environnant, ainsi que des avis ou des ordres de changement de trajectoire, produits par un calculateur à bord. Dans ce contexte de gestion distribuée du trafic, les avions sont censés choisir leur route et il n'est nul besoin d'organiser les flux de trafic, ni de découper l'espace aérien en secteurs de contrôle. Bien que séduisante au premier abord, cette solution est plus complexe qu'il n'y paraît et nécessite, pour pouvoir être mise en oeuvre dans des situations de trafic dense, l'utilisation d'algorithmes distribués permettant de coordonner les choix de trajectoire des avions (cf [Gra02]). En outre, elle n'est pas optimale du point de vue du système global.

Notre première hypothèse consistera donc à pré-supposer l'existence de systèmes centralisés gérant le trafic (typiquement des centres de contrôle aérien).

Quel degré d'automatisation ?

Dès 1982 aux États-Unis et peu après la grande grève des contrôleurs aériens, le projet AERA ([OTA82]) envisageait une automatisation complète des tâches de détection et de résolution de conflits, avec à terme la relégation des contrôleurs au rôle de superviseurs d'automatismes. Du côté européen, des projets comme SAINTEX ou ARC2000 avaient également en ligne de mire l'automatisation de la résolution de conflits. Ces projets aboutirent à des réussites partielles, voire à des échecs (pour AERA), la difficulté du problème ayant été à l'époque sous-estimée. Aujourd'hui, il existe des algorithmes permettant de faire voler (en simulation) le trafic d'une journée complète en résolvant la totalité des conflits. Le lecteur pourra se reporter à la thèse de Géraud Granger ([Gra02], réalisée au LOG) qui décrit un tel simulateur utilisant des méthodes stochastiques de résolution. Dans cette même thèse ainsi que dans le rapport Eurocontrol [KF02], on trouvera une revue extensive de la littérature sur les nombreux projets de résolution automatisée des conflits. Toutefois, si les techniques informatiques semblent arriver à maturité, les problèmes de transition éventuelle entre le système actuel et un système complètement automatisé ne manquent pas.

Selon, les cas, on pourra donc se placer soit dans le contexte actuel, où des êtres humains sont en charge de la gestion du trafic, ou bien dans un contexte plus futuriste avec une automatisation totale ou partielle d'un certain nombre de tâches critiques. L'exemple le plus parlant à cet égard est celui de la construction d'un réseau de routes aériennes. Dans un système futuriste totalement automatisé, rien n'interdit aux avions de choisir par défaut la route la plus directe possible, et notre réseau de routes est alors vite construit. Par contre, si l'on reste dans le contexte actuel, un critère clé du processus de construction est que le réseau doit rester intelligible et exploitable par des humains.

En particulier, il doit permettre le découpage harmonieux de l'espace aérien en secteurs élémentaires. En conséquence, les points de croisement des routes ne doivent pas être trop près les uns des autres, ni trop près d'une frontière d'un secteur. Cependant, comme nous allons le voir dans le point suivant, le critère de l'automatisation n'est pas le seul qui détermine le choix d'un réseau direct ou avec une sectorisation.

Routes directes ou réseau avec sectorisation de l'espace ?

Selon les principes actuels de l'organisation des services de contrôle, l'espace aérien est découpé en secteurs gérés chacun par deux ou trois contrôleurs, et les flux de trafic sont astreints à suivre des routes pré-définies traversant ces secteurs. L'ensemble routes/secteurs actuellement en place est la résultante des modifications successives ayant eu lieu depuis les débuts de la gestion du trafic aérien. Si l'on reste dans le cadre des méthodes de travail actuelles, il faudra donc tenir compte de la nécessité d'un découpage équilibré de l'espace, afin de rester dans chaque secteur à un degré de complexité permettant une gestion sûre du trafic par des contrôleurs humains.

Une alternative plus prospective consiste à faire l'hypothèse d'un changement radical des méthodes de contrôle et à supposer que le système centralisé ne nécessite pas de sectoriser l'espace. C'est le cas par exemple avec un contrôle complètement automatisé, où l'on peut envisager qu'un seul calculateur résolve l'ensemble des conflits aériens sur toute l'Europe (ou tout autre région du monde). Sans aller jusqu'à une automatisation complète et en laissant le contrôleur humain au coeur du système, certains projets de recherche innovante d'Eurocontrol ([DGNS01]) envisagent un changement radical des méthodes de contrôle en définissant des secteurs de grande taille (*super-sector*), ou en supprimant la sectorisation (*sector-less*). Dans ce dernier cas, chaque contrôleur serait alors chargé de la sécurité d'un ou plusieurs vols entre origine et destination. Un certain nombre de tâches, comme le séquençage des avions au sein d'un flux, pourraient être déléguées aux pilotes des avions. La faisabilité et l'impact de ce type de délégation sont actuellement étudiés par Eurocontrol ([ZeEHR02],[ReEHZ02]). Le concept *sector-less* en est encore à ses balbutiements de l'avis même de ses auteurs. Il implique certainement une réorganisation de la structure de l'espace aérien, avec pour objectif de réduire les coordinations entre contrôleurs gérant les flux ou groupes d'avions.

On voit là tout l'intérêt qu'il y aurait à planifier des trajectoires séparées pour les flux d'avions les plus importants, ou encore pour des trains d'avions gérés chacun comme un groupe homogène.

Réguler le trafic et/ou modifier le réseau ?

Le lecteur pourra trouver dans la littérature un grand nombre de problèmes traitant de la régulation du trafic aérien. Considérant une structure donnée de l'espace aérien contrôlé (réseau de routes, secteurs aériens), l'objectif est alors de réguler le trafic pour le conformer à la capacité du système à traiter une certaine quantité de trafic.

A l'opposé, le système et son fonctionnement peuvent être modifiés et améliorés en prenant le trafic comme une donnée fixée, que l'on ne cherche pas à réguler. On peut alors faire varier la géométrie du réseau de routes, la sectorisation, l'orientation des flux d'avions à travers le réseau, pour répondre au mieux à la demande de trafic.

Ces deux approches sont bien entendu complémentaires et sont actuellement utilisées à des horizons temporels différents, la planification du réseau se faisant à un plus long terme que la régulation. Dans la suite de ce document, c'est à la deuxième approche que nous nous intéresserons. On ne cherchera pas à réguler la demande de trafic pour se conformer à la capacité du système, mais plutôt à augmenter celle-ci en modifiant le réseau aérien.

A quel horizon temporel se placer ?

Le trafic aérien est actuellement géré avec des horizons temporels divers, qui engendrent chacun des problèmes spécifiques. Le contrôle aérien proprement dit se fait en temps réel. Son objectif consiste à éviter les conflits entre les trajectoires d'avions. Ceux-ci sont résolus au plus tard quelques minutes avant la collision présumée. A un horizon temporel encore plus court, on trouve les systèmes

embarqués d'anticollision d'urgence. De nombreux travaux ont abordé les problèmes liés à l'automatisation d'un certain nombre de tâches du contrôleur (détection et résolution de conflits notamment), voire à l'automatisation totale du système, dans un contexte centralisé ou réparti. Ces problèmes ne seront pas abordés dans la thèse présentée ici.

La régulation du trafic se prépare quant à elle avec quelques jours d'avance, mais s'affine le jour même en fonction des impondérables du moment (conditions climatiques sur les plateformes aéroportuaires, activation réelles des zones militaires, etc...). Comme nous venons de le dire précédemment, la régulation de la demande n'est pas dans le champ de notre recherche. Toutefois, on pourra aborder d'autres problèmes, comme l'optimisation des schémas prévisionnels d'ouverture des secteurs, qui se situent à cet horizon temporel.

Les modifications stratégiques de la structure de l'espace contrôlé (routes, sectorisation, schémas d'orientation de trafic,...) se font généralement sur une base à long terme, mensuelle ou annuelle par exemple.

Cette imbrication de filtres (planification stratégique, régulation pré-tactique puis tactique, contrôle tactique, et en dernier recours anti-collision), opérant à des horizons temporels différents, a pour unique but d'assurer l'écoulement du trafic, en évitant que des avions entrent en collision. Rien n'interdit bien sûr de proposer des alternatives à ce système, mais il faudra tout de même tenir compte de ces différents horizons temporels.

2.3.2 ... et aussi une question de critères

En dehors des hypothèses sur le contexte, une autre question que l'on doit se poser est de savoir selon quels critères cherche-t-on exactement à améliorer notre système ? Selon le contexte choisi et le problème que l'on se pose, on pourra chercher à :

- diminuer le nombre de conflits entre trajectoires,
- minimiser les allongements de trajectoires,
- réduire les délais imposés au décollage,
- alléger la charge de travail du contrôleur aérien,
- mieux répartir la charge de travail entre contrôleurs aériens,
- etc...

Notons que ces critères peuvent être antagonistes : par exemple, résoudre des conflits amène généralement à dévier, et donc allonger, des trajectoires. D'un autre côté, on pourra être amené à améliorer simultanément plusieurs critères. La plupart des problèmes que l'on se posera seront multiobjectifs : séparer les principaux flux de trafic en minimisant les allongements de trajectoires, ou encore répartir au mieux la charge de travail entre plusieurs équipes de contrôleurs tout en restant en dessous des capacités des secteurs de contrôle, etc...

Selon le cas, nous chercherons soit à résoudre au mieux des problèmes issus d'un contexte réel en utilisant les critères objectifs les plus réalistes possibles, soit à résoudre d'abord un problème épuré en utilisant des critères et une modélisation simples, que nous enrichirons ensuite pour tenir compte des contraintes du monde réel.

Après cette présentation générale du système existant et de ses évolutions possibles, et après cette discussion sur le choix du contexte et des critères d'optimisation, nous allons, dans le chapitre suivant, présenter les méthodes et les algorithmes que nous appliquerons par la suite, dans les différentes étapes de notre démarche.

Chapitre 3

Description des algorithmes

3.1 Introduction

Dans ce chapitre, nous allons présenter les méthodes et algorithmes employés dans la suite de la thèse. La première section décrit la méthode de classification par nuées dynamiques qui sera ensuite utilisée au chapitre 5 pour grouper les points de croisement des routes directes selon leur proximité. Cette même méthode sera également employée dans le chapitre 8 pour regrouper les avions en sous-flux ayant des caractéristiques similaires, pour les niveaux de vol en entrée, en croisière, et en sortie de l'espace. Au chapitre 9, elle sera à nouveau employée pour regrouper les avions en *trains d'avions*, selon un critère temporel.

Nous présenterons ensuite, en section 3.4, l'algorithme de Fortune, permettant de calculer le diagramme de Voronoï d'un ensemble de points dans le plan. Cette méthode sera utilisée au chapitre 5 pour calculer les mini-secteurs associés aux nouveaux points de croisement du réseau. La triangulation de Delaunay, duale du diagramme de Voronoï, nous permettra de calculer les nouveaux tronçons de routes du réseau.

La section 3.5.2 décrit deux algorithmes classiques de recherche dans des arbres : le *Branch & bound* et l' A^* . Ces algorithmes seront tous deux utilisés au chapitre 4 pour équilibrer des configurations de secteurs. L'algorithme A^* sera en outre utilisé au chapitre 5 pour trouver des plus courts chemins dans un réseau de routes aériennes, puis aux chapitres 6, 7, 8 pour trouver, de façon itérative, des trajectoires 3D séparées, et enfin au chapitre 9 pour calculer des trajectoires 4D sans conflits.

Enfin, la dernière partie de ce chapitre est consacrée aux algorithmes génétiques, avec deux types de codage des données : binaire, ou réel. Les algorithmes avec un codage binaire, historiquement les plus anciens, sont les plus proches, dans leur formalisme, du modèle d'évolution biologique qui les a inspiré. Il permettent au profane de bien appréhender les mécanismes fondamentaux mis en jeu, et facilitent la compréhension des algorithmes avec un codage réel (RCGA : *Real Coded Genetic Algorithms*), qui seront utilisés, dans la suite de la thèse, au chapitre 4 pour regrouper les secteurs de façon optimale, puis aux chapitres 7 et 8 pour calculer des trajectoires 3D séparées, en optimisant globalement l'ensemble des trajectoires. Les algorithmes génétiques avec un codage réel ne sont pas une transposition aussi directe du modèle biologique que ceux avec un codage binaire. Depuis leur apparition, les algorithmes fondés sur le paradigme de l'évolution naturelle ont subi de nombreuses modifications, et donné lieu à des variantes parfois éloignées des algorithmes originels, et l'on désigne souvent cette classe d'algorithmes sous le nom d'*algorithmes évolutionnaires*.

3.2 Justification du choix des méthodes employées

Le choix des méthodes employées est fonction en premier lieu de la nature des problèmes spécifiques abordés dans cette thèse : classification, triangulation et sectorisation, ou bien optimisation. Dans le cadre de l'optimisation des flux de trafic aérien, ces méthodes sont complémentaires et nous permettent de traiter différents aspects de notre sujet, en suivant la démarche présentée dans l'introduction générale.

Parmi les divers algorithmes de classification, nous avons retenu un algorithme de partitionnement par nuées dynamiques, non-hierarchique, qui correspondait mieux aux besoins rencontrés que les méthodes de classification hiérarchiques. Pour la triangulation et la sectorisation, nous avons choisi l'algorithme de S. Fortune, de complexité $O(n \cdot \log(n))$.

Pour l'optimisation, le choix de la méthode utilisée dépend aussi de la difficulté du problème. Certains problèmes sont accessibles à des méthodes déterministes classiques. Pour calculer séquentiellement des trajectoires dans un espace encombré d'obstacles, le choix d'un algorithme A^* est tout à fait justifié, tout comme l'utilisation du *Branch & bound* sur des instances abordables du problème de regroupement de secteurs.

Dans certains cas, le problème est trop difficile pour être traité par de telles méthodes. La notion de problème d'optimisation difficile couvre notamment les problèmes à forte combinatoire, les problèmes NP-difficiles, ou ceux pour lesquels il n'existe pas de méthode de résolution exacte connue. Il existe tout un ensemble de méthodes, que l'on désigne sous le terme de *méta-heuristiques*, développées dans le but de résoudre de tels problèmes. On peut citer, parmi ces *méta-heuristiques*, les algorithmes de recuit simulé, les algorithmes évolutionnaires, les méthodes de recherche Tabou, les algorithmes de colonies de fourmis. Ces méthodes ont en commun une approche *stochastique*, introduisant une marche aléatoire dans l'espace d'états. Un certain nombre de mécanismes, inspirés par des analogies avec le monde réel, permettent d'orienter cette marche aléatoire vers des solutions optimales, ou presque optimales.

S'il existe des résultats théoriques pour certaines *méta-heuristiques* – le lecteur pourra se reporter p.25 de [DPST03] pour une présentation générale des résultats convergence théorique du recuit simulé, ou au travaux de Raphaël Cerf ([Cer91]) sur la convergence asymptotique des algorithmes génétiques – ceux-ci sont de peu d'utilité en pratique, que ce soit pour guider le choix d'une méthode particulière, ou pour le réglage des valeurs des paramètres des algorithmes.

Parfois, la nature du problème peut orienter le choix d'une *méta-heuristique* : par exemple, le recuit simulé ou les stratégies d'évolution (ES) pourront s'avérer plus adaptés que l'algorithme génétique lorsque l'on a des difficultés à définir pour ce dernier un opérateur de croisement qui ait un sens. Dans le cadre de cette thèse, nous avons choisi un algorithme génétique avec un codage réel, pour le regroupement optimal des secteurs, et pour l'optimisation globale d'un ensemble de trajectoires 3D. Ce choix se justifie, d'une part, par la difficulté des problèmes abordés, qui sera discutée dans les chapitres concernés, et, d'autre part, par le fait que, dans les deux cas, il est possible de produire un opérateur de croisement dont on peut espérer qu'il favorise la recherche d'un optimum. D'autres méta-heuristiques auraient certes pu être testées sur les problèmes cités. Ce travail reste à produire. Le choix du codage réel, plutôt que du codage binaire, est justifié en section 3.6.

Nous allons maintenant aborder la présentation des algorithmes, en commençant par la classification par nuées dynamiques.

3.3 Méthode de classification par nuées dynamiques

3.3.1 Généralités sur les méthodes de classification

Cette section regroupe quelques notions utiles sur les méthodes de classification automatique, et en particulier la méthode de type «nuées dynamiques» (cf [CDG⁺89], et plus spécifiquement [Mac67] pour la méthode des *k-means*) utilisée à plusieurs reprises dans cette thèse, en particulier dans le chapitre 5 pour l'élaboration d'un réseau de routes.

Le but de ces méthodes est de construire une partition d'un ensemble d'objets, ou une hiérarchie de partitions pour les méthodes hiérarchiques, en fonction d'un critère de similarité ou de proximité. La ou les partitions recherchées doivent être telles que, d'une part, les différents groupes d'une partition soient les plus dissemblables (éloignés) possible, et, d'autre part, les éléments d'un groupe donné soient les plus semblables (proches) possible, selon le critère choisi.

Au vu du nombre de partitions d'un ensemble à n éléments (cf section 3.3.2), il n'est pas envisageable, sauf dans des cas élémentaires, de construire toutes les partitions possibles, puis de choisir la meilleure. C'est pourquoi les algorithmes de classification procèdent par itérations successives à partir d'une partition initiale, en améliorant à chaque étape le critère de dispersion choisi. Ils ne garantissent donc pas de trouver la partition globalement optimale pour le critère choisi, mais seulement de trouver un optimum local, dépendant du choix de la partition initiale.

Dans la littérature, on distingue généralement deux types de méthodes :

- les méthodes hiérarchiques, qui procèdent soit par regroupements successifs d'éléments semblables (méthodes ascendantes), soit en partant d'une classe unique et en scindant la classe la plus hétérogène (méthodes descendantes). Le résultat obtenu est alors un "dendrogramme" (un arbre) représentant un ensemble de partitions hiérarchisées selon le critère de similarité choisi (souvent une distance moyenne entre groupes, ou encore une distance minimale ou maximale). Les résultats trouvés peuvent varier sensiblement selon le critère choisi.
- les méthodes non-hiérarchiques ou de partitionnement, qui ne produisent pas de dendrogrammes, mais permettent d'obtenir une partition de l'ensemble d'objets. C'est à celles-ci que nous allons nous intéresser dans ce qui suit.

3.3.2 Partitions d'un ensemble à n éléments

Intéressons-nous au nombre de partitions d'un ensemble de n éléments en k sous-ensembles. Soit $P(n, k)$ ce nombre. Il vérifie les relations suivantes :

$$\begin{aligned} \forall n \geq 1 \\ P(n, 1) &= 1 \quad (\text{un seul groupe de } n \text{ éléments}) \\ P(n, n) &= 1 \quad (\text{une seule partition avec } n \text{ groupes d'un élément}) \\ P(n, k) &= 0 \text{ if } k > n \quad (\text{on ne peut pas faire plus de } n \text{ groupes}) \\ P(n, k) &= k * P(n - 1, k) + P(n - 1, k - 1) \text{ if } 1 < k < n \end{aligned}$$

Pour comprendre la dernière relation, imaginons un processus itératif de création des partitions, où l'on considère successivement chaque élément de l'ensemble. Pour obtenir $P(n, k)$ partitions lorsqu'on considère le $n^{\text{ième}}$ élément, il y a deux possibilités :

- soit cet élément a été ajouté dans un des k groupes déjà définis à l'itération $n - 1$, ce qui fait $k * P(n - 1, k)$ possibilités,
- soit il forme une nouvelle partition à lui tout seul, ce qui signifie qu'on n'avait défini que $k - 1$ partitions à l'itération $n - 1$ (soit $P(n - 1, k - 1)$ possibilités).

Le nombre $P(n, k)$ est identifié dans la littérature comme le *nombre de Stirling de deuxième espèce*.

Le nombre de partitions non vides d'un ensemble à n éléments (*nombre de Bell*), sera alors :

$$P(n) = \sum_{k=1}^n P(n, k)$$

D'un point de vue algorithmique, l'ensemble des partitions peut théoriquement être obtenu en explorant l'arbre des possibilités, comme illustré sur la figure 3.1. En pratique, cela n'est possible que pour de faibles valeurs de n . Les dix premiers nombres de Bell sont : 1, 2, 5, 15, 52, 203, 877, 4140, 21147, 115975. Le nombre de partitions croît donc très rapidement en fonction des valeurs de n (pour 15 éléments, on est déjà à 1382958545 partitions).

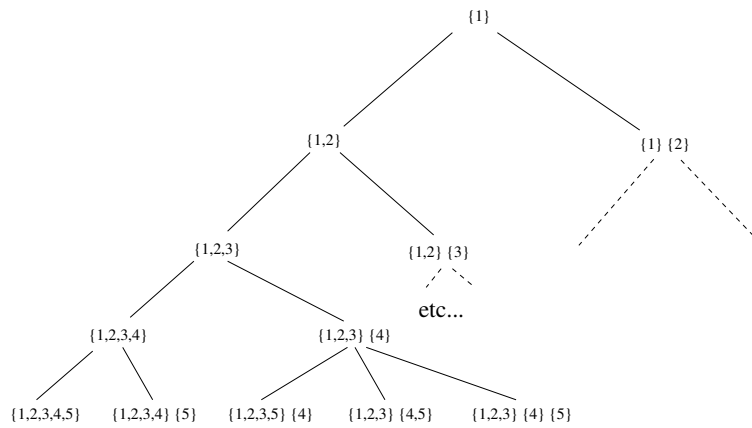


FIG. 3.1 – Construction des partitions d'un ensemble à 5 éléments.

3.3.3 Les méthodes de partitionnement

Typiquement, on cherchera une partition en k groupes d'un nuage de n points de l'espace considéré, ceci en fonction de la densité de ces points. Le critère de proximité peut être une distance, ou bien une inertie.

Plus formellement, plaçons-nous dans un espace muni d'une distance d . Soit $C = \{C_1, C_2, \dots, C_k\}$ un ensemble des k classes (ou groupes) constituant une partition de l'ensemble des n points. On peut, à titre d'exemple, se doter d'un critère de qualité de la partition fondé sur l'inertie barycentrique. Pour cela, on définit d'une part l'inertie intra-classe par :

$$I_W = \sum_{i=1}^k \sum_{j \in C_i} m_j d^2(p_j, g_i)$$

où m_j est la masse du point j , p_j sa position, et où g_i est le barycentre de la classe C_i , et d'autre part l'inertie inter-classe par :

$$I_B = \sum_{i=1}^k M_i d^2(g_i, g)$$

où $M_i = \sum_{j \in C_i} m_j$ est la masse de la classe C_i , et où g est le barycentre du nuage des centres de classe.

L'inertie totale est alors (théorème de Huyghens sur les moments d'inertie) :

$$I = I_B + I_W$$

Par analogie avec la variance $var(X) = \frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2$ d'une variable X , qui mesure une dispersion unidimensionnelle, avec un poids uniforme de $1/n$ associé à chaque observation, l'inertie pourra être vue comme une mesure de dispersion multidimensionnelle.

Le but de la classification sera alors de minimiser l'inertie intra-classe pour obtenir des groupes homogènes, ce qui équivaut à rendre maximale l'inertie inter-classe représentant l'hétérogénéité des différents groupes.

La méthode de partitionnement de base, dite des «centres mobiles», est due à Forgy ([For65], voir aussi [Sap90]). Au départ, k points c_1, c_2, \dots, c_k sont choisis arbitrairement ou tirés au hasard, et pris comme centres des classes C_1, C_2, \dots, C_k d'une partition initiale du nuage de points. On identifie ensuite, pour chaque centre de classe, une zone d'espace associée, dont les frontières sont les hyperplans médiateurs entre le centre de la classe considérée et les centres voisins (à ce sujet, le lecteur pourra se reporter à la section 3.4 sur les diagrammes de Voronoï). Les classes sont alors complétées en fonction de la zone dans laquelle se situe chacun des points restants. Chaque classe C_i contient alors les points les plus proches du centre c_i que de tout autre centre.

On remplace alors les k points pris au hasard par les barycentres des classes et on recommence. Cet algorithme converge en un nombre assez faible d'itérations vers un optimum local, dont on montre qu'il minimise l'inertie intra-classe.

Il existe un certain nombre de variantes améliorées de cette méthode. Dans l'algorithme de Forgy, on attend que tous les individus aient été affectés dans une classe avant de recalculer les centres de gravité et de recomposer les classes. La variante des *k-means* ([Mac67]) recalcule les centres et recompose les classes à chaque affectation de point dans une classe. Une extension des précédentes méthodes, proposée par E. Diday [CDG⁺89] est la méthode des «nuées dynamiques», dans laquelle une classe n'est plus forcément représentée par un centre de classe du type barycentre, mais par un noyau qui peut être un ensemble de points centraux, un axe principal, etc. Il faut alors définir formellement, en plus du critère de proximité :

- une fonction de représentation des classes permettant de calculer le noyau à partir des éléments de la classe,
- une fonction d'affectation des objets aux classes.

L'algorithme consiste alors, à partir d'une partition initiale choisie ou tirée au hasard, à itérer les phases d'affectation et de représentation, jusqu'à la convergence.

3.3.4 Description de l'algorithme utilisé

L'algorithme utilisé à plusieurs reprises dans cette thèse nécessite de se doter des fonctions suivantes, opérant sur des points d'un espace à définir en fonction de l'utilisation de l'algorithme :

- une distance,
- une fonction calculant le barycentre de deux points, chaque point ayant une masse associée (1 par exemple, ou toute autre masse définie en fonction des besoins de l'utilisateur de l'algorithme),

et également deux seuils de décision, permettant de régler la granularité des agrégats créés par l'algorithme :

- d_1 : un point sera intégré à l'agrégat le plus proche (avec la distance choisie) s'il est à une distance inférieure à d_1 du barycentre de l'agrégat,

- d_2 : deux agrégats seront fusionnés s'ils sont à une distance inférieure à d_2 .

L'algorithme proprement dit consiste à traiter itérativement chaque point de l'ensemble comme décrit sur la figure 3.2.

Pour chaque point p :

1. Rechercher l'agrégat \mathcal{G} le plus proche
2. Si la distance de p au centre de classe de \mathcal{G} (le barycentre) est supérieure à d_1 :
 - (a) Créer un nouvel agrégat comportant le point p comme unique élément (celui-ci étant naturellement le centre de classe)
 - (b) Passer au point suivant
3. Sinon :
 - (a) Ajouter le point p à l'agrégat \mathcal{G}
 - (b) Recalculer le centre de classe de l'agrégat. C'est le barycentre du point p , avec sa masse associée, et du précédent centre de classe ayant pour masse associée la somme des masses individuelles des éléments de \mathcal{G} (avant ajout du nouveau point)
 - (c) Examiner s'il faut fusionner certains agrégats. Ceci se fait récursivement, avec une fonction *fusion* opérant sur le nouvel agrégat et la liste des autres agrégats. On parcourt itérativement cette liste, en examinant la distance entre le nouvel agrégat et un agrégat de la liste :
 - si cette distance est supérieure à d_2 , on passe à l'agrégat suivant dans la liste. La fusion s'arrête lorsqu'il ne reste aucun autre agrégat à examiner
 - si la distance est inférieure à d_2 , on fusionne le nouvel agrégat et celui de la liste, et on applique à nouveau la fonction *fusion* sur l'agrégat ainsi créé et la liste complète, de laquelle on retire l'agrégat qui a été fusionné.

FIG. 3.2 – Algorithme de classification.

3.4 Diagrammes de Voronoï et triangulation de Delaunay

3.4.1 Définition d'un diagramme de Voronoï

Dans la méthode des centres mobiles évoquée en section 3.3, nous avons vu qu'il pouvait être utile, pour chaque centre de classe, de trouver la zone d'espace associée, dont les frontières sont les hyper-plans médiateurs entre le centre de la classe considérée et les centres voisins. En dimension 2, le diagramme représentant ces zones est le diagramme de Voronoï, pour lequel nous allons présenter un algorithme qui nous servira dans la construction d'un réseau de routes aériennes au chapitre 5.

Le lecteur pourra se reporter à [For92], pour plus de détails sur les diagrammes de Voronoï, et sur la triangulation de Delaunay, ainsi qu'à [For95], où l'on trouvera une présentation de différents algorithmes, avec une comparaison de leur complexité.

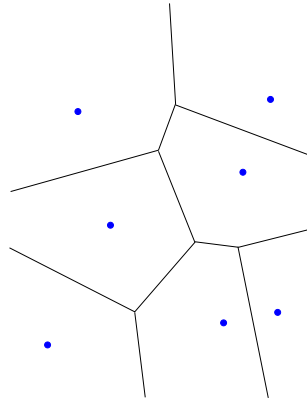


FIG. 3.3 – Diagramme de Voronoï.

Une illustration classique d'un diagramme de Voronoï consiste à considérer le problème de la distribution du courrier, en considérant que chaque habitant sera desservi par le bureau de poste le plus proche (à vol d'oiseau). Une bonne démarche consiste alors à découper le pays en zones, chacune d'entre elles associée à un bureau de poste. La définition d'une telle zone est que chacun des points la constituant est plus proche du bureau de poste de la zone que d'aucun autre bureau de poste. Le résultat d'un tel découpage est le diagramme de Voronoï associé aux sites des bureaux de poste. Dans la suite, on nommera *sites* les points à partir desquels on cherche à construire le diagramme de Voronoï.

3.4.2 Dualité entre la triangulation de Delaunay et le diagramme de Voronoï

La triangulation de Delaunay T d'un ensemble de points E est la triangulation telle qu'aucun triangle ABC de T ne contient un autre point de E à l'intérieur de son cercle circonscrit.

Le diagramme de Voronoï est le dual de la triangulation de Delaunay. Pour obtenir une triangulation de Delaunay à partir d'un diagramme de Voronoï, il suffit de relier les sites entre lesquels il existe une frontière (une arête du diagramme de Voronoï), comme illustré en figure 3.4.

En conséquence, nous n'utiliserons qu'un seul algorithme, calculant le diagramme de Voronoï d'un ensemble de points, à partir duquel nous serons en mesure de déduire la triangulation de Delaunay de ce même ensemble de points.

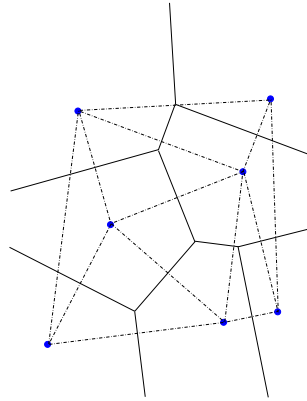


FIG. 3.4 – Diagramme de Voronoï et triangulation de Delaunay.

3.4.3 L'algorithme de Fortune

L'algorithme choisi pour déterminer le diagramme de Voronoï associé à un ensemble de points du plan est l'algorithme de Fortune ([For92]). Une description de cet algorithme de complexité $O(n \cdot \log(n))$ peut également être trouvée dans [dBvKOS00]. Le principe de base consiste à balayer le plan avec une ligne horizontale (ou verticale, au choix) en tenant à jour un certain nombre d'informations nécessaires à la détermination des sommets du graphe de Voronoï.

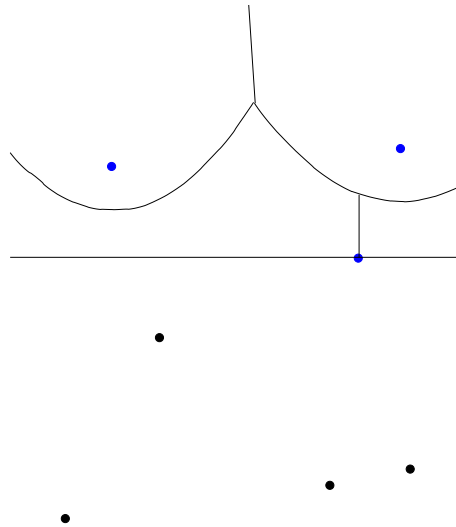
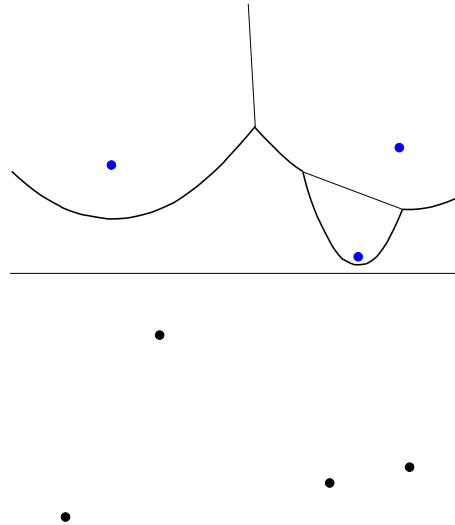


FIG. 3.5 – Algorithme de Fortune : événement «site».

Remarquons d'abord que le lieu des points équidistants entre un site et la droite de balayage est une parabole. En conséquence, si i et j sont deux sites, \mathcal{D} la droite de balayage, et d la fonction distance, le point d'intersection p des deux paraboles est tel que $d(i, p) = d(p, \mathcal{D}) = d(j, p)$. Donc l'intersection des deux paraboles est à égale distance des deux sites, et, lors du déplacement de la droite de balayage, cette intersection décrira la médiatrice des deux sites, c'est-à-dire précisément la frontière que l'on recherche entre les zones associées aux sites.

Toutefois, toutes les médiatrices ne font pas partie du diagramme de Voronoï. Il faut donc gérer, pendant le balayage, les événements de création et de destruction des arcs de paraboles, dont on sait

FIG. 3.6 – Algorithme de Fortune : ligne de ressac (*beachline*).

qu'il généreront des segments de médiatrice faisant partie du diagramme de Voronoï. L'enveloppe des arcs de paraboles utiles à la mise à jour du diagramme de Voronoï peut être vue comme une ligne de ressac (*beachline* est le terme utilisé dans [dBvKOS00]) dont chacun des points de cassure génère une arête du diagramme de Voronoï, lorsque la ligne de balayage se déplace.

Un arc de parabole sera créé à chaque fois que la ligne de balayage rencontre un site. On parle alors d'«événement site».

Un arc de parabole disparaîtra lorsqu'il sera réduit à un point. Dans ce cas, les deux arcs de paraboles voisins de celui qui disparaît s'intersectent au point auquel se réduit ce dernier. Les distances entre chacun des trois foyers et le point en question sont alors égales entre elles, et égales à la distance entre ce point et la droite de balayage.

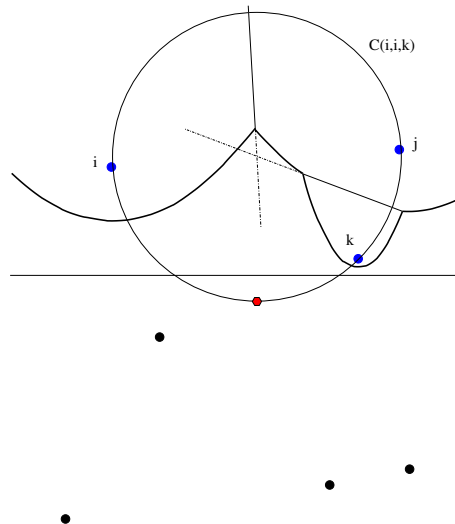


FIG. 3.7 – Algorithme de Fortune : cercle passant par les foyers de trois arcs consécutifs de la ligne de ressac.

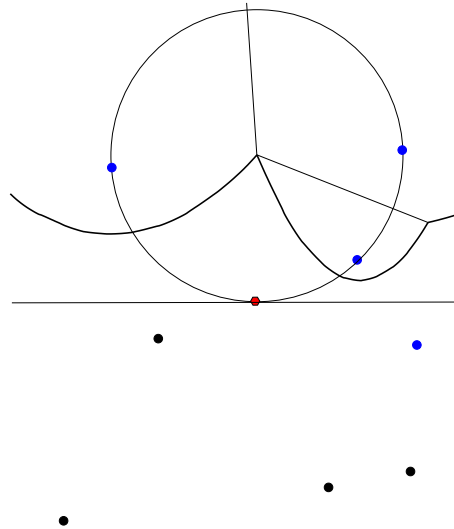


FIG. 3.8 – Algorithme de Fortune : «événement cercle».

Le point de disparition de l'arc de parabole (marquant le bout d'une arête du diagramme de Voronoï) est donc le centre du cercle passant par les trois foyers et tangent à la droite de balayage. On parle d'«événement cercle» lorsque la droite de balayage rencontre le point extrémal¹ du cercle passant par les foyers de trois arcs de paraboles consécutifs de la ligne de ressac. Cet événement correspond à la disparition d'un des arcs de parabole.

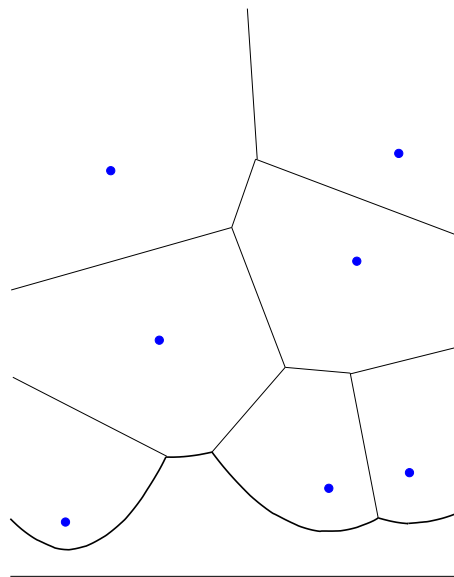


FIG. 3.9 – Algorithme de Fortune : diagramme de Voronoï en construction.

¹Par point extrémal, on entend ici le point d'abscisse minimale, si l'on suppose que l'on balaye de bas en haut.

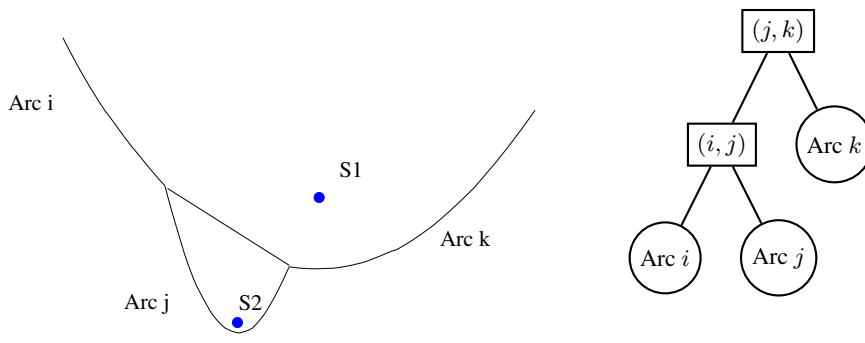


FIG. 3.10 – Arbre de représentation de la ligne de ressac.

3.4.4 Implantation

Une implantation en langage Ocaml de l'algorithme de S. Fortune a été réalisée. Elle tient essentiellement en deux points : la gestion de la liste des événements («site» ou «cercle») triée par ordre d'occurrence, et la gestion de la ligne de ressac. Le résultat (le diagramme de Voronoï) est représenté par une table permettant d'associer à tout couple (p, q) l'arête, si elle existe, qui est le segment de la droite médiatrice entre les deux sites S_p et S_q .

Les événements sont gérés dans une *priority queue* triée par ordre d'occurrence (ordonnée ou abscisse, selon le choix du balayage).

Une structure d'arbre binaire est utilisée pour représenter la ligne de ressac, les feuilles de cet arbre étant les arcs de paraboles de la ligne de ressac, et les noeuds étant les points d'intersection de ces arcs. A chaque noeud est donc associé un couple d'entiers identifiant l'arc le plus à droite de la branche gauche et l'arc le plus à gauche de la branche droite : la donnée de ces deux arcs permettant de retrouver leur intersection.

L'insertion d'un nouvel arc dans l'arbre lors d'un événement «site» se fait de la manière suivante : on recherche sur quel arc existant va venir se greffer le nouvel arc. Cette recherche se fait en comparant l'abscisse (dans le cas d'une ligne de balayage horizontale) du site rencontré à celles des points d'intersection définis par les noeuds parcourus. L'arc existant se scinde alors en deux nouveaux arcs (ayant le même foyer), entre lesquels vient s'intercaler le nouvel arc.

La suppression d'un arc se fait à l'occasion d'un événement «cercle». Il faut alors rechercher dans l'arbre de représentation l'arc à supprimer. Il s'agit de l'arc du milieu, parmi les trois arcs consécutifs de la ligne de ressac qui sont à l'origine de l'événement cercle.

Pour peu que l'arbre binaire représentant la ligne de ressac soit équilibré, ces opérations d'insertion et de suppression seront de complexité $O(\log(n))$. Le rééquilibrage se fait par rotation droite ou gauche de chaque noeud pour lequel la différence de hauteur entre les branches gauche et droite est supérieure à 1, lors des opérations d'insertion ou de suppression modifiant l'arbre.

L'algorithme s'écrit donc comme indiqué en figure 3.13, où \mathcal{Q} est la *priority queue* utilisée pour gérer les événements dans l'ordre chronologique, \mathcal{T} est l'arbre représentant la ligne de ressac, et \mathcal{D} est le diagramme de Voronoï.

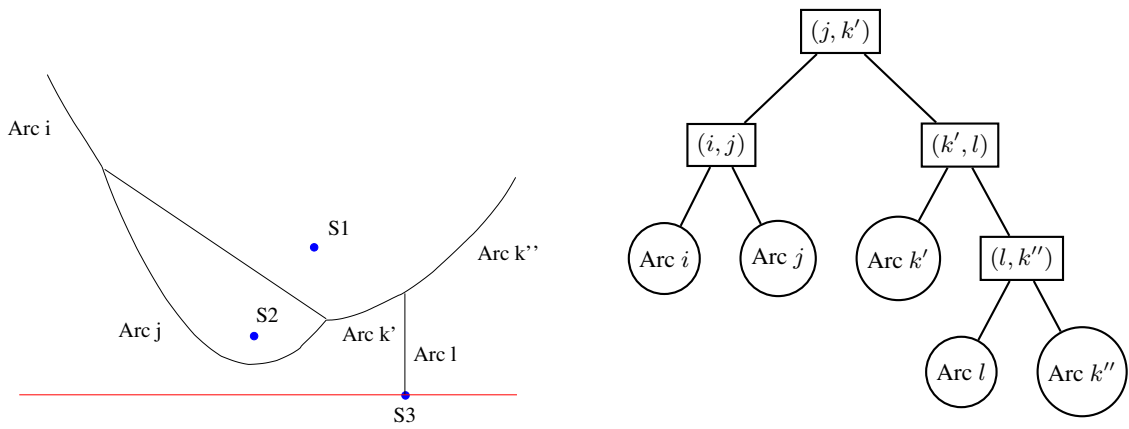


FIG. 3.11 – Insertion d'un nouvel arc.

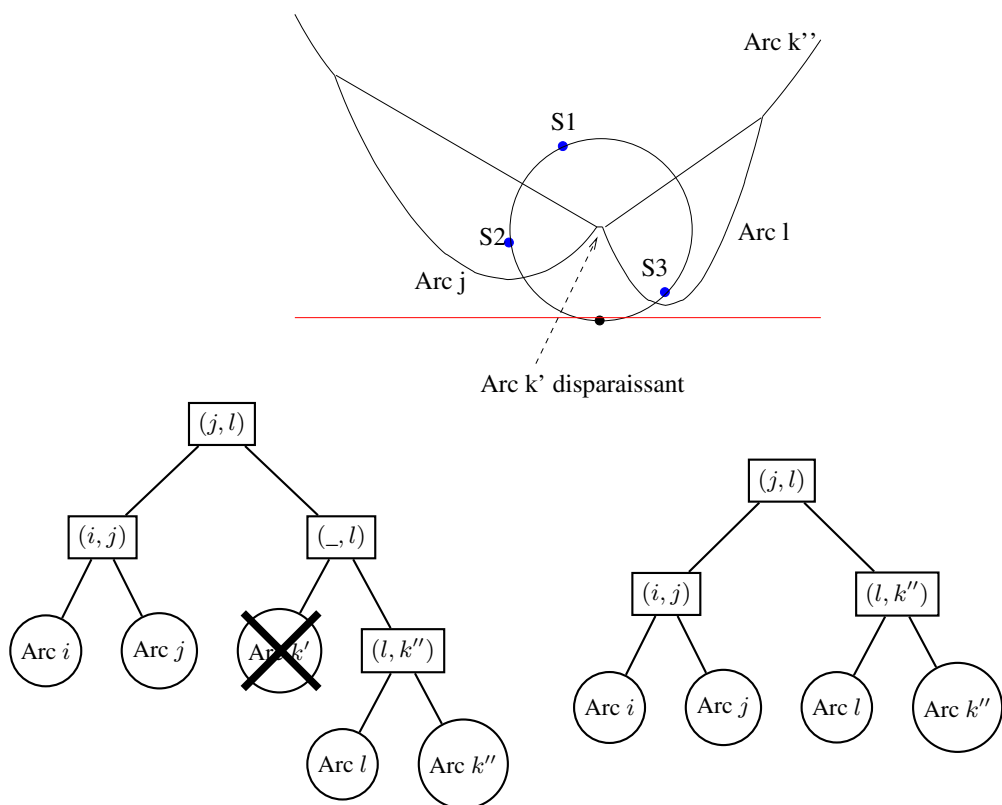


FIG. 3.12 – Suppression d'un arc lors d'un événement cercle.

-
1. **Tant que** Q n'est pas vide :
 2. Retirer de Q le prochain événement
 3. **Si** c'est un événement site
 4. **Alors** $\text{TRAITER_EVT_SITE}(\mathcal{T}, \mathcal{D}, Q, S_p)$ où S_p est le nouveau site :
 - (a) Trouver dans \mathcal{T} l'arc situé «en face» du site S_p . Soit S_q le site foyer de cet arc
 - (b) Supprimer dans Q tous les événements cercles liés à cet arc
 - (c) Remplacer cet arc par un sous-arbre comprenant trois nouveaux arcs : celui dont le foyer est S_p et deux nouveaux arcs (à gauche et à droite) issues de l'arc remplacé de foyer S_q
 - (d) Associer au couple (p, q) dans \mathcal{D} une arête (dont les extrémités sont encore inconnues)
 - (e) Chercher les nouveaux événements cercle et les insérer dans la *priority queue* Q
 5. **Sinon** $\text{TRAITER_EVT_CERCLE}(\mathcal{T}, \mathcal{D}, Q, i, j, k, c)$ où c est le centre du cercle passant par les trois foyers des arcs i, j, k consécutifs de \mathcal{T} :
 - (a) Trouver l'arc j dans \mathcal{T}
 - (b) Supprimer dans Q les événements cercle liés à cet arc
 - (c) Mettre à jour \mathcal{D} : si S_p, S_q, S_r sont les foyers des arcs i, j, k respectivement, le point c est une extrémité des arêtes associées à (p, q) et (q, r) . Notons que S_p et S_r peuvent être un même point.
 - (d) Chercher les nouveaux événements cercle et les insérer dans la *priority queue* Q .
 6. Les points restants de la ligne de ressac (dans \mathcal{T}) sont les extrémités des arêtes du diagramme se prolongeant à l'infini dans la direction vers laquelle se déplace la ligne de balayage. Compléter \mathcal{D} en calculant l'intersection de ces arêtes avec la *bounding box*
-

FIG. 3.13 – Algorithme de Fortune.

3.5 Algorithmes d'exploration d'arbres

3.5.1 Séparation & évaluation (*Branch & bound*)

Principe du B&B

Le principe de l'algorithme de Séparation & évaluation, plus connu sous son appellation anglaise *Branch & bound* (BB, ou B&B), est assez simple.

Considérons un arbre d'états, avec des noeuds, et des feuilles représentant des états solutions. Dotons-nous d'une fonction de coût ayant la propriété suivante : le coût d'un noeud est un minorant des coûts des solutions pouvant être atteintes à partir de ce noeud. Le principe, si l'on cherche une solution de coût minimal, consiste à mémoriser la solution de plus bas coût rencontrée durant l'exploration, et à comparer le coût de chaque noeud parcouru à celui de la meilleure solution. Si le coût du noeud considéré est plus élevé que le meilleur coût, on arrête l'exploration de la branche : toutes les solutions de cette branche seront nécessairement de coût plus élevé que la meilleure solution déjà trouvée.

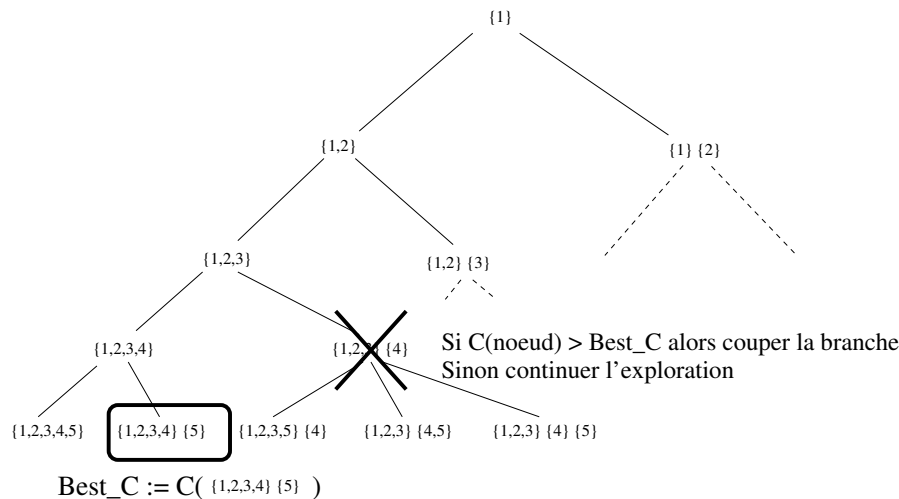


FIG. 3.14 – Exemple de parcours d'arbre par un *Branch & bound*.

La figure 3.14 illustre ce mécanisme sur un arbre permettant d'explorer toutes les partitions possibles d'un ensemble d'éléments, que nous avons déjà vu en section 3.3. Sur cette figure, C est la fonction de coût, et $Best_C$ le meilleur coût trouvé pour une solution. Pour rendre cet exemple plus concret, on peut s'imaginer que les éléments sont des secteurs de contrôle, et les partitions des configurations de regroupements, et que l'on cherche à équilibrer la charge de travail entre les groupes de secteurs. Nous verrons au chapitre 4 qu'il faut bien choisir la fonction de coût, afin que le coût d'un noeud minore celui de n'importe quelle solution atteignable à partir du noeud...

Illustration du B&B sur un exemple

Illustrons le fonctionnement du *Branch & bound* sur un exemple plus simple. Considérons un réseau routier reliant un certain nombre de villes, et supposons que l'on cherche le parcours le plus direct entre une ville de départ et une destination.

Un noeud sera ici un chemin dans le réseau, dont une extrémité sera la ville de départ. On saura que l'on a atteint une feuille de l'arbre (un état solution) lorsque l'autre extrémité du chemin sera la

ville de destination. La racine de l'arbre est un chemin de longueur nulle, ne contenant que la ville de départ. Les noeuds fils sont simplement les chemins obtenus en considérant les villes voisines de la dernière ville atteinte, comme illustré sur la figure 3.15. On choisit comme fonction de coût la distance parcourue.

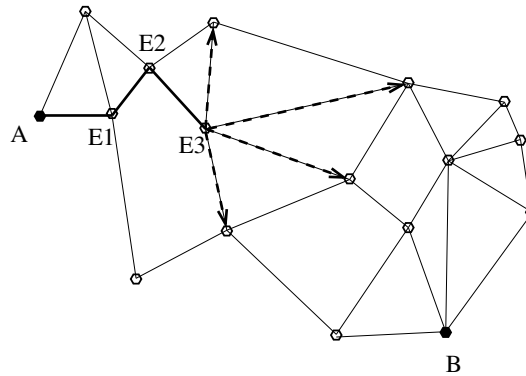


FIG. 3.15 – Choix des noeuds fils.

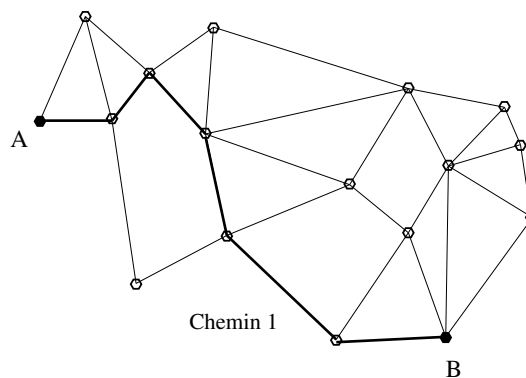


FIG. 3.16 – Premier chemin trouvé avec une recherche en profondeur.

Supposons qu'en explorant l'arbre en profondeur, on ait trouvé le chemin 1, décrit en figure 3.16. Cette solution est mémorisée, puis on remonte dans l'arbre d'états en revenant à la ville précédant la ville d'arrivée dans ce chemin, avant d'essayer un autre noeud fils, comme illustré sur la figure 3.17. Le nouveau chemin testé, qui se termine par la ville C, est plus long que la solution déjà trouvée. Ce n'est donc pas la peine de continuer à chercher à joindre la ville B en considérant les villes voisines de C, et on arrête l'exploration de cette branche de l'arbre d'états.

3.5.2 L'algorithme A^*

Principe de l' A^*

Le principe de l' A^* (cf [Pea84]) consiste à rechercher le meilleur parcours dans un arbre de possibilités, en repartant à chaque étape du meilleur noeud possible. Alors que le *Branch & bound* parcourrait l'arbre en profondeur (stratégie *depth-first*) en mémorisant la meilleure solution, l'algorithme A^* parcourt l'arbre selon une stratégie de type *meilleur noeud en premier*, ce qui implique de mémoriser chaque noeud parcouru, en ordonnant les noeuds selon un critère d'évaluation.

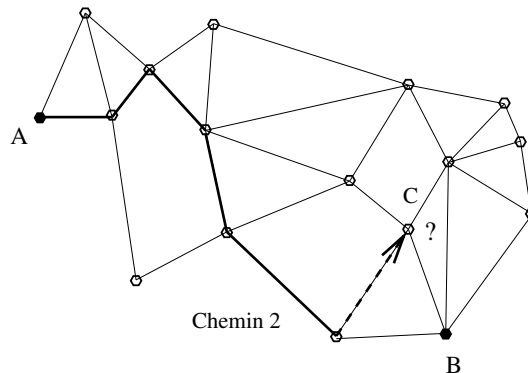


FIG. 3.17 – Essai d'un autre noeud fils.

L'algorithme A^* opère sur deux structures de données :

- un arbre d'états. La racine est l'état initial, les noeuds de l'arbre correspondent à des états intermédiaires, et les feuilles de l'arbre sont des états solutions.
- une file d'attente à priorités (*priority queue*), qui est une structure d'arbre binaire permettant de mémoriser des états et d'y accéder en fonction d'une priorité : l'état en tête de la file (on devrait dire la racine de l'arbre binaire) est celui de plus haute priorité (dans notre cas, de plus bas coût). On peut insérer ou extraire des états de la file d'attente, avec la propriété suivante : c'est toujours l'état de plus haute priorité qui est extrait.

Tant que l'on n'a pas atteint une feuille, ou terminé l'exploration de l'arbre :

1. Prendre le noeud P en tête de la file d'attente à priorités (*priority queue*)
 2. Calculer les états suivants (les noeuds fils)
 3. Pour chaque noeud fils F_j :
 - calculer le coût $\mathcal{C}(P, F_j)$ de la transition de l'état père à l'état fils,
 - calculer l'heuristique $\mathcal{H}(F_j)$,
 - insérer le noeud fils dans la file d'attente, avec une priorité dépendant de la somme des coûts des transitions passées et de $\mathcal{C}(P, F_j) + \mathcal{H}(F_j)$ (la valeur la plus basse se retrouve en tête de la file à priorités).
-

FIG. 3.18 – Algorithme A^* .

En plus de ces structures de données, il faut définir trois fonctions :

- une fonction évaluant le *coût de transition* entre deux états,
- une fonction *heuristique*, qui calcule une estimation minorante du coût des transitions restant à parcourir pour atteindre un état solution, à partir d'un état donné,
- une fonction déterminant si un état donné est un état solution ou non.

L'algorithme lui-même consiste à itérer les opérations décrites sur la figure 3.18, jusqu'à ce que l'on trouve un état solution (une feuille de l'arbre), ou bien jusqu'à ce que l'on soit sûr qu'il n'y ait

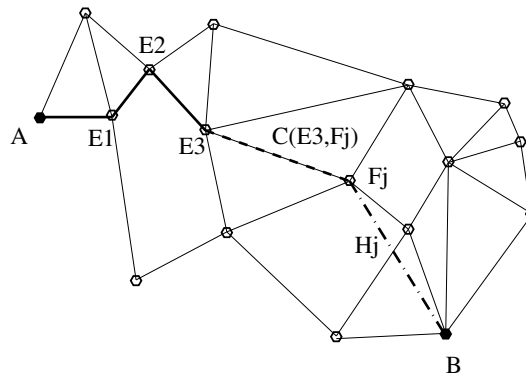


FIG. 3.19 – Coût de transition et heuristique.

pas de solution.

Illustration de l' A^* sur un exemple

Reprenons le même exemple que pour le *Branch & bound*, c'est-à-dire la recherche d'un plus court chemin sur un réseau routier.

Le coût d'une transition entre le noeud père et un noeud fils est alors simplement la distance entre la dernière ville du noeud père et la nouvelle ville ajoutée dans le noeud fils. Choisissons comme heuristique la distance à vol d'oiseau jusqu'à la ville d'arrivée. C'est bien une borne inférieure de tous les chemins terrestres menant à destination.

Pour chaque noeud fils, on peut donc calculer la somme des distances parcourues pour arriver à la dernière ville étape, et y ajouter la distance à vol d'oiseau entre cette ville et la destination. Le noeud est ajouté dans la file d'attente avec une priorité qui est fonction de cette valeur (on prend l'opposé par exemple).

Lorsqu'un des noeuds fils est la ville de destination, la recherche est terminée. On a alors trouvé le chemin le plus court entre la ville de départ et la ville d'arrivée.

Quelques commentaires sur l' A^*

L'algorithme A^* est tout à fait adapté à des problèmes de recherche de plus court chemin dans un espace encombré d'obstacles. Imaginons dans notre exemple précédent qu'un certain nombre de villes aient leur périphérie bloqué par des bouchons, ou des travaux, que l'on souhaite éviter. Il suffit dans ce cas d'interdire les étapes passant par ces villes lors du calcul des noeuds fils.

Les difficultés éventuelles que l'on peut rencontrer en utilisant cet algorithme sont de plusieurs types. Tout d'abord, il peut ne pas y avoir de solution : il peut n'exister aucun chemin évitant les bouchons. La modélisation choisie dans notre exemple conduit donc à des choix du type tout ou rien : soit l' A^* trouve un chemin, et c'est alors le plus court possible en fonction des contraintes, soit il ne trouve pas de chemin, et notre automobiliste reste chez lui. Si l'on souhaite faire en sorte qu'il parte tout de même, on peut chercher à minimiser son temps de parcours, bouchons éventuels compris, plutôt que la distance parcourue. Il faut alors changer de modélisation, et tenir compte dans les coûts de transition et dans l'heuristique des temps de parcours entre villes et du temps d'embouteillage prévu dans chaque ville.

Une autre difficulté, liée à la fois à la taille de l'arbre d'états et au choix de l'heuristique, est le *backtracking*, c'est-à-dire le retour-arrière dans l'arbre d'états. Supposons qu'au lieu de la distance

à vol d'oiseau, on ait choisi une heuristique nulle. Dans ce cas, l'état représentant le cheminement (A, E_1, E_2, E_3, F_j) de la figure 3.19 sera inséré dans la file d'attente avec une priorité moindre que n'importe quel chemin plus court, et donc avec une priorité inférieure à la plupart des noeuds parcourus précédemment. Par exemple, sur le schéma 3.19, n'importe quel noeud (A, E_1, X) , avec $X \neq E_2$, est un chemin plus court que le noeud que l'on vient d'insérer. Cela se traduit par de nombreux retours en arrière dans l'arbre d'états, afin d'explorer des sous-arbres paraissant plus prometteurs. Lorsque l'arbre d'états est de grande taille et lorsque l'heuristique choisie est trop minorante, le retour-arrière (*backtracking*) peut induire des temps de calcul trop importants pour espérer obtenir un résultat dans un délai raisonnable.

A l'inverse, le fait de choisir une heuristique qui n'est pas minorante peut conduire à trouver des solutions plus rapidement, mais qui ne sont pas optimales. Cela peut d'ailleurs être une stratégie pour éviter le *backtracking* lorsque l'on peut se contenter d'une solution non-optimale. Il suffit de définir une nouvelle heuristique en multipliant l'ancienne (minorante, elle) par un facteur légèrement supérieur à 1. Pour l'exemple de la section précédente, si l'on prend comme heuristique $1,05 \times d_{directe}$, on trouvera plus rapidement un chemin qui ne sera peut-être pas le plus direct, mais dont la longueur n'excédera pas de plus de 5% celle du chemin le plus court.

L'efficacité de l'algorithme dépend donc en grande partie du choix judicieux de la modélisation et de l'heuristique.

3.6 Algorithmes génétiques

3.6.1 Principe et généralités

Les algorithmes génétiques sont des algorithmes de recherche et d'optimisation globale inspirés du processus naturel de l'évolution. La sélection des individus les mieux adaptés, le croisement et la mutation à travers plusieurs générations d'une population donnée : voilà les principes qui ont été adaptés à l'optimisation par John Holland [Hol75]. Les travaux de David Goldberg ont également largement contribué à enrichir et mieux faire connaître le domaine [Gol89a, Gol89b]. Le lecteur trouvera dans le rapport de DEA de Y. Le Fablec [Fab92] et dans la thèse de N. Durand [Dur96] un bon état de l'art sur l'optimisation par algorithmes génétiques. Pour un descriptif des développements plus récents basés sur le paradigme de l'évolution naturelle, on pourra également consulter les ouvrages [ES03], [BFM00a], [BFM00b], et [DPST03].

L'idée est d'utiliser un *critère d'adaptation* pour les éléments de l'espace de recherche. Ce critère peut être la valeur d'une fonction à optimiser par exemple. On part d'un ensemble d'éléments pris au hasard dans l'espace de recherche : la population initiale. Ensuite, on sélectionne les meilleurs éléments en fonction de leur critère d'adaptation, puis on recombine ces éléments dans une nouvelle population, par croisement et mutation. Le processus de sélection/recombinaison est réitéré jusqu'au test d'arrêt. L'arrêt survient lorsqu'on s'estime assez proche de l'optimum, ou bien quand l'optimum trouvé par l'algorithme génétique n'évolue plus significativement, ou encore au bout d'un nombre fixé de générations.

Un des intérêts des algorithmes génétiques est la possibilité de guider la recherche au moyen d'une heuristique dépendant du problème. Les opérateurs de sélection, de croisement et de mutation sont choisis et adaptés en fonction des particularités du problème traité. La difficulté consiste à bien équilibrer la pression sélective qui doit être appliquée aux populations successives, afin de guider la recherche vers une ou plusieurs solutions, tout en préservant la diversité nécessaire pour explorer tout l'espace d'état. La taille de la population, les probabilités de croisement et de mutation, le choix des opérateurs, et aussi le codage des individus, sont des paramètres importants de cet équilibre.

```

 $k \leftarrow 0$ 
Initialiser  $Pop(k)$  aléatoirement
Evaluer  $Pop(k)$ 
Pour chaque génération faire
     $k \leftarrow k + 1$ 
    Sélectionner  $Pop(k)$  à partir de  $Pop(k - 1)$ 
    Recombiner  $Pop(k)$  par croisement et mutation
    Evaluer  $Pop(k)$ 
Fin faire

```

FIG. 3.20 – Principe d'un algorithme génétique.

3.6.2 L'algorithme génétique classique

Le codage des données

Dans l'algorithme initial de Holland, chaque solution éligible est codée par une séquence binaire de longueur fixe, appelée *chromosome*. Ce codage binaire, historiquement le plus ancien, permet d'utiliser des opérateurs de croisement et de mutation simples. Les premiers résultats théoriques sur les algorithmes génétiques ont été obtenus avec ce type de codage, à travers la notion de schéma [Gol89a]. Un schéma est une suite du type $*110**01$, où $*$ peut être indifféremment un 0 ou un 1. L'ordre d'un schéma est son nombre de positions fixes (0 ou 1). Sa longueur fondamentale est la distance séparant la première position fixe de la dernière. Par exemple, $*110**01$ est d'ordre 5 et de longueur fondamentale $8-2=6$.

Un inconvénient majeur du codage binaire est qu'il tient généralement peu compte de la structure du problème. Les différentes informations utiles sont concaténées en une chaîne de bits. Or les résultats théoriques (cf [ASBG02], section 21.3, pour une introduction à la théorie des schémas) montrent que les schémas binaires de longueur fondamentale faible et d'ordre petit sont plus favorisés que les autres lors de la génération d'une nouvelle population. Il faut donc s'efforcer d'avoir un codage qui préserve les informations importantes. Or, dans les problèmes de grande dimension, où les différentes variables sont *a priori* d'égale importance, chaque variable sera codée sur quelques bits de la séquence. L'emplacement de ces quelques bits conditionnera le poids accordé à la variable dans la résolution : les données des variables situées plus près du bord seront plus volatiles au fil des générations que celles proches du centre de la séquence.

Un codage des données par des vecteurs réels permet d'éviter ce genre de problèmes. C'est celui que nous utiliserons par la suite.

La génération de la population initiale

La population initiale doit être suffisamment diversifiée et de taille assez importante pour que la recherche puisse parcourir l'espace d'état dans un temps limité. Si l'on prend comme population initiale des duplicata d'un seul élément, on comprend qu'il faudra le plus souvent un très grand nombre de mutations et de croisements pour progresser vers une solution qui soit un optimum global. C'est pourquoi la population de départ est le plus souvent obtenue par des tirages aléatoires.

Toutefois, il peut être utile d'orienter la génération initiale vers des sous-domaines intéressants de l'espace d'état. Par exemple lors d'une recherche d'optima dans un problème de minimisation sous

contraintes, il est préférable, quand c'est possible, de produire directement des éléments satisfaisant les contraintes. Sinon, il faudra attribuer une pénalisation supplémentaire aux éléments qui violent les contraintes, et adapter la taille de la population et le temps de recherche.

La sélection

La sélection consiste à favoriser statistiquement les bons éléments, au détriment des mauvais. On introduit dans la génération suivante un certain nombre de clones des éléments de la génération courante. Ce nombre dépend du critère d'adaptation de l'élément. Comme la taille de la population reste fixe, les plus mauvais éléments disparaissent, ce qui peut être dommageable, car certains sont porteurs d'informations menant à des solutions intéressantes après divers croisements et mutations. Le processus de sélection doit donc être conçu pour laisser également une chance aux mauvais éléments.

Parmi les nombreux principes existants dans la littérature, celui du *stochastic remainder without replacement selection* a été retenu, notamment parce qu'il n'introduit pas de biais de sélection fonction de la taille de la population. Il se décrit comme suit :

- pour chaque élément i , prendre un nombre de clones égal à la partie entière $E[r_i]$, où $r_i = f(i)/\text{moy}(f)$ est le rapport du critère d'adaptation $f(i)$ sur la moyenne des adaptations. Si on s'en tient là, aucun des éléments dont l'adaptation est au-dessous de la moyenne n'est retenu et la nouvelle population est incomplète,
- compléter la population en tirant aléatoirement, selon le principe de la roulette biaisée (*roulette wheel selection*) le nombre nécessaire d'éléments parmi les individus affectés de coefficients $r_i - E[r_i]$.

Eviter que les meilleurs éléments ne prennent trop rapidement le dessus et garantir le maintien d'une certaine diversité fait l'objet d'améliorations de la sélection (*mise à l'échelle, opérateur de partage*, présentés en 3.6.3 et 3.6.3), d'importance cruciale dans l'utilisation pratique des algorithmes génétiques.

Le croisement

Le croisement s'avère très simple lorsqu'on utilise un codage binaire. Lorsqu'on veut engendrer deux enfants à partir de deux parents, on découpe chaque chromosome parent en deux parties (croisement mono-point) et on recombine les demi-chromosomes pour former les enfants. A travers ce principe très simple, on voit intuitivement que si chaque parent est porteur de "bonnes" et de "mauvaises" informations dans ses gènes (les bits composant le chromosome), on espère qu'un des deux enfants au moins héritera des bons gènes des deux parents. Il s'agit donc à la fois de diversifier la population en recombinant les chromosomes, et de chercher à améliorer l'adaptation des individus.

Dans le cas d'un codage par des vecteurs réels, qui est celui que nous utiliserons dans cette thèse, d'autres principes sont utilisés, comme le croisement barycentrique, qui s'exprime simplement, pour deux parents \vec{p}_1 et \vec{p}_2 , par :

$$\begin{cases} \vec{e}_1 = \alpha\vec{p}_1 + (1 - \alpha)\vec{p}_2 \\ \vec{e}_2 = (1 - \alpha)\vec{p}_1 + \alpha\vec{p}_2 \end{cases}$$

où le coefficient α est choisi aléatoirement, par exemple dans $[-0.5, 1.5]$, si on veut pouvoir produire des enfants à l'extérieur du segment $[\vec{p}_1, \vec{p}_2]$.

Le croisement barycentrique est une possibilité parmi d'autres. L'opérateur doit être adapté au problème traité. Une autre technique de croisement introduite par Nicolas Durand dans [Dur96] donne ainsi d'excellents résultats pour les fonctions partiellement séparables. Cette technique sera présentée dans la section 3.6.3.

La mutation

Dans le but d'obtenir un optimum global, il faut être certain que l'algorithme, moyennant un temps de calcul suffisamment long, peut théoriquement accéder à tout point donné de l'espace d'états. Les opérateurs de sélection et de croisement présentés ci-dessus ne suffisent pas à assurer cette propriété, dite d'ergodicité.

La mutation se fait de la façon la plus simple en modifiant quelques bits d'un chromosome dans le cas du codage binaire, ou en introduisant un bruit aléatoire dans les données codées par des réels. Cet opérateur, et celui de sélection, sont à la base des résultats de convergence des algorithmes génétiques. Certains algorithmes sont d'ailleurs implantés de cette façon, sans opérateur de croisement. Toutefois, afin d'accélérer la convergence pour des problèmes complexes, il est préférable d'utiliser un opérateur de croisement et un opérateur de mutation, de préférence adaptés au problème.

3.6.3 Améliorations du principe de sélection

La mise à l'échelle (*scaling*)

La sélection s'opère en fonction du critère d'adaptation de chaque individu. Mais la distribution des valeurs des critères bruts peut conduire à trop favoriser les meilleurs éléments, ce qui est pénalisant, surtout au début de la recherche. C'est le cas notamment lorsque quelques éléments seulement ont d'excellentes adaptations, et que les autres critères oscillent autour d'une valeur moyenne faible. Pour pallier à cet inconvénient, on effectue une redistribution des valeurs, par une mise à l'échelle linéaire ou exponentielle.

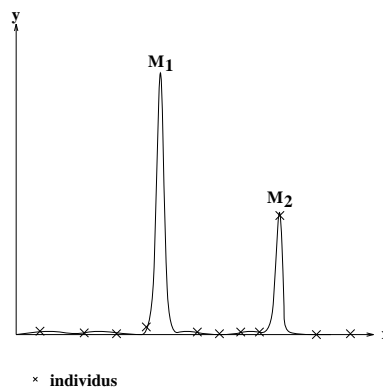


FIG. 3.21 – Exemple où les sélections classiques risquent de ne reproduire qu'un individu.

La mise à l'échelle linéaire (*linear scaling*) consiste simplement à resserrer linéairement les valeurs du critère d'adaptation, par exemple entre 0 et 1. Cette opération est identique pour chaque génération, et l'adaptation après mise à l'échelle (*scaled fitness*) s'écrit $f_s = af_r + b$, où f_r est l'adaptation brute (*raw fitness*). Mais l'idéal consisterait à favoriser la diversité en début de recherche et à faire ressortir les modes dominants en fin de recherche.

Dans la mise à l'échelle exponentielle (*power law scaling*), la redistribution dépend du numéro de génération. Elle consiste à rapprocher les faibles valeurs d'adaptation des meilleures en début de recherche, de façon à augmenter la diversification, et à les en éloigner vers la fin, favorisant une sélection élitiste. La nouvelle adaptation s'écrit $f_s = (f_r)^{c(k)}$, où k est le numéro de génération.

La mise à l'échelle permet une meilleure exploration de l'espace d'états mais ne permet pas à coup sûr, qu'elle soit linéaire ou exponentielle, la survivance des modes sous-optimaux.

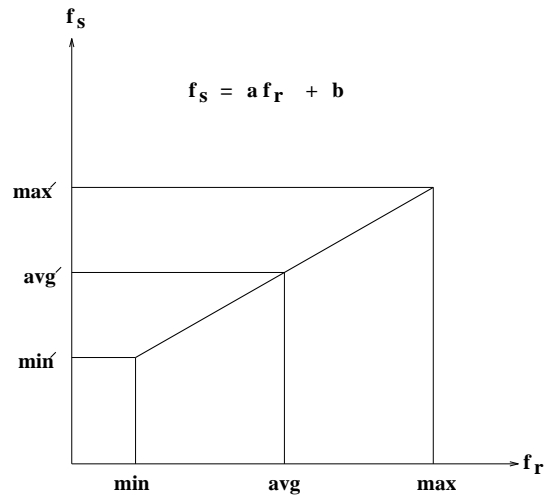
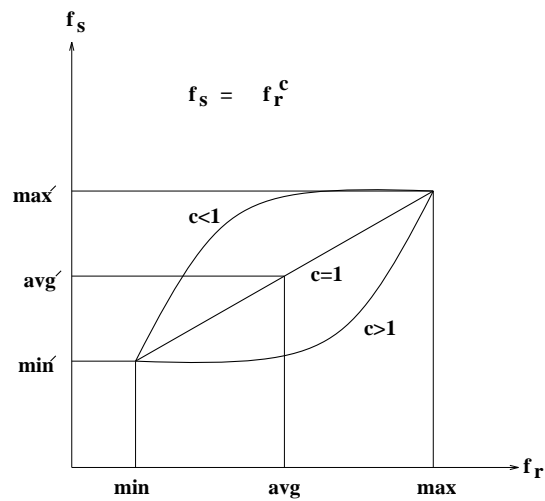
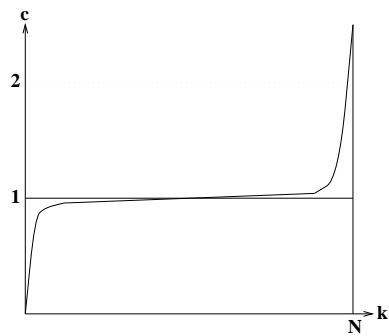


FIG. 3.22 – Fonction de mise à l'échelle linéaire.

FIG. 3.23 – Fonction de mise à l'échelle exponentielle (*power law scaling*).FIG. 3.24 – Allure de l'évolution de c en fonction des générations.

L'opérateur de partage (*sharing*)

L'opérateur de partage modifie les valeurs du critère d'adaptation, comme pour la mise à l'échelle. Le but du partage est de répartir, sur chaque optimum de la fonction objectif, un nombre d'éléments de la population proportionnel au critère d'adaptation estimé de chaque optimum. Cette répartition évite l'agrégation des individus autour du mode dominant (cf figure 3.25).

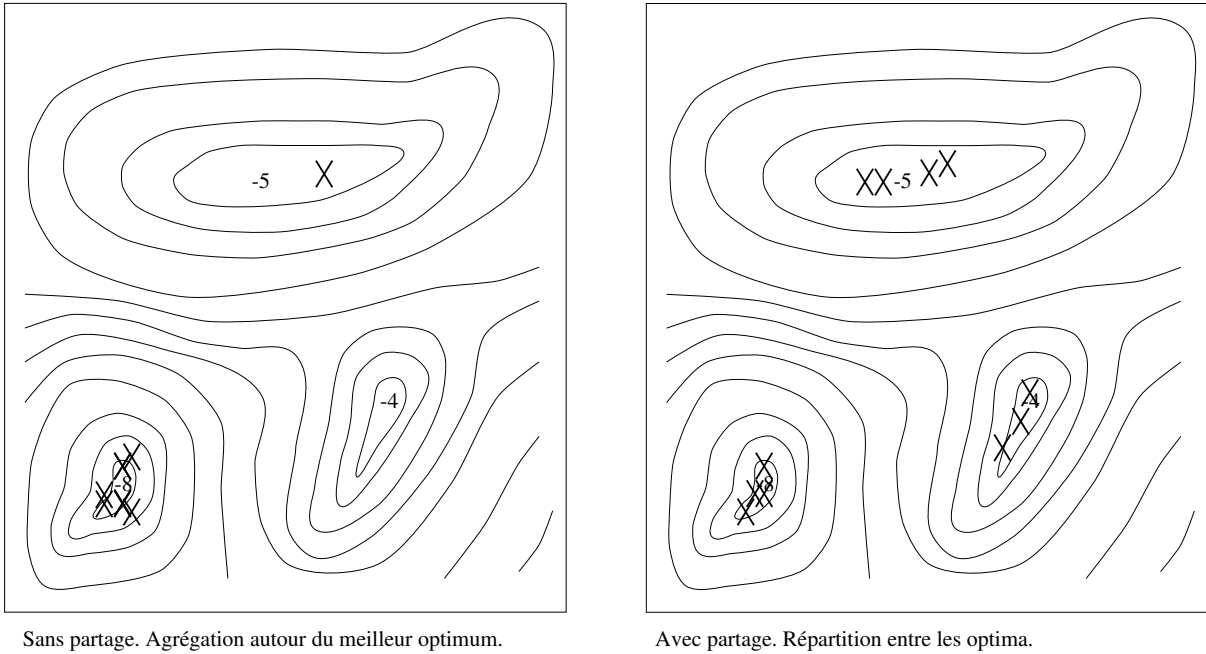


FIG. 3.25 – Illustration du partage sur une fonction à deux variables.

En conséquence, on modifiera les valeurs d'adaptation des éléments selon la densité, le taux d'agrégation de ces éléments dans l'espace d'états. Pour cela, il faut définir une distance d dans l'espace d'états, et un voisinage pour chaque point, délimité par le paramètre σ_{share} . Le taux d'agrégation autour d'un élément x_i est représenté par :

$$m_i = \sum_{j=1}^N S(d(x_i, x_j))$$

où

$$S(d) = \begin{cases} 1 - \left(\frac{d}{\sigma_{share}}\right)^\alpha & \text{si } d \leq \sigma_{share} \\ 0 & \text{si } d > \sigma_{share} \end{cases}$$

N est la taille de la population. Les nouvelles valeurs d'adaptation sont alors données par :

$$f'_i = \frac{f_i}{m_i}$$

Cette modification des valeurs d'adaptation permet de pénaliser les groupes agglomérés, en particulier avec $\alpha < 1$. Ce principe donne de bons résultats mais la complexité du calcul est en $O(N^2)$. Afin de réduire cette complexité, il est intéressant d'identifier les paquets d'éléments agglutinés (*clusters*), et d'utiliser dans le calcul les barycentres, pondérés du nombre d'éléments des agglomérats. C'est le principe du partage par paquets [YG93].

Le partage par paquets (*clusterized sharing*)

L'identification des différents paquets d'éléments agglomérés (*clusters*, fait appel à deux paramètres : d_{min} pour la fusion des agglomérats, et d_{max} pour la décision de création d'un nouvel agglomérat. Initialement, chaque élément de population est considéré comme un agglomérat (contenant donc un seul élément). Le processus d'identification des agglomérats est alors le suivant :

- si deux centres sont à une distance inférieure à d_{min} , on fusionne les paquets d'éléments. Le nouveau centre est le barycentre des deux paquets.
- si un individu est à une distance supérieure à d_{max} de tout centre, on crée un nouveau paquet contenant ce seul individu. Sinon, il est agrégé au paquet le plus proche, dont on recalcule le barycentre.

Une fois les paquets identifiés, le principe du partage est appliqué. Le taux d'agrégation autour d'un élément x_i appartenant à un agglomérat (x_c, n_c) est représenté par :

$$m'_i = n_c \left(1 - \left(\frac{d(x_i, x_c)}{2d_{max}}\right)^\alpha\right)$$

où

- n_c est le nombre d'éléments du paquet auquel appartient x_i ,
- x_c est le barycentre de ce paquet,
- $d(x_i, x_c)$ est la distance entre l'élément x_i et le centre du paquet.

Les nouvelles adaptations sont données par :

$$f'_i = \frac{f_i}{m_i}$$

La complexité de ce principe de partage est en $O(N \log N)$ pour des résultats tout à fait comparables à ceux du partage simple. Toutefois, les valeurs de d_{min} et d_{max} doivent être adaptées à la fonction à optimiser.

3.6.4 Amélioration du principe de croisement

Nous allons présenter ici un principe de croisement adapté aux fonctions partiellement séparables, introduit par Nicolas Durand dans [Dur96]. D'autres types d'améliorations sont également possibles, comme par exemple l'introduction du recuit simulé dans un mécanisme de tournoi entre parents et enfants. Ce dernier principe a été testé pour la résolution de conflits aériens, mais il a donné de moins bons résultats que celui présenté ci-dessous.

Nous avons vu, dans le cas du codage binaire, qu'on espérait notamment du croisement par découpage des chromosomes qu'il produise des individus qui héritent des bons gènes des parents. Cependant, on n'a aucune certitude qu'un tel héritage se produise, et, par ailleurs, cette notion ne se retrouve pas dans le croisement barycentrique de vecteurs réels. Ce dernier peut plutôt s'interpréter comme une exploration de l'espace d'états le long de la droite reliant les deux parents.

L'idée du croisement présenté ici est la suivante : pour pouvoir hériter des "bons" gènes, on mesure ce qu'apporte chaque gène à l'adaptation du chromosome. La notion de *valeur locale d'adaptation* est ici nécessaire. Le principe consistera à comparer, pour chaque gène, les adaptations locales des deux parents et à faire hériter, selon certaines modalités, de la meilleure valeur aux enfants.

Dans le cas du codage par vecteurs réels, on pourra considérer chaque valeur de variable réelle du chromosome comme un gène. Un tel mécanisme est particulièrement utile dans le cas des fonctions

partiellement séparables. Pour fixer les idées, soit à chercher le minimum d'une fonction

$$F : R^n \longmapsto R$$

$$\begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \longmapsto F(x_1, x_2, \dots, x_n)$$

positive partiellement séparable, c'est-à-dire que F peut se mettre sous la forme d'une somme de fonctions positives ne dépendant chacune que d'une partie des variables $(x_1, x_2, \dots, x_n)^2$.

Un chromosome sera ici simplement un vecteur réel (x_1, x_2, \dots, x_n) , auquel on associera un vecteur des adaptations locales $(g(x_1), g(x_2), \dots, g(x_n))$.

Si la fonction est totalement séparable, il suffit de chercher à la minimiser selon chaque variable séparément pour trouver le minimum global. Pour les fonctions partiellement séparables, ce principe reste intéressant à exploiter dans le croisement des chromosomes.

Pour obtenir un fils à partir de deux chromosomes (x_1, x_2, \dots, x_n) et (y_1, y_2, \dots, y_n) , on compare successivement, pour chaque rang i , les valeurs des adaptations locales $g(x_i)$ et $g(y_i)$. En supposant que plus l'adaptation locale est grande, plus la valeur de la variable contribue à minimiser la fonction, on applique le principe suivant :

- si $g(x_i) > g(y_i) + \Delta$ alors le fils hérite de la valeur x_i
- si $g(x_i) < g(y_i) - \Delta$ alors le fils hérite de la valeur y_i
- si $|g(x_i) - g(y_i)| \leq \Delta$ alors on tire aléatoirement la valeur de la variable de rang i du fils (selon le principe du croisement barycentrique par exemple).

Si l'on applique le même principe pour obtenir deux enfants, ceux-ci risquent fort de se ressembler, surtout pour Δ petit. Pour éviter cet inconvénient, on peut prendre une valeur de Δ différente pour chaque fils, ou encore affecter la valeur retenue selon le principe présenté au premier fils, et la valeur non retenue au deuxième, comme dans le croisement classique en codage binaire.

3.6.5 Amélioration du principe de mutation

Dans le cas du codage par vecteur réel, la mutation se fait normalement en choisissant au hasard des chromosomes sur lesquels on bruite un certain nombre de variables. La notion d'adaptation locale introduite pour le croisement adapté est également utilisable pour construire des opérateurs de mutation plus ou moins déterministes, selon la nature du problème traité.

On peut notamment choisir de muter préférentiellement les valeurs des variables ayant la plus mauvaise adaptation locale en essayant "d'accélérer la descente" vers un minimum. Le risque est alors de rester bloqué dans un mode sous-optimal de la fonction. Là encore, la difficulté est de trouver un compromis entre rapidité de convergence et bonne exploration de l'espace d'états.

3.6.6 Optimisation sous contraintes avec des algorithmes évolutionnaires

Nous avons jusqu'à présent présenté les principes et les opérateurs des algorithmes génétiques, dans le cadre de l'optimisation d'un critère d'adaptation, sans imposer de contraintes particulières sur les solutions.

Or, il existe toute une classe de problèmes pour lesquels on recherche des solutions maximisant ou minimisant une fonction objectif, et dont on souhaite qu'elles restent dans un certain domaine de l'espace d'états, ou qu'elles satisfassent certaines conditions. Ces contraintes sur les solutions

²Par exemple, une fonction telle que $F(x_1, x_2, \dots, x_n) = \sum_{i \neq j} (x_i - x_j)^2$ est positive partiellement séparable.

s'expriment sous forme d'une liste d'inégalités $g_i(x) \leq 0$, ($i = 1, \dots, p$), et d'égalités $h_j(x) = 0$ ($j = 1, \dots, q$), où x représente un vecteur de l'espace d'états.

Les algorithmes génétiques, et plus généralement les algorithmes évolutionnaires, permettent de traiter des problèmes d'optimisation globale sous contraintes, selon plusieurs approches. Le lecteur pourra se reporter à [SM96] qui présente clairement différentes façons de gérer les contraintes dans un algorithme évolutionnaire. De façon schématique, disons que les contraintes peuvent être prises en compte soit dans les opérateurs, soit dans le critère d'adaptation, des approches hybrides étant également possibles.

L'utilisation d'opérateurs spécifiquement adaptés pour gérer les contraintes (cf [KM98], [SM98], [SM97]) permet de ne produire que des éléments de population qui soient à l'intérieur du domaine admissible, ou au moins à proximité de la frontière, réduisant ainsi la taille de l'espace de recherche.

L'alternative consiste à pénaliser les solutions violant les contraintes, en leur affectant des valeurs d'adaptation nettement inférieures à celles des solutions qui les satisfont. De cette façon, la pression sélective oriente la recherche vers les solutions situées dans le domaine admissible. Cette manière de procéder est simple à mettre en oeuvre, mais ne permet pas de réduire la taille de l'espace de recherche.

3.6.7 Une explication intuitive du fonctionnement des AG

Une des raisons de la méfiance que peuvent parfois inspirer les algorithmes génétiques est sans aucun doute l'aspect un peu opaque de leur fonctionnement. On introduit des données aléatoires dans la "boîte noire", on croise et on mute plus ou moins au hasard et, ô miracle, on obtient des résultats, avec apparemment un peu de chance.

Comparativement, les méthodes locales, et notamment les méthodes de descente sont plus faciles à appréhender : il s'agit, partant d'un point initial de l'espace d'états, de choisir à chaque itération une direction intéressante, dite *de descente*, et un pas, qui permettent de diminuer la valeur de la fonction dont on cherche le minimum. Ce processus est illustré sur la figure 3.26. En choisissant bien le pas et la direction de descente, on finit par converger vers un minimum. Un des inconvénients de ces méthodes est qu'elles nécessitent, le plus souvent, un certain nombre de conditions de régularité sur la fonction. Mais surtout il faut avoir une bonne idée de l'endroit où se trouve le minimum absolu de la fonction, et choisir le point initial en conséquence, sans quoi on peut parfaitement rester bloqué sur un minimum local.

En utilisant la technique du recuit simulé, on s'autorise occasionnellement à suivre des directions, et à choisir des pas, qui ne diminuent pas la valeur de la fonction (cf figure 3.27). L'idée est de "sauter des bosses" pour voir si la fonction ne descend pas plus loin vers un meilleur minimum. La probabilité de telles actions doit diminuer au fur et à mesure de la recherche (selon une loi de "température de recuit") pour que l'algorithme converge. Avec ce principe, on explore d'abord grossièrement l'espace d'états, puis on focalise progressivement la recherche vers le minimum.

Dans les algorithmes génétiques, illustrés sur la figure 3.28, l'exploration de l'espace d'états est liée à la taille de la population et à la répartition des éléments. La génération de la population initiale joue à cet égard un rôle crucial. La mutation est également un facteur d'exploration de l'espace d'états. Le principe de mutation utilisant les adaptations locales peut s'assimiler, lorsqu'on dispose d'une méthode déterministe d'amélioration de ces adaptations locales, à un mécanisme de descente. La direction et le pas sont donnés par les variations des variables sur lesquelles on opère la mutation. Mais la propriété de "descente" vers des minima est fondamentalement liée pour l'algorithme génétique au mécanisme de sélection, qui élimine les mauvais éléments au cours des générations. Le croisement, s'il n'est pas strictement nécessaire, accélère considérablement la convergence. On peut considérer le

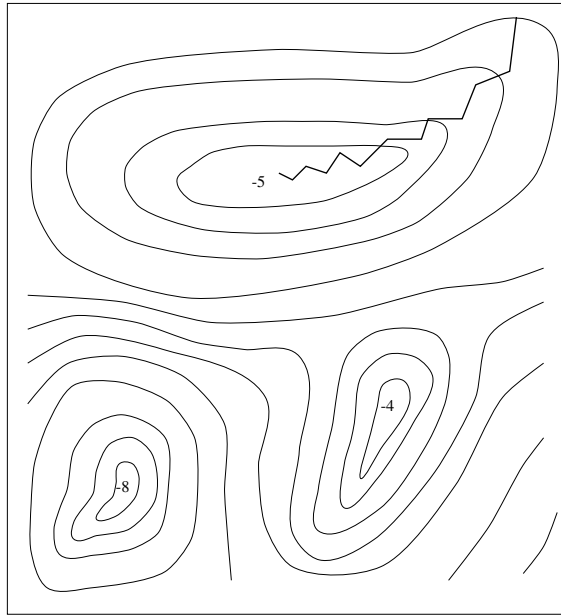


FIG. 3.26 – Recherche de minimum par méthode de descente.

croisement barycentrique comme une exploration (dans le sens d'une recherche aléatoire) sur la droite reliant deux parents. Le croisement adapté consiste plutôt, à partir de deux éléments, à rechercher une direction de descente dans laquelle on produit un fils.

L'avantage des algorithmes génétiques est qu'ils ne requièrent aucune condition sur la nature de la fonction dont on recherche l'optimum, ni sur l'espace de recherche. Toutefois, la convergence se fera d'autant mieux que la fonction aura des propriétés "sympathiques" (continuité par morceaux, convexité, etc...). Par ailleurs, un autre point fort de ce type d'algorithmes est qu'ils sont capables de rechercher *plusieurs* solutions proches de la meilleure solution trouvée.

L'inconvénient majeur de ces algorithmes est que rien ne garantit que la meilleure solution trouvée soit effectivement l'optimum global de la fonction. Par ailleurs, ils sont consommateurs de mémoire (on utilise souvent une population de vecteurs réels de taille assez importante), et *a priori* assez lents dans leur version classique.

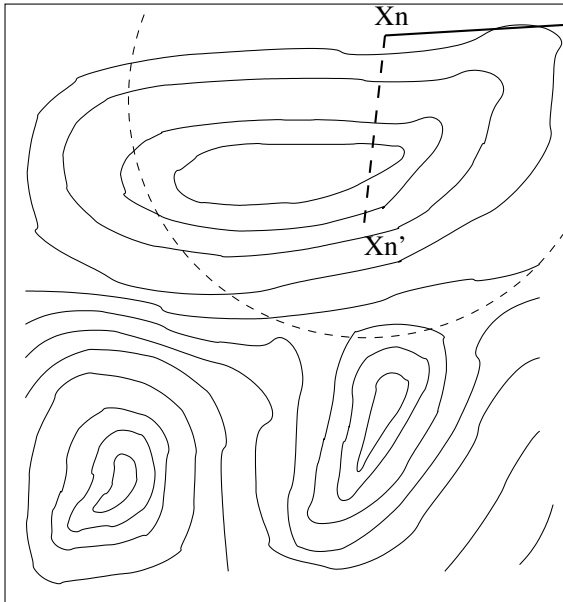
Leur utilisation pratique pour des problèmes complexes passe par l'association avec d'autres méthodes, afin d'améliorer l'heuristique de la recherche d'optimum.

3.7 Conclusion sur la description des algorithmes

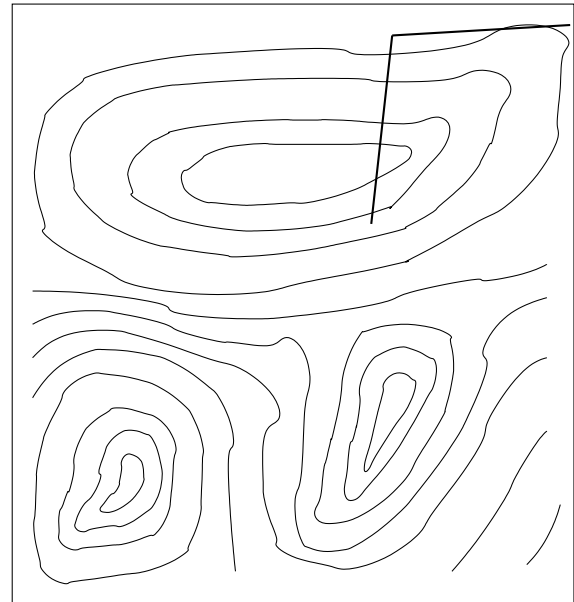
Nous venons de présenter l'ensemble des algorithmes qui seront utilisés dans la suite de la thèse, sans toutefois entrer dans le détail de la façon dont ils seront appliqués à chaque problème particulier.

Notamment, nous n'avons pas indiqué ce que pourraient représenter les éléments de population de l'algorithme génétique, ni fourni la description des opérateurs de croisement et de mutation pour les problèmes que nous allons traiter. Ces éléments seront détaillés dans chaque chapitre concerné.

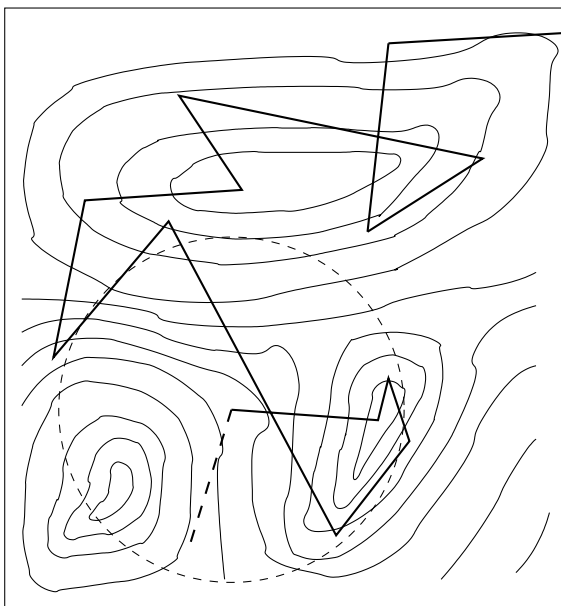
Les chapitres suivants sont consacrés à l'application de ces algorithmes à l'optimisation des flux de trafic aérien, en suivant la démarche décrite dans l'introduction générale. Nous allons donc tout d'abord nous intéresser aux regroupements de secteurs aériens, dans le contexte du système existant.



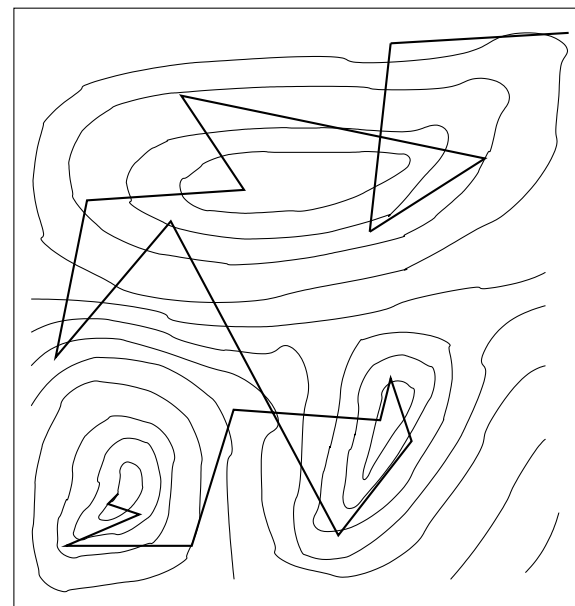
Choix d'un point au hasard dans un voisinage de X_n



Si $f(X_{n'}) < f(X_n)$ ou $\text{rand}() < \exp(-|f(X_{n'}) - f(X_n)|/T)$
alors $X_{n+1} \leftarrow X_{n'}$, sinon on retire X_n au hasard

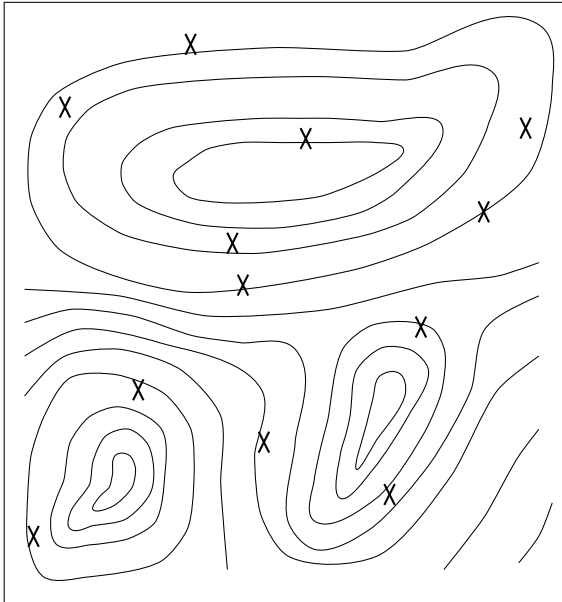


T diminue pendant la recherche qui devient de plus en plus locale

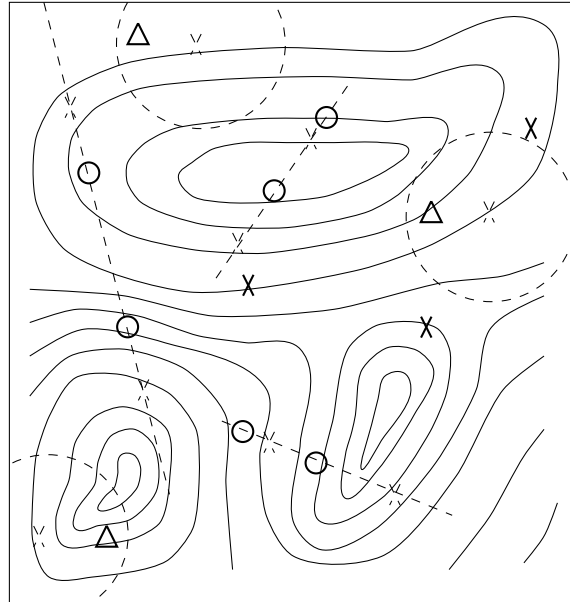


Fin de la recherche

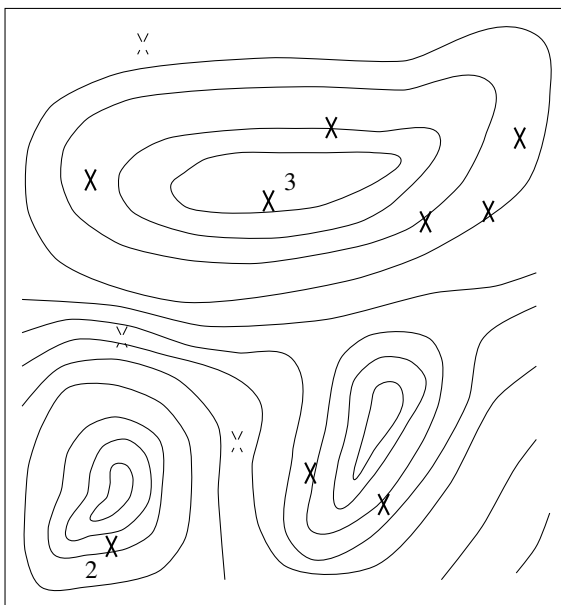
FIG. 3.27 – Recherche de minimum par recuit simulé.



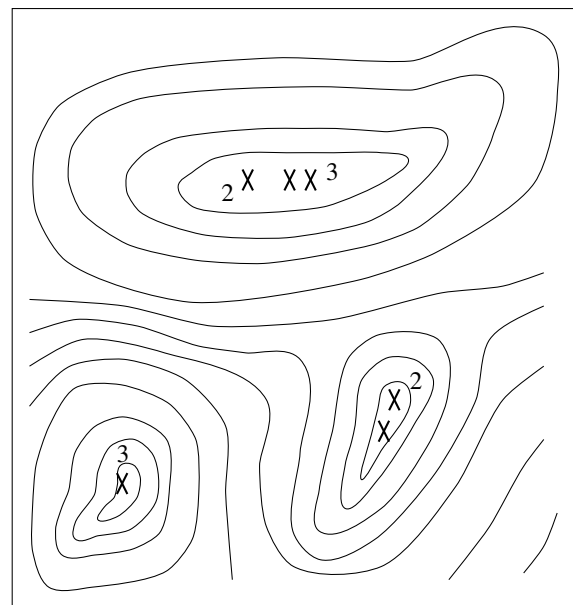
Répartition aléatoire des éléments de la population initiale



Croisements barycentriques et mutations



Sélection des meilleurs éléments au détriment des plus mauvais



A la fin de la recherche, concentration sur les minima (ou sur les surfaces de Pareto)

FIG. 3.28 – Recherche de minima par algorithme génétique.

Chapitre 4

Optimisation des regroupements de secteurs

4.1 Introduction

La première étape du processus ATFM (*Air Traffic Flow Management*) est de définir le schéma d'ouverture des secteurs avec un ou deux jours d'avance, afin de déterminer quelles configurations de secteurs seront les plus adaptées au trafic prévu et si des surcharges sont prévisibles.

En Europe, les schémas d'ouvertures sont élaborés par les FMP (*Flow Management Position*) de chaque centre de contrôle. Pour ce faire, des outils automatisés sont utilisés par les opérateurs : par exemple en France, un opérateur FMP peut choisir une configuration parmi un ensemble pré-défini et la comparer au trafic prévu afin de faire ressortir les dépassements de capacité. Mais en dehors de sa propre expérience, il n'a aucun moyen de savoir si la configuration choisie est la mieux adaptée ou si une autre aurait permis de mieux équilibrer le trafic entre les positions de contrôle. Un inconvénient supplémentaire de la méthode de travail actuelle est qu'elle se limite à un ensemble de configurations pré-définies qui ne représente qu'une toute petite part de toutes les configurations possibles, comme nous le verrons plus loin.

Le présent chapitre décrit deux algorithmes de recherche dans des arbres et un algorithme génétique permettant de construire des configurations de secteurs optimales en fonction des flux de trafic, des contraintes de capacité des secteurs, et éventuellement du nombre maximal de positions de contrôle qu'il est possible d'armer à chaque heure de la journée. Ces algorithmes sont appliqués à un contexte réaliste, en utilisant les données de trafic, de description de l'espace aérien, de nombre de positions, et de capacités enregistrées par les centres de contrôle français.

4.2 Description du problème

Le problème de l'optimisation des configurations de secteurs a déjà été abordé dans [DASF95] (au sein du LOG), dans un contexte peu réaliste, et aussi dans [VM01] (division RFM du CENA). Dans [DASF95], un algorithme génétique est utilisé sur un modèle très simplifié d'environnement ATC, pour équilibrer les charges de travail en regroupant des secteurs. Seuls des secteurs convexes sont envisagés et le nombre de positions est une donnée fixée en entrée.

Dans [VM01], des techniques de programmation sous contraintes et de programmation linéaire en nombres entiers sont utilisées dans un contexte réaliste afin de minimiser la somme des dépassements de capacité en sélectionnant des *patterns* parmi un ensemble de configurations pré-définies. Un *pattern*

est une configuration associée à une période de temps. Le nombre de positions est également un paramètre fixé en entrée, comme dans [DASF95]. Une tentative est faite de considérer le trafic comme une variable, en autorisant des déplacements macroscopiques de charges de trafic le long de l'axe du temps ou d'un secteur à un autre. La convergence de l'algorithme itératif employé dans ce cas n'est pas assurée. Cependant, les résultats sont encourageants au regard de la méthode de travail actuelle.

Dans ce chapitre, nous abordons un problème d'optimisation sensiblement différent : l'optimum que l'on cherche à atteindre est la configuration pour laquelle le trafic est au plus près de la capacité, pour chaque secteur ou groupe de secteurs de la configuration. Les surcharges, mais également les sous-charges, sont donc considérées. Par ailleurs, nous ne nous restreindrons pas, dans la recherche d'une configuration optimale, à un sous-ensemble pré-défini : les algorithmes exploreront l'ensemble des configurations que l'on peut obtenir en combinant des secteurs opérationnels. Le nombre de positions de contrôle est une variable de la fonction de coût que l'on cherche à minimiser. Ce nombre est éventuellement contraint par une limite supérieure, sachant qu'il n'y a pas toujours suffisamment de contrôleurs pour armer toutes les positions nécessaires.

4.3 Modélisation

Une "bonne" configuration est une configuration pour laquelle il n'y a pas de dépassements de capacité (ou les plus petits possibles), et pour laquelle la charge de trafic est aussi équilibrée que possible entre les positions de contrôle, tout en armant le moins possible de positions dans un souci d'efficacité. Ceci peut se formuler comme un problème de minimisation.

Définissons une fonction Δ par

$$\Delta(x, t) = workload(x, t) - capacity(x, t)$$

où x est un secteur ou un groupe de secteurs, t est le temps, $workload$ est la charge de trafic (par exemple N avions entrant dans le secteur x entre t et $t + w$, où w est un intervalle de temps choisi), et $capacity$ la valeur seuil de la charge de trafic. La capacité de chaque secteur peut également dépendre de critères spécifiques (activité militaire, etc...)

L'opérateur peut autoriser certaines tolérances autour de la valeur nominale de la capacité. Ces tolérances inférieure l et supérieure u sont prises en compte dans l'évaluation d'une configuration. A cette fin, définissons les surcharges et sous-charges comme suit :

$$\Delta_{++}(x, t) = \begin{cases} \Delta(x, t) & \text{si } \Delta(x, t) > u \\ 0 & \text{sinon} \end{cases}$$

$$\Delta_{+}(x, t) = \begin{cases} \Delta(x, t) & \text{si } 0 \leq \Delta(x, t) \leq u \\ 0 & \text{sinon} \end{cases}$$

$$\Delta_{-}(x, t) = \begin{cases} |\Delta(x, t)| & \text{si } l \leq \Delta(x, t) \leq 0 \\ 0 & \text{sinon} \end{cases}$$

$$\Delta_{--}(x, t) = \begin{cases} |\Delta(x, t)| & \text{si } \Delta(x, t) < l \\ 0 & \text{sinon} \end{cases}$$

Définissons alors les fonctions suivantes :

$N_{pos}(t)$ le nombre de positions de contrôle de la configuration

$$C_{++}(t) = \sum_{x \in config} (\Delta_{++}(x, t))^2$$

$$C_{+}(t) = \sum_{x \in config} \Delta_{+}(x, t)$$

$$C_{-}(t) = \sum_{x \in config} \Delta_{-}(x, t)$$

$$C_{--}(t) = \sum_{x \in config} (\Delta_{--}(x, t))^2$$

Le problème consiste alors à minimiser la fonction de coût suivante :

$$cost_{config} = a.C_{++} + b.N_{pos} + c.C_{--} + d.(C_{+} + C_{-}) \quad (4.1)$$

tout en respectant la contrainte : $N_{pos}(t) \leq M_{pos}(t)$ où $a, b, c,$ et d sont des coefficients choisis d'ordre décroissant et où M_{pos} est le nombre maximum de positions de contrôle disponibles à chaque heure du jour.

En pratique, et afin de mieux affirmer le poids relatif des différents coûts, plutôt que de minimiser $cost_{config}$, on maximisera $eval_{config}$ telle que les k_1 digits de poids fort de $eval_{config}(x, t)$ soient relatifs au coût C_{++} , les k_2 suivants à N_{pos} , et ainsi de suite :

$$\begin{array}{cccc} \underbrace{xxxxx}_{k_1} & \underbrace{xx}_{k_2} & \underbrace{xxxxx}_{k_3} & \underbrace{xxx}_{k_4} \\ eval_{config} = & 10^{k_2+k_3+k_4} \times N(k_1, C_{++}) & & \\ & +10^{k_3+k_4} \times N(k_2, N_{pos}) & & \\ & +10^{k_4} \times N(k_3, C_{--}) & & \\ & +N(k_4, C_{+} + C_{-}) & & \end{array} \quad (4.2)$$

où N est une fonction telle que

$$N(k, C) = \lfloor \max(0, (10^k - 1) - C) \rfloor$$

et les exposants $k_1, k_2, k_3,$ et k_4 sont choisis si possible de façon que le terme $(10^k - 1)$ soit un majorant du coût correspondant.

4.4 Difficulté du problème

La difficulté du problème tient essentiellement à la combinatoire du nombre de configurations que l'on peut construire à partir d'un ensemble de secteurs. Une configuration est une répartition de n secteurs sur k positions de contrôle.

Cherchons le nombre de partitions un ensemble de n éléments en k sous-ensembles. Si $P(n, k)$ est ce nombre, il vérifie les relations suivantes, comme nous l'avons déjà vu au chapitre 3, dans la section 3.3 présentant les méthodes de classification :

$$\begin{array}{l} \forall n \geq 1 \\ P(n, 1) = 1 \quad (\text{un seul groupe de } n \text{ éléments}) \\ P(n, n) = 1 \quad (\text{une seule partition avec } n \text{ groupes d'un élément}) \\ P(n, k) = 0 \text{ if } k > n \quad (\text{on ne peut pas faire plus de } n \text{ groupes}) \\ P(n, k) = k * P(n - 1, k) + P(n - 1, k - 1) \text{ if } 1 < k < n \end{array}$$

Le nombre de partitions possibles sera alors :

$$P(n) = \sum_{k=1}^n P(n, k)$$

D'un point de vue algorithmique, l'ensemble des partitions peut théoriquement être obtenu en explorant l'arbre des possibilités, quoique cela devienne rapidement infaisable en pratique.

Pour 17 secteurs (Brest ATCC), la formule $P(17)$ donne environ 83 milliards de possibilités. Cependant, cette valeur n'est pas réaliste : bon nombre des partitions trouvées ne pourraient pas être utilisées dans un contexte opérationnel. Par exemple, une partition contenant un groupe dans lequel un des secteurs n'est voisin d'aucun autre secteur du groupe n'est pas opérationnellement exploitable.

Afin d'estimer la difficulté du problème réel, considérons seulement les secteurs et groupes de secteurs opérationnels définis dans les bases de données des centres de contrôle. L'ensemble des configurations opérationnelles possibles sera obtenu en parcourant un arbre dont les noeuds sont des listes de couples (g, \mathcal{G}) (cf figure 4.1 illustrant le Branch&Bound), où g est un groupe en construction et \mathcal{G} est l'ensemble des groupes opérationnels compatibles avec g , dans le contexte de la configuration en construction. Un élément h de \mathcal{G} est "compatible" avec g s'il contient tous les secteurs de g , mais aucun secteur des autres groupes de la configuration (les autres " g " du noeud). Si un des ensembles \mathcal{G} est vide, il est inutile de continuer l'exploration à partir du noeud considéré : elle ne mènera à aucune configuration valide.

	Nombre de secteurs		Nombre de partitions	Nombre de configs
	éléments	groupes		
Aix	24	42	4.4610 ¹⁷	123 965
Bordeaux	22	65	4.4510 ¹⁵	551 032
Brest	17	52	8.2810 ¹⁰	14 832
Paris (Ouest)	11	17	678 570	192
Paris (Est)	12	22	4 213 597	399
Reims	12	17	4 213 597	249

TAB. 4.1 – Nombre de partitions et de configurations possibles pour les centres de contrôle français (données 1999).

Les méthodes présentées ci-dessus nous permettent de compter le nombre de configurations et d'estimer la difficulté du problème. Les résultats pour les centres français sont présentés dans le tableau 4.1, qui souligne bien la relation combinatoire entre le nombre de secteurs et le nombre de partitions et de configurations.

Le nombre de configurations opérationnelles que l'on peut construire à partir des secteurs et groupes de secteurs décrits dans les données des centres va de quelques centaines à plus d'un demi-million. Nous pouvons donc espérer qu'une méthode déterministe d'exploration d'arbre nous mène à des configurations optimales en un temps de calcul raisonnable, si l'on s'en tient aux centres considérés et aux groupes de secteurs opérationnels définis.

Cependant, comme ce problème d'optimisation pourrait être envisagé pour de plus grands centres ou avec un plus grand ensemble de combinaisons de secteurs, nous allons également tester un algorithme génétique, afin de pouvoir trouver des configurations optimales ou presque optimales en un temps borné.

4.5 Algorithmes déterministes

Un algorithme basique : Dans le chapitre 3, section 3.3, nous avons décrit un algorithme construisant toutes les partitions possibles d'un ensemble, que l'on peut utiliser directement pour calculer les configurations opérationnelles. Une méthode basique d'optimisation consiste à évaluer le coût (cf définition 4.1) de chacune de ces configurations, afin de trouver la ou les meilleures d'entre elles. Cette méthode est consommatrice de temps de calcul et de mémoire, mais elle assure l'optimalité des solutions trouvées. Elle sera donc utilisée comme référence pour l'évaluation des autres algorithmes présentés ci-dessous, lorsque c'est possible.

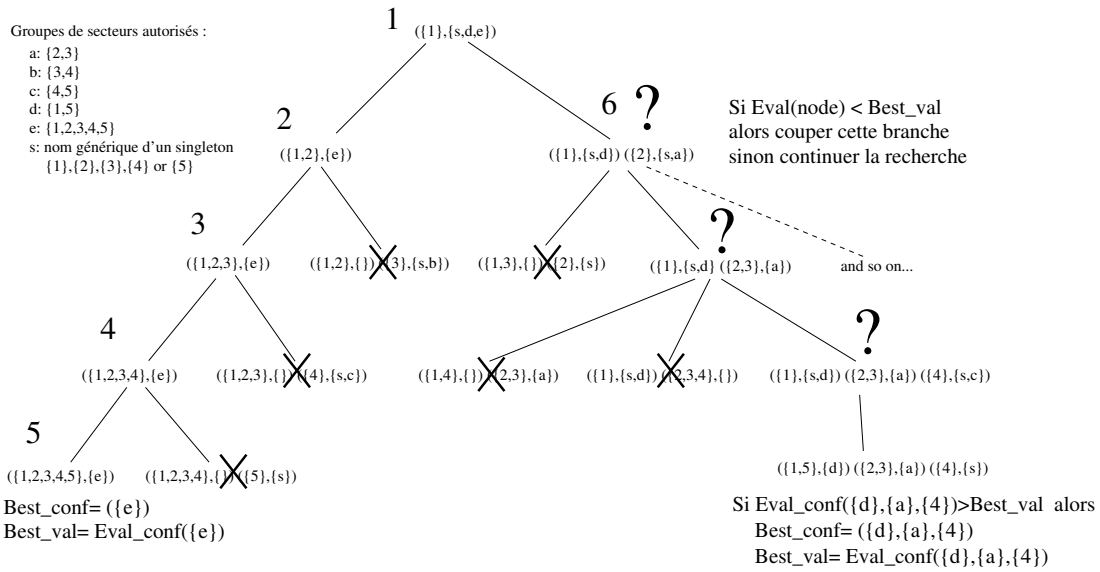


FIG. 4.1 – Branch & bound pour la recherche d'une configuration optimale

Un algorithme de Branch & bound : L'idée du branch & bound illustré sur la figure 4.1 est d'éviter l'exploration de tout l'arbre, en s'interdisant de suivre certaines branches dont on sait qu'elle ne mèneront pas à une meilleure solution que celle qu'on connaît déjà. Il est alors nécessaire d'évaluer les noeuds pendant le parcours. Le coût associé à un noeud doit être un minorant du coût de toutes les configurations que l'on peut atteindre à partir de ce noeud. Un noeud de notre algorithme est une configuration en construction (cf figure 4.1), c'est-à-dire une liste de couples (g, \mathcal{G}) , où g est un groupe en construction et \mathcal{G} est l'ensemble des groupes opérationnels compatibles avec g dans le contexte de la configuration en construction.

Pour une meilleure compréhension de cette notion de compatibilité, considérons le noeud 6 de l'exemple de la figure 4.1. Ce noeud est représenté par : $(\{1\}, \{s, d\}) (\{2\}, \{s, a\})$. Les groupes compatibles avec le groupe en construction $\{1\}$ sont le singleton s (représentant le secteur $\{1\}$ seul), et le groupe $d = \{1, 5\}$. Ces groupes sont les seuls contenant $\{1\}$ mais pas $\{2\}$.

La fonction de coût pour un noeud est analogue au coût d'une configuration. Soit $best$ une fonction telle que $best(\mathcal{G})$ retourne l'élément h (un secteur ou groupe de secteurs) de \mathcal{G} pour lequel la différence $workload(h, t) - capacity(h, t)$ est la plus petite sous-charge possible, ou alors la plus petite surcharge possible, si tous les éléments de \mathcal{G} sont en surcharge. Le coût d'un noeud est alors défini comme dans la définition 4.1, si ce n'est que N_{pos} est le nombre de couples (g, \mathcal{G}) du noeud et que seules les fonctions Δ_{++} et Δ_+ sont considérées, en prenant

$best(\mathcal{G})$ comme paramètre, au lieu d'un groupe opérationnel x d'une configuration.

Un algorithme inspiré de l' A^* : Comme le *Branch & bound*, l' A^* est un algorithme de recherche dans un arbre. Cependant, au lieu de mémoriser simplement la meilleure feuille de l'arbre trouvée pendant la recherche, il mémorise les noeuds parcourus en leur affectant une priorité dépendant du coût estimé pour atteindre l'optimum. A cet effet, l' A^* tel que décrit dans [Pea84] nécessite une fonction calculant le coût de chaque transition d'un noeud à l'autre, et une fonction, appelée heuristique, estimant au mieux (en le minorant) le coût des transitions restantes entre le noeud courant et la fin de la recherche.

Pour notre problème, nous cherchons à minimiser la fonction $cost_{config}$ décrite dans la définition 4.1, en cherchant un chemin dans l'arbre illustré dans la figure 4.1. Mais au lieu de comparer l'évaluation d'un noeud avec celle de la meilleure configuration trouvée comme dans le *Branch & bound*, les noeuds sont mémorisés dans une file d'attente à priorité (*priority queue*), ordonnée selon le coût des noeuds. L' A^* considère itérativement le noeud de coût le plus faible, évalue les noeuds fils et les insère dans la queue, jusqu'à ce qu'une configuration complète soit atteinte. Cette configuration est alors optimale.

4.6 Algorithme génétique

Nous avons également testé un algorithme génétique pour résoudre le problème d'optimisation des configurations de secteurs. Cet algorithme considère une population de chromosomes qui évolue par croisement, mutation, et sélection des individus les plus adaptés, comme décrit dans [Gol89a] et [Mic92].

Une configuration est donc considérée ici comme un chromosome composé de plusieurs gènes. Un gène est un secteur ou un groupe de secteurs. Les chromosomes sont évalués selon leur adaptation (*fitness*). La valeur brute de l'adaptation d'une configuration est $f = eval_{config}$ (voir définition 4.2).

Un opérateur de partage, dit *clusterized sharing*, est appliqué aux valeurs brutes de l'adaptation. Le but de cet opérateur est d'éviter qu'un chromosome ayant une bonne adaptation ne se reproduise au détriment des autres chromosomes, focalisant ainsi la recherche sur un seul mode optimal ou sous-optimal. Le *clusterized sharing* modifie les valeurs d'adaptation en fonction de la densité de chromosomes, en identifiant des agrégats (*clusters*) dont on souhaite qu'il évoluent chacun vers un optimum local. Le processus d'agrégation est basé sur un critère de distance entre chromosomes. Pour notre problème, la difficulté du partage consiste à définir une distance entre configurations de secteurs. De façon analogue à la distance de Hamming qui comptabilise les différences entre les gènes de deux chromosomes, lorsque le nombre de gènes est une constante fixée, la pseudo-distance choisie pour les configurations se base sur une comparaison des gènes, qui peuvent toutefois être en nombre différent selon les configurations. Soit n_i (respectivement n_j) le nombre de gènes du chromosome i (respectivement j). La pseudo-distance choisie est définie par $d(i, j) = \min(n_i, n_j) - n$ où n est le nombre de gènes identiques entre les deux chromosomes.

Les valeurs d'adaptation après partage sont ensuite mises à l'échelle (*sigma truncation*) afin d'atténuer les différences entre les adaptations des bons chromosomes et des mauvais. Cette mise à l'échelle laisse une meilleure chance de se reproduire aux mauvais chromosomes, et permet une exploration plus large de l'espace d'états.

L'opérateur de croisement coupe en deux les chromosomes parents et complète chaque morceau (une configuration incomplète) avec les gènes de l'autre parent compatibles avec la configuration incomplète. Une fois tous les gènes possibles de l'autre parent ainsi utilisés, il faut généralement encore compléter la configuration en choisissant au hasard des secteurs ou groupes de secteurs com-

patibles. Une fonction d'adaptation locale (*local fitness*) permet d'assigner une valeur d'adaptation à chaque gène. Elle est utilisée pour sélectionner les gènes des parents en choisissant ceux dont l'adaptation locale est la meilleure. Il est démontré dans [DA98] qu'un opérateur de croisement utilisant des adaptations locales accélère la convergence dans l'optimisation de fonctions à variables partiellement séparables. Dans notre problème, l'adaptation locale est définie de façon analogue à l'adaptation d'un chromosome, en ne tenant compte des surcharges et sous-charges que du seul gène considéré.

L'opérateur de mutation choisit d'abord un premier gène au hasard. Puis un deuxième choix est effectué dans une liste comprenant le premier gène choisi et ses voisins (les secteurs ou groupes de secteurs de la configuration ayant une frontière commune avec le premier gène). Les secteurs des gènes ainsi choisis sont alors recombinaisonnés en plusieurs nouveaux gènes (jusqu'à trois). Ce choix de deux gènes au lieu d'un seul se justifie par le fait que la recombinaison d'un seul groupe se traduirait par un ou plusieurs "dégrouperments" (voir glossaire) menant ainsi à des configurations plus éclatées. Ce type de mutation n'aurait aucune chance de faire évoluer la population vers des configurations plus regroupées.

4.7 Résultats

Une interface graphique a été développée afin de visualiser les résultats des optimisations. Les langages Ocaml et OcamlTk ont été utilisés pour coder respectivement les algorithmes et l'interface. Le programme opère sur un PC Pentium IV (1.8 GHz) avec Linux comme système d'exploitation.

Les paramètres d'entrée sont la date, le centre de contrôle choisi, le type de trafic (demande initiale, finale, trafic réalisé), les tolérances sur la capacité, le nombre maximum de positions de contrôle, le pas de calcul et la fenêtre de temps pour le calcul des flux.

Le programme affiche une configuration optimisée pour chaque pas de calcul. Le code de couleur est : Vert (gris) quand la charge est en-dessous de la capacité moins la tolérance inférieure, Jaune (blanc) quand la charge est dans les limites de tolérance de la capacité, Rouge (gris foncé) quand la charge est au-dessus de la capacité plus la tolérance supérieure.

Le haut de la figure 4.2 montre le résultat d'une optimisation pour le centre de contrôle de Bordeaux, avec en entrée la demande de trafic initiale et aucune contrainte sur le nombre maximal de positions de contrôle. Dans ce cas, les secteurs surchargés ne peuvent être que des secteurs élémentaires ou des groupes de secteurs ne pouvant pas être dégroupés¹.

Le bas de la figure 4.2 montre la même optimisation en ajoutant des contraintes en nombre maximum de positions disponibles. Ces contraintes sont prises en compte dans les fonctions d'évaluation en diminuant fortement la valeur d'évaluation lorsque le nombre de positions de contrôle de la configuration est supérieur au maximum disponible. Les contraintes de l'exemple de la figure 4.2 sont issues directement des schémas déposés par le centre de Bordeaux ce jour-là. Ces contraintes induisent des dépassements de capacités supplémentaires (entre 4 et 5 heures du matin sur l'exemple) sur des regroupements de secteurs que l'algorithme ne peut dégroupier par manque de positions disponibles.

On constate, en comparant ces deux schémas, que Bordeaux était sous-capacitif aux premières heures du matin, et sur-capacitif le reste de la journée.

¹certaines secteurs élémentaires ne peuvent être armés seuls. Ils sont définis et utilisés uniquement dans un objectif de flexibilité, en choisissant la façon de les combiner avec d'autres secteurs.

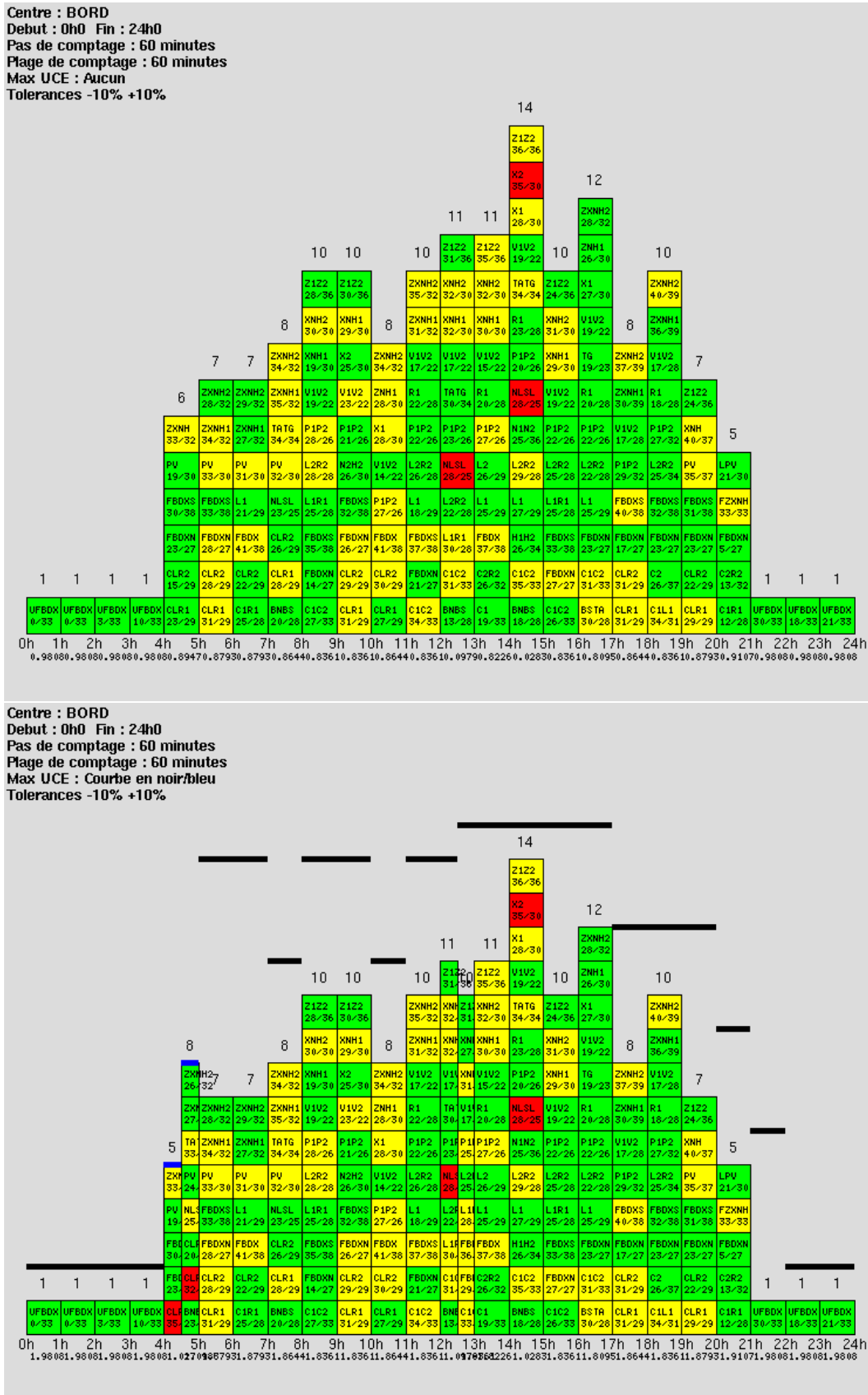


FIG. 4.2 – Configurations optimales avec (en bas) ou sans (en haut) contraintes sur le nombre maximal de positions disponibles (exemple de Bordeaux ACC).

4.7.1 Comparaison des algorithmes

Les algorithmes classiques fournissent des solutions optimales pour chaque centre, vérifiées en utilisant l'algorithme basique lorsque c'est possible (pour Brest, Paris Ouest, Paris Est, et Reims). Le tableau 4.2 fait état des temps de calcul pour les algorithmes déterministes. Le plus rapide est le *branch & bound*. L' A^* est plus lent, car l'heuristique choisie n'est pas suffisamment proche du coût réel des transitions menant à la configuration optimale, ce qui induit un fort taux de retour-arrière (*backtracking*).

	Aix	Bordeaux	Brest	Paris(E)	Paris(O)	Reims
Basique	n.t.	n.t.	14.44	0.06	0.03	0.02
B&B	34.06	6.06	1.45	0.08	0.07	0.04
A^*	134.88	10.12	10.05	0.08	0.04	0.02

TAB. 4.2 – Temps de calcul (en secondes) pour les algorithmes déterministes.

Bien qu'il y ait plus de configurations possibles pour Bordeaux que pour Aix (voir tableau 4.1 sur la difficulté du problème), le temps de calcul pour Aix est plus important, étant donné qu'il y a plus de secteurs à Aix qu'à Bordeaux. L'arbre des possibilités est exploré avec une stratégie de recherche en profondeur, et la profondeur de l'arbre dépend du nombre de secteurs.

Pour l'algorithme génétique, 10 valeurs initiales du générateur aléatoire ont été testées pour Bordeaux, avec une probabilité de croisement de 0.6 et une probabilité de mutation de 0.2. Le tableau 4.3 fait état pour chaque pas de temps du nombre de configurations différentes de l'optimum. Les lignes (+1) montrent le nombre d'occurrences où la configuration trouvée comprenait exactement une position de plus que la configuration optimale. Les lignes (> +1) montrent le nombre d'occurrences où la différence est supérieure à une position de contrôle. Lorsqu'elles ne sont pas optimales, les solutions trouvées sont qualitativement proches de l'optimum.

Heure		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
220 gen. 120 elem.	Echecs	0	0	0	0	0	0	0	0	2	0	1	5	2	0	1	3	0	1	0	0	1	0	0	0
	+1	0	0	0	0	0	0	0	0	0	0	0	2	0	0	1	0	0	1	0	0	1	0	0	0
	> +1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
220 gen. 130 elem.	Echecs	0	0	0	0	0	0	0	0	3	0	0	3	3	0	2	2	0	0	0	0	1	0	0	0
	+1	0	0	0	0	0	0	0	0	0	0	1	0	0	2	0	0	0	0	0	0	1	0	0	0
	> +1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
300 gen. 220 elem.	Echecs	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
	+1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	> +1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TAB. 4.3 – Résultats de l'algorithme génétique pour Bordeaux ATCC.

L'algorithme génétique est plus lent que les méthodes de recherche dans des arbres, mais fournit plusieurs solutions optimales ou presque optimales. Il n'est pas très significatif de comparer le temps de calcul de l'algorithme génétique avec les temps de calcul des algorithmes déterministes lorsque le problème est facile. Dans de tels cas (la nuit par exemple), l'algorithme génétique effectuera le même nombre de générations quelle que soit la difficulté du problème, alors que le B&B trouvera la solution optimale parmi les toutes premières branches explorées. Le tableau 4.4 compare pour Bordeaux les temps de calcul pour chaque pas de temps entre 5h et 19h pour 220 générations et 120 éléments.

Heure	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Temps B&B (s.)	0.23	0.05	0.17	0.55	0.95	0.21	0.38	0.54	0.21	0.66	0.59	0.58	0.35	0.56
Temps GA (s.)	15.69	14.88	17.19	16.08	15.37	15.80	15.75	16.42	15.62	17.41	16.85	15.65	14.98	13.70

TAB. 4.4 – Comparaison détaillée des temps de calcul de l'AG et du B&B pour Bordeaux ATCC

4.7.2 Comparaison entre schéma optimal et schéma déposé

Nous avons vu sur la figure 4.2 que, même sans contrainte sur le nombre de positions, il pouvait rester des secteurs élémentaires surchargés lorsqu'on considérait en entrée la demande initiale (brute) de trafic. Ceci peut être évité en lissant d'abord le trafic en considérant que tous les secteurs élémentaires sont armés. Comparons les stratégies suivantes en termes de délais produits et de ressources utilisées, pour un jour de pointe de 1999 (le 21 mai) :

DEP : allocation de délais sur la base des schémas déposés par les centres de contrôle français, modifiés en fonction des capacités enregistrées dans les bases de données des centres, lorsque celles-ci sont différentes des capacités déclarées dans les schémas déposés.

OPT.0.0 : il consiste en deux étapes, la première étant une allocation de délais avec tous les secteurs élémentaires armés, la seconde étant l'optimisation du schéma d'ouverture des secteurs en prenant en entrée le trafic lissé par la première étape.

	Retards (en minutes)	
	Total	Max
DEP	211335	460
OPT.0.0	65505	225
Gain (%)	69 %	

TAB. 4.5 – Retards au décollage engendrés par les différentes stratégies

	Temps d'activité cumulé des positions de contrôle (en minutes)						
	Aix	Bordeaux	Brest	Paris(E)	Paris(O)	Reims	Total
DEP	19230	14160	14700	9600	9150	10500	77340
OPT.0.0	18240	11400	8520	7020	7620	9120	61920
Gain (%)	5.1	19.5	42	26.9	16.7	13.1	19.9

TAB. 4.6 – Ressources nécessaires selon la stratégie choisie

Il faut être prudent dans l'analyse des résultats : en particulier les valeurs des capacités peuvent être légèrement différentes entre le schéma déclaré et les capacités enregistrées dans les bases de données des centres. Ceci est dû à plusieurs facteurs : en 1999, les capacités n'étaient pas enregistrées régulièrement et le jeu de données utilisé ici peut ne pas correspondre à la journée choisie. Par ailleurs, pour l'élaboration du schéma déposé, l'opérateur choisit parmi plusieurs valeurs de capacité en fonction du contexte du jour (activité militaire, etc...). Quoi qu'il en soit, tout ceci ne change pas la nature et la validité des résultats présentés.

Une méthode ("SHAMAN") d'allocation de délais par programmation par contraintes a été utilisée. Elle minimise le délai maximum. Pour plus de détails sur cette méthode, le lecteur peut se référer à [BBR01]. Les allocations de créneaux, comme les optimisations de schémas, se font sur la base d'un pas de calcul de 60 mn et d'une fenêtre temporelle de 60 mn pour le calcul des flux, et avec des

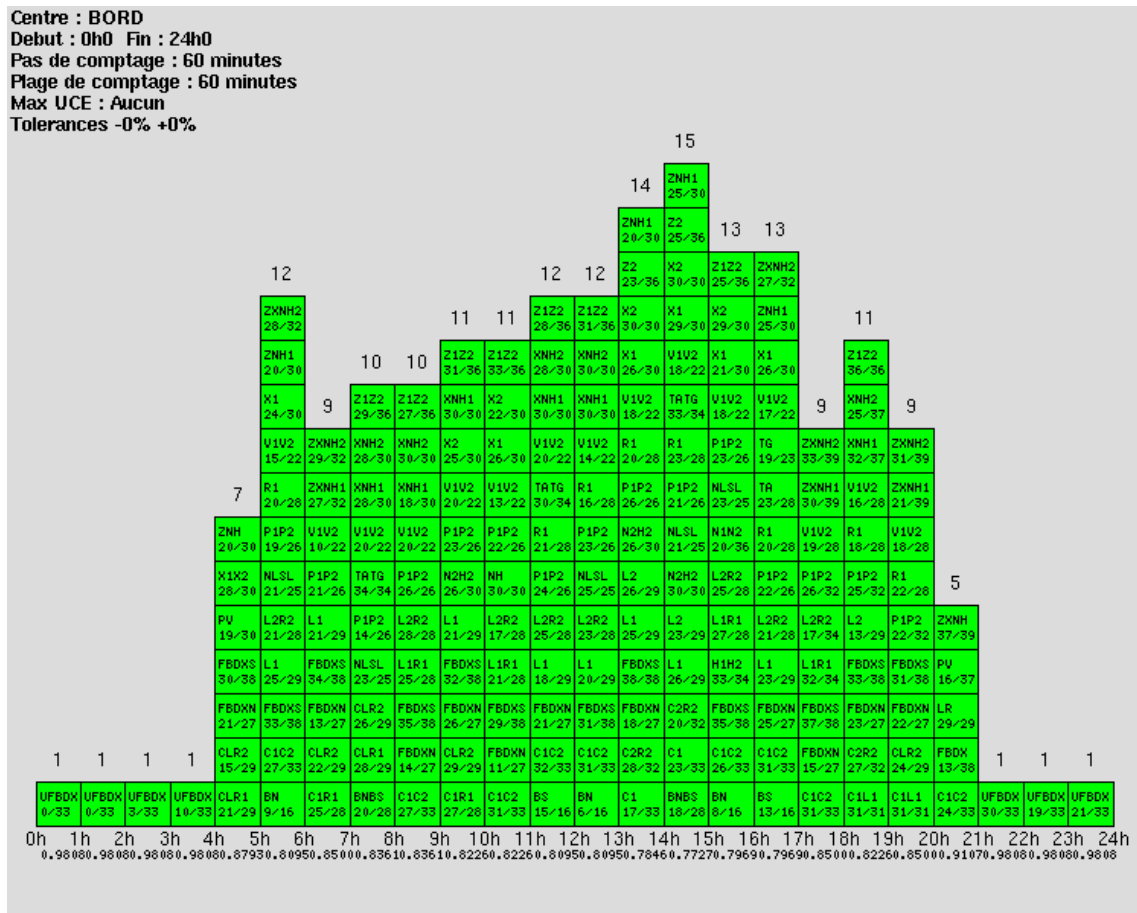


FIG. 4.3 – Configurations optimales pour Bordeaux, avec lissage préalable du trafic (OPT.0.0/

tolérances nulles sur les valeurs des capacités. Les délais produits par chaque méthode sont présentés dans le tableau 4.5, et les ressources utilisées dans chaque cas sont détaillées pour chaque centre dans le tableau 4.6. Ces dernières sont calculées en cumulant les temps d’ouvertures, en minutes, des positions de contrôle.

La figure 4.3 présente le résultat de la stratégie OPT.0.0 pour Bordeaux, avec des tolérances nulles sur les capacités : nous constatons qu’il ne reste plus aucun dépassement de capacité. Sur la figure 4.4, il est possible de comparer, pour chaque moment, de la journée le schéma optimisé et le schéma déposé.

On constate que la stratégie optimisée apporte une diminution plus qu’appréciable du total des retards au décollage (69%), tout en utilisant près de vingt pour cent de ressources en moins... Face à d’aussi bons résultats, on ne peut s’empêcher de se dire que c’est trop beau pour être vrai. Et effectivement, il ne faut pas s’imaginer que les méthodes d’optimisation décrites dans ce chapitre permettraient de tels gains si elles étaient utilisées dans le processus de régulation des flux de trafic. Les résultats doivent uniquement être considérés comme une indication de l’efficacité des algorithmes et de la méthode d’optimisation proposés, *sur la base des indicateurs de charge actuellement utilisés dans le processus opérationnel*. Nous avons implicitement supposé depuis le départ que le flux d’avions entrant dans un secteur dans une période de temps donnée était représentative de la charge de travail du contrôleur. Les valeurs seuils que sont les capacités horaires associées à chaque secteur sont censées

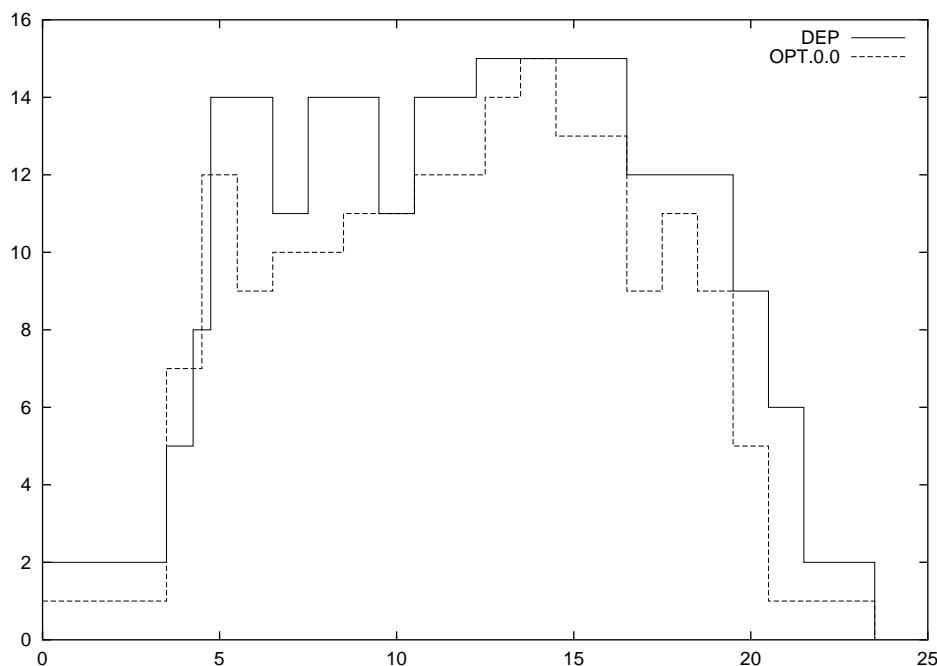


FIG. 4.4 – Comparaison entre OPT.0.0 et DEP pour Bordeaux

représenter une limite au-delà de laquelle la charge de travail est telle qu'il faut dégroupier lorsque c'est possible, ou bien réguler le trafic, pour ne pas dépasser la capacité.

Or, comme le savent bon nombre de professionnels du milieu ATC ou ATFM, même s'ils l'oublient parfois lors des tractations sur les valeurs des capacités, le nombre d'avions qui va entrer dans l'heure suivante dans un volume d'espace donné n'est pas un bon indicateur de la charge de travail du contrôleur gérant ce volume. Et, comme va nous le confirmer une étude statistique sommaire des dégroupements de secteurs sur l'année 1999, les flux entrants d'avions ne sont pas corrélés avec les décisions de dégroupement d'un secteur.

4.8 Etude statistique sommaire des dégroupements de secteurs

4.8.1 Démarche

L'objectif de cette étude est d'examiner le comportement de quelques indicateurs de base autour des instants où un secteur composé de plusieurs secteurs élémentaires est *dégroupé* en deux secteurs ou plus, de plus petite taille. Les indicateurs choisis sont le nombre d'avions présents dans le secteur, et les flux entrants avec différents horizons temporels.

Les données dont on dispose sont, d'une part, les archives des plans de vol pour l'année 1999 et, d'autre part, les archives des ouvertures de secteurs réalisées par le chef de salle sur sa console de supervision. Rappelons que l'espace aérien français est composé de secteurs élémentaires. Chaque position de contrôle a sous sa responsabilité un secteur ou un groupe de secteurs (regroupement). L'affectation des secteurs aux positions est gérée en temps réel par le chef de salle, en fonction du trafic et de contraintes diverses, dont notamment le nombre de positions qu'il est possible d'armer. L'affectation d'un secteur sur une position implique essentiellement, pour ce qui est de la gestion des systèmes, le basculement des fréquences radio sur la position, le basculement des lignes téléphoniques

spécialisées, et une affectation logique du secteur sur la position dans le Système de Traitement des Plans de Vol (STPV).

Dans ce qui suit, le terme secteur sera entendu au sens large (secteur élémentaire, ou regroupement de secteurs élémentaires). Les données utilisées sont les archives COURAGE de l'année 1999, et plus précisément :

- les schémas réalisés d'ouvertures secteurs, issus des affectations logique secteur/position du STPV,
- les plans de vol réalisés,
- les données de description de l'espace aérien ("données CA").

Les plans de vols archivés contiennent en particulier la liste des secteurs traversés et les heures d'entrée et de sortie des secteurs. Une autre donnée intéressante, également archivée, est l'heure de fourniture des *strips* aux positions de contrôle concernées par un vol donné. Un *strip* est une bande de papier sur laquelle figure un certain nombre d'informations nécessaires au suivi du vol. On peut donc calculer le flux de strips arrivant sur une position, ainsi que leur nombre sur le tableau de strips à un instant donné, sachant que ces informations sont susceptibles de donner des indications sur la charge de travail du contrôleur.

La démarche a consisté, pour les secteurs dégroupables, à rechercher parmi différentes variables si certaines d'entre elles étaient plus significatives de l'acte de dégroupement. A cet effet, on examine au moment du dégroupement ou dans son voisinage, les diverses variables suivantes :

- le nombre d'avions présents dans le secteur à l'instant t ,
- le nombre de strips présents sur la position de contrôle, à l'instant t ,
- les flux entrants (en nombre d'avions ou en nombre de strips) avec un horizon temporel allant de 10 à 60 minutes.

La pertinence de chacune des variables peut être appréciée en examinant les différences entre les valeurs prises avant dégroupement, et celles prises après dégroupement. S'il n'y a pas de différences dans la distribution des valeurs avant ou après dégroupement, la variable considérée ne pèse pas beaucoup dans la décision de dégroupement, alors que si l'on peut clairement discriminer les deux classes de valeurs, la variable est pertinente.

On restreint l'étude à une plage horaire située entre 2h du matin et 22h, afin d'éviter les biais aux alentours de minuit dus à l'incomplétude des données plan de vol. Les résultats sont présentés pour un seul secteur (ZX, composé des secteurs élémentaires ZS,ZU,XS, et XU), sachant que les résultats pour les autres secteurs dégroupables sont analogues.

4.8.2 Valeurs des variables au moment du dégroupement

Ici, différentes variables sont mesurées aux instants où le secteur est dégroupé en deux autres secteurs ou plus. L'idée étant que si la décision de dégroupement est effectivement liée au passage d'un seuil unique pour une variable significative du dégroupement, alors la distribution des valeurs au moment du dégroupement sera étroitement centrée sur cette valeur seuil. Ceci ne peut cependant être vrai que si la valeur seuil en question est la même pour chaque tandem de contrôleurs et à tout moment de l'année, et si les décisions de dégroupement ne sont prises qu'en fonction de la seule variable considérée. On se doute bien que ce n'est pas le cas en pratique.

Les courbes de distribution suivantes représentent, pour un secteur donné et pour chacune des variables : en abscisse les valeurs des charges secteurs et en ordonnée le nombre de dégroupements effectués à cette mesure de charge.

Plusieurs courbes sont tracées, en prenant comme mesure de charge le nombre de strips ou bien le nombre d'avions, en considérant une charge instantanée, ou bien un flux entrant avec un horizon

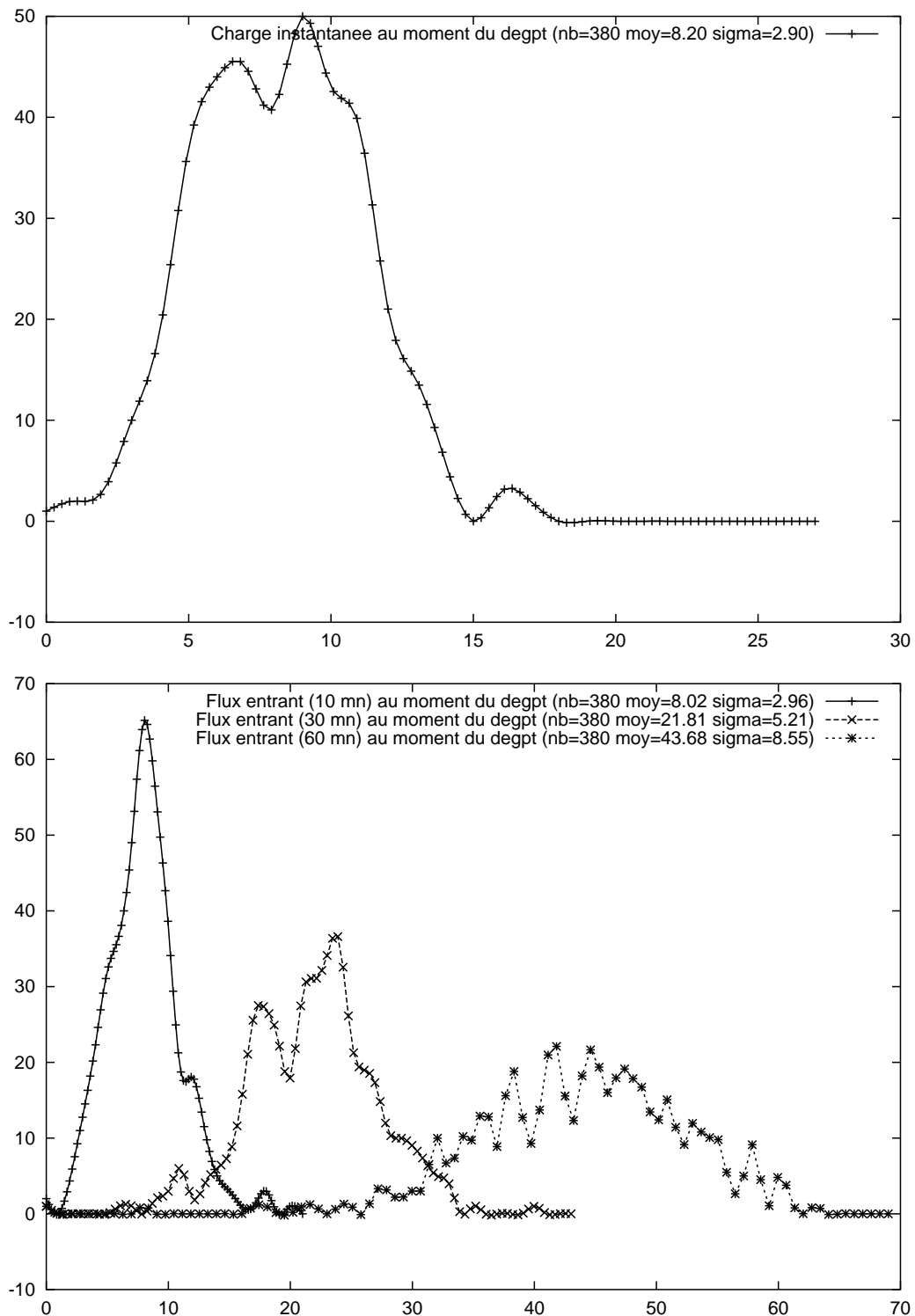


FIG. 4.5 – ZX : charges instantanées et flux entrants, en nombre d'avions, aux instants de dégroupement (abscisse : valeur de l'indicateur ; ordonnée : nombre d'occurrences).

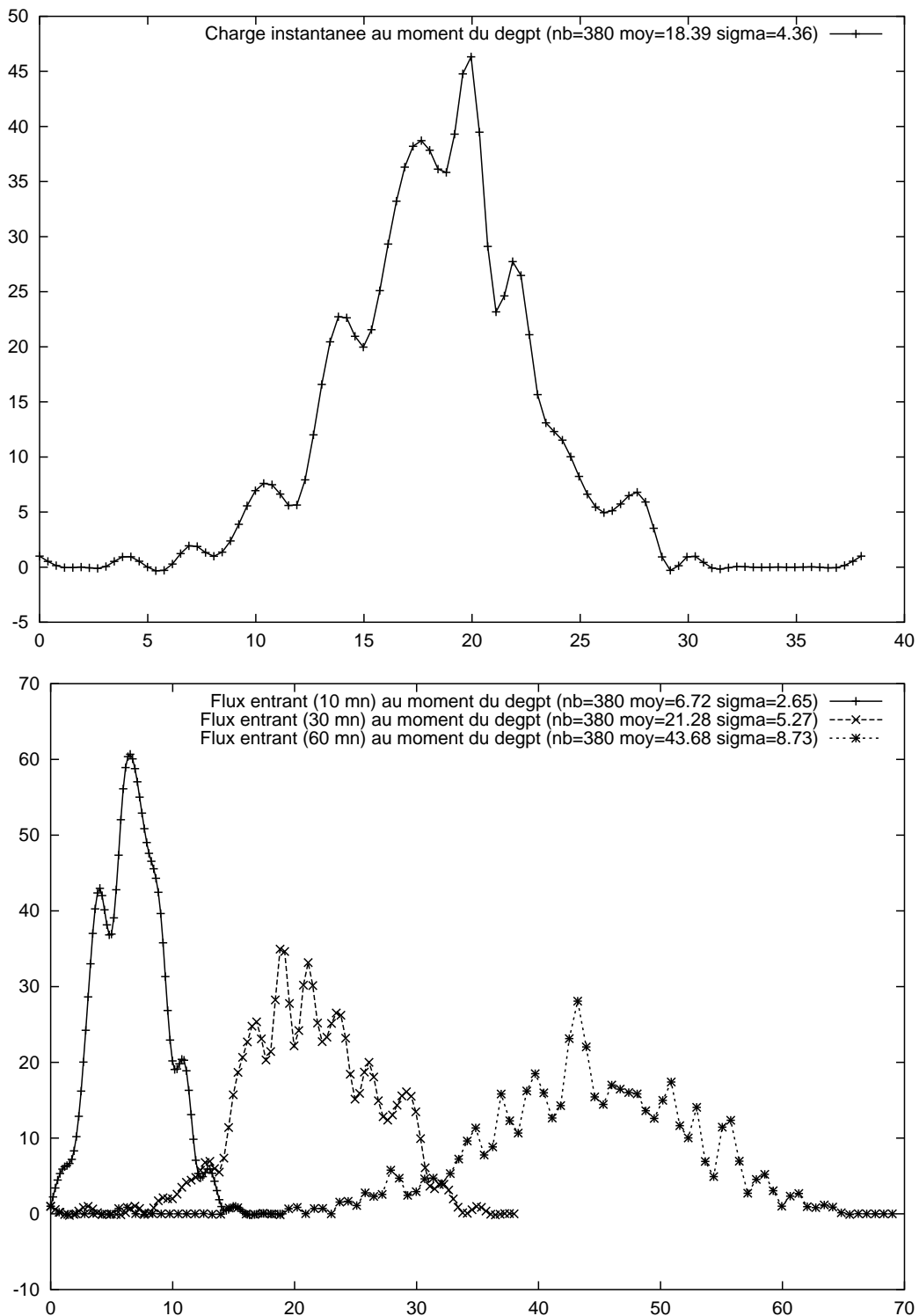


FIG. 4.6 – ZX : charges instantanées et flux entrants, en nombre de strips, aux instants de dégroupement (abscisse : valeur de l'indicateur ; ordonnée : nombre d'occurrences).

temporel allant de 10 à 60 minutes.

Sur les figures 4.5 et 4.6, on voit que si l'on fait l'hypothèse (non vérifiée) que les courbes peuvent être assimilées à des gaussiennes, alors 95% des dégroupements de ZX se font :

- avec une charge instantanée en nombre d'avions entre 2,5 et 13,9 ;
- avec un flux entrant horaire en nombre d'avions entre 26,9 et 60,4 ;
- avec une charge instantanée en nombre de strips entre 9,8 et 26,9 ;
- avec un flux entrant horaire en nombre de strips entre 26,6 et 60,8.

Notons à propos des charges en nombre de strips que chaque strip est compté entre le moment de la desserte du strip et la sortie de l'avion des limites géographiques du secteur. Ce n'est pas forcément réaliste, mais on ne connaît pas l'heure exacte à laquelle chaque strip est effectivement retiré du tableau. Le strip est distribué une dizaine de minutes environ avant l'entrée géographique de l'avion dans le secteur, ce qui explique le décalage vers les charges supérieures de la courbe de charge instantanée en nombre de strips par rapport à la courbe de charge instantanée en nombre d'avions.

On constate, sur toutes les courbes, que la fourchette des valeurs extrêmes est assez large. Ceci traduit sans doute la diversité des capacités de chaque tandem de contrôleurs, mais aussi la diversité des causes possibles de dégroupement du secteur : ce ne sont pas les seuls impératifs de charge de travail qui sont à l'origine d'un dégroupement. Parmi les autres causes possibles, on peut citer, à titre d'exemple, la formation de jeunes contrôleurs, des tests et expérimentations diverses de matériels ou de nouvelles procédures, la nécessité d'enregistrer des heures de contrôle en vue du renouvellement d'une qualification, un problème technique, etc...

Cependant, l'existence d'une borne supérieure peut être une donnée intéressante, à condition qu'elle ne soit pas une simple conséquence de la répartition du trafic (il n'y a pas de dégroupements au delà de la valeur X, car il n'y a pas de valeurs supérieures à X pour lesquelles il y a un trafic significatif dans le secteur), et si l'on peut montrer qu'au delà de cette valeur seuil, le secteur est toujours dégroupé.

Toutefois, il n'y a de réel intérêt dans ces informations que si la variable considérée est significative du dégroupement. Or, les courbes présentées jusqu'à présent ne permettent pas de déterminer si une des variables est plus pertinente que les autres quant à la décision de dégroupement.

4.8.3 Comparaison des valeurs avant et après dégroupement

Afin d'y voir un peu plus clair, nous allons mesurer ces variables toutes les minutes dans une période allant, lorsque c'est possible, de dix minutes avant chaque dégroupement jusqu'à dix minutes après. Les valeurs mesurées sont classées en deux groupes (avant et après dégroupement) et visualisées de la même façon que précédemment, avec en abscisse la valeur de la variable, et en ordonnée le nombre de points mesurés à la valeur de la variable.

On constate sur les deux courbes présentées, que, pour les flux entrants, il est difficile de distinguer les deux classes, les courbes étant quasi confondues, alors que, pour les charges instantanées, les courbes avant et après dégroupement sont un peu plus nettement séparées. Cette constatation tend à confirmer que la décision de dégroupement est plus liée, pour ce secteur, à la charge instantanée qu'au flux d'avions entrants, quel que soit l'horizon temporel choisi pour ce dernier.

Il faut toutefois garder à l'esprit certains facteurs pouvant fausser partiellement l'interprétation des résultats, Par exemple, les éventuelles modifications en cours d'année des limites géographiques des secteurs peuvent plus ou moins biaiser les résultats. Par ailleurs, la diversité des causes de dégroupement d'un secteur ne peut pas apparaître dans le type de résultats présentés. Il est évident que, dans la pratique, l'ouverture d'un secteur peut être liée à autre chose que l'atteinte d'un certain seuil dans la charge de travail (comme la formation d'un contrôleur débutant, etc...).

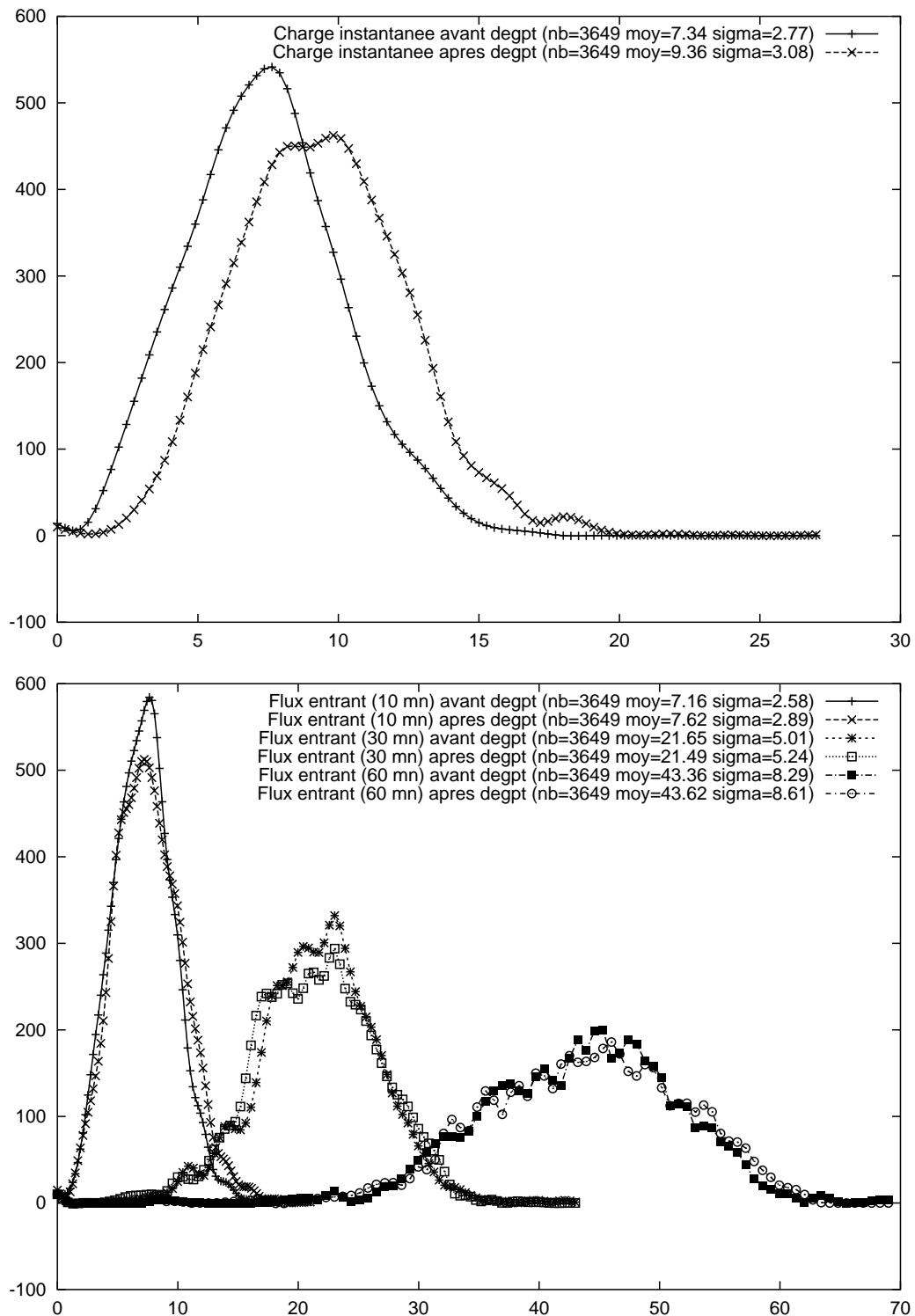


FIG. 4.7 – ZX : charges instantanées et flux entrants, en nombre d'avions, avant et après dégroupement (abscisse : valeur de l'indicateur ; ordonnée : nombre d'occurrences).

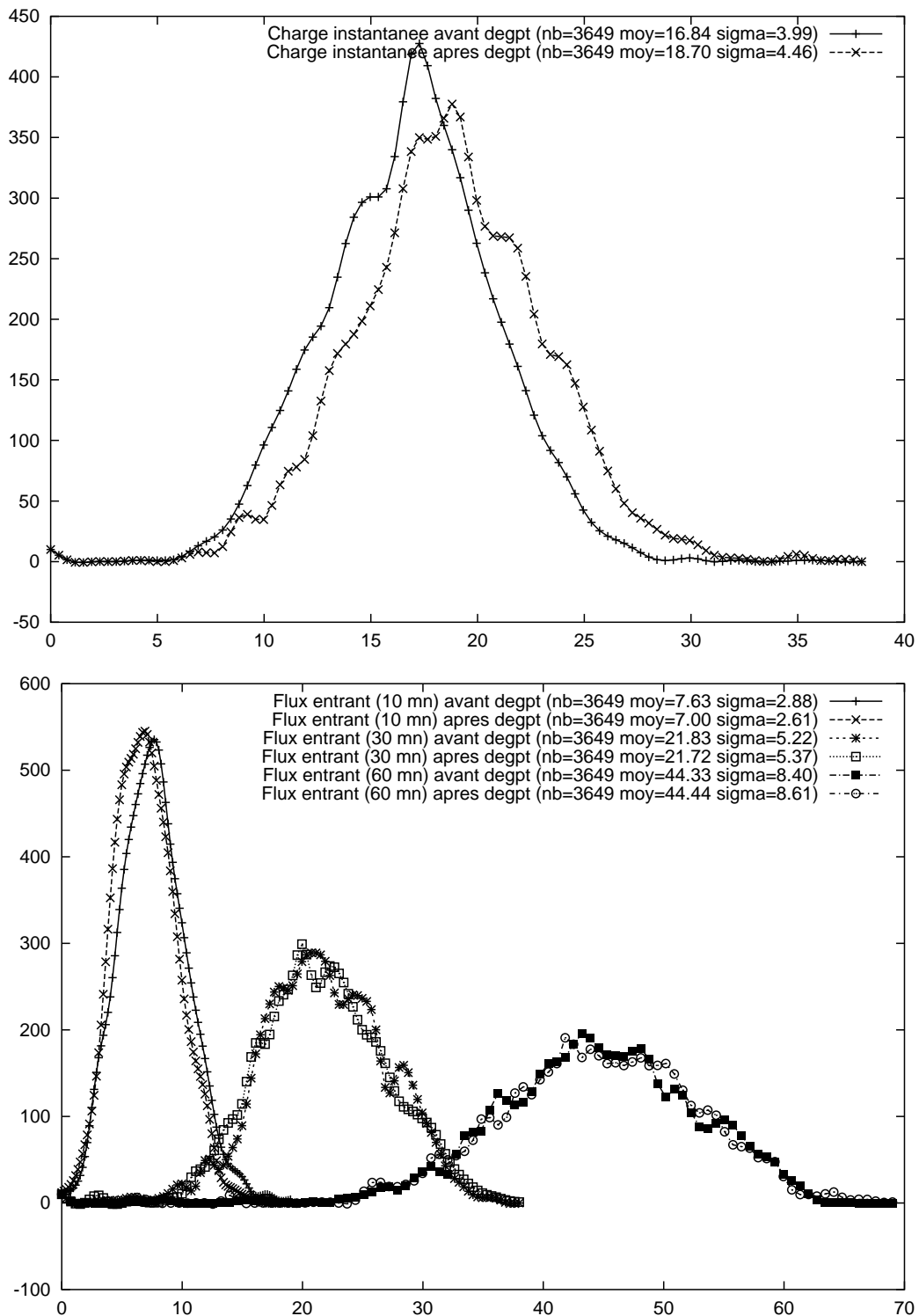


FIG. 4.8 – ZX : charges instantanées et flux entrants, en nombre de strips, avant et après dégroupement (abscisse : valeur de l'indicateur ; ordonnée : nombre d'occurrences).

Malgré tout, il semble que la charge instantanée soit généralement une variable un peu plus pertinente dans la décision de dégroupement que les flux entrants. Ceci tendrait à confirmer que les valeurs déclarées des capacités horaires, utilisées dans l'élaboration des schémas prévisionnels d'ouvertures des secteurs, sont essentiellement une entrée du processus de régulation FMP/CFMU, et sont ajustées en fonction du retour d'expérience. Elles ne doivent pas être considérées comme des valeurs seuils dans la décision de dégroupement ou de régulation.

On voit donc, pour revenir au sujet de départ (l'optimisation des configurations d'ouverture des secteurs) que chercher à optimiser des schémas prévisionnels construits uniquement sur la base de contraintes en capacités horaires n'est sans doute pas opérationnellement très intéressant.

Cette étude n'est qu'une première approche et les quelques résultats présentés demanderaient à être confirmés et affinés par une véritable analyse discriminante sur un ensemble de variables pouvant entrer en jeu dans la décision de dégroupement, à définir plus exhaustivement.

4.9 Conclusion sur l'optimisation des regroupements de secteurs

Les algorithmes classiques et l'algorithme génétique trouvent des configurations optimales de secteurs, en prenant en compte les mêmes paramètres et contraintes que dans le processus FMP/CFMU. Les résultats sont obtenus en un temps compatible avec une utilisation opérationnelle. Les algorithmes de recherche dans des arbres sont plus rapides que l'algorithme génétique, pour les centres français et dans le contexte des groupes de secteurs issus des données des centres.

Cependant, la relation combinatoire entre le nombre de secteurs et le nombre de partitions et de configurations est telle que les algorithmes classiques risquent d'être impraticables dans le contexte de plus grands centres de contrôle ou si l'on souhaite élargir l'ensemble des groupes de secteurs opérationnellement utilisables. Dans un tel cas, l'algorithme génétique est une bonne alternative. Il fournit plusieurs configurations optimales ou presque optimales en un temps de calcul borné par l'utilisateur.

Les résultats sur les stratégies d'allocation de créneaux sont bons : environ 70% de gain en délais en utilisant environ 20% moins de ressources. Cependant, la nouvelle stratégie ne tient pas compte des incertitudes sur la prévision de trafic. De plus, comme nous l'avons vu dans l'étude statistique autour des instants de dégroupement, les flux de trafic et les capacités horaires sont de mauvais indicateurs de la charge de travail du contrôleur aérien et de son seuil de saturation. En conséquence, il n'est pas certain qu'une méthode optimisant les délais alloués et les schémas d'ouvertures sur la base de ces indicateurs conduise vraiment à un réel lissage de la charge de travail des contrôleurs. Notons que ces remarques sont également valables pour la méthode actuelle, mais celle-ci a au moins l'avantage d'être validée par un retour d'expérience.

Jusqu'à présent, nous avons considéré l'optimisation des configurations de secteurs uniquement dans le contexte des schémas d'ouvertures prévisionnels réalisés par les FMP, sur la base des flux de trafic prévus et des capacités horaires. Les algorithmes proposés pourraient également être envisagés, avec une définition adaptée de la charge de travail et des seuils et contraintes, pour proposer en temps réel des configurations de secteurs aux chefs de salles de contrôle.

Dans ce chapitre, nous avons cherché à améliorer le système existant sans le modifier, en jouant simplement sur les regroupements de secteurs de contrôle. Nous allons dorénavant nous autoriser à modifier ce système, en cherchant tout d'abord, dans le chapitre suivant, à construire un réseau de routes selon des critères simples d'espacement des points de croisement.

Chapitre 5

Construction d'un réseau de routes

5.1 Introduction

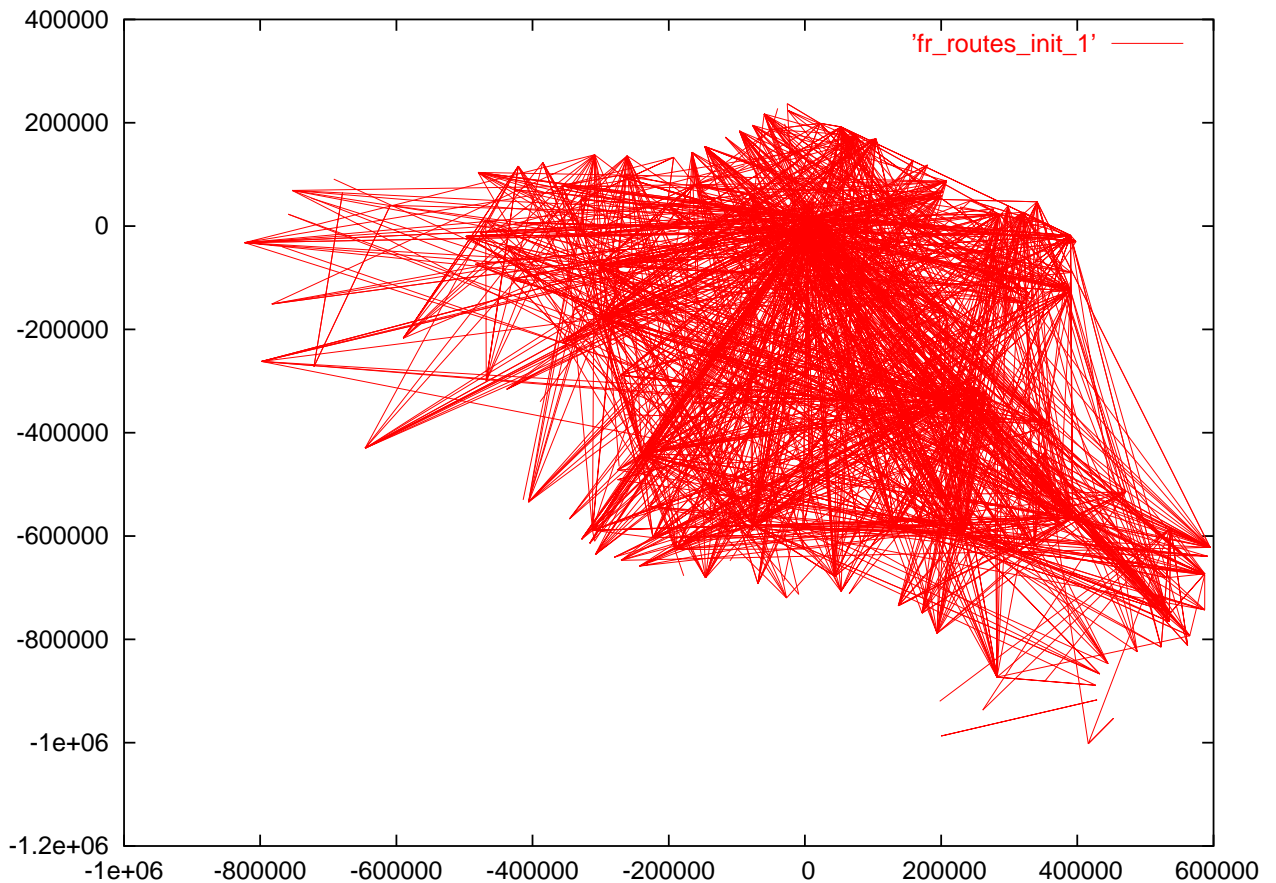


FIG. 5.1 – Routes directes en France (trafic du 21/05/99 ; projection stéréographique centrée sur Paris ; l'unité est le mètre).

Le problème de la construction d'un réseau de routes, en partant initialement des routes directes, a déjà été traité dans [Meh00], selon une approche en deux étapes, la première consistant à fusionner

les points de croisement trop proches les uns des autres, et la seconde à déplacer les points obtenus, afin d'optimiser un critère de densité ATC. Cet indicateur de densité est calculé à partir du nombre et de la nature des croisements, et de la surface des polygones de Voronoï autour de chaque point de croisement. La première étape tient compte de contraintes d'angle maximal entre tronçons de routes, sans toutefois garantir leur respect absolu dans tous les cas. La deuxième étape, ayant pour objectif de diminuer localement le critère de densité autour des points congestionnés, n'est pas détaillée dans [Meh00].

Dans ce chapitre, nous allons chercher à construire un réseau de routes "intelligible" pour l'être humain – nous reviendrons un peu plus loin sur cette notion – en utilisant d'abord la méthode d'agrégation présentée en 3.3 sur les points de croisement des routes directes, puis une triangulation de Delaunay pour reconstituer le réseau à partir des nouveaux points obtenus, et enfin un algorithme A^* pour trouver les plus courts chemins dans ce réseau, en respectant une contrainte d'angle maximum entre segments successifs de route.

Revenons maintenant sur la notion d'un réseau "intelligible" par l'être humain. Du point de vue d'un opérateur aérien, la situation idéale serait de faire voler ses avions sur des routes directes entre le point de départ et le point d'arrivée. En supposant que tous les opérateurs suivent ce raisonnement, on obtiendrait un ensemble de routes comme celui représenté sur la figure 5.1, en considérant les plans de vols d'une journée de trafic au-dessus de la France (le 21/05/99 sur l'exemple). On conçoit facilement qu'un contrôleur aérien ou un opérateur de régulation du trafic puisse avoir quelque difficulté à se représenter la circulation du trafic sur un tel réseau, notamment à cause du nombre important de points de croisement des routes (249455 sur l'exemple, pour 5337 vols).

On peut déjà simplifier ce réseau de routes en ne considérant pas la totalité du trafic, mais seulement les flux les plus importants. La figure 5.2 représente les flux de plus de dix avions. Il ne reste plus que 2159 points de croisements pour 504 flux.

Malgré tout, il reste des zones où la densité des points de croisements reste importante. Si l'on reste dans le cadre des méthodes de contrôle actuelles, où chaque unité de contrôle d'espace a en charge une portion de l'espace aérien, on voit qu'il est impossible de faire un découpage équilibré de cet espace en secteurs exploitables, sans imposer au préalable certaines contraintes sur la densité des points de croisement.

5.2 Agrégation des points de croisement

L'idée développée ici consiste à imposer un espacement minimal entre les points de croisement des routes dans le plan horizontal, afin de rendre le réseau aérien suffisamment intelligible aux êtres humains. Pour cela, on cherche à classifier notre ensemble de points de croisement selon un critère de distance, par un processus de partitionnement, puis à remplacer chacun des points de croisement par le barycentre de l'agrégat auquel il appartient.

Le principe de l'agrégation par nuées dynamiques a été exposé au chapitre 3, dans la section 3.3. La méthode appliquée ici consiste à considérer successivement tous les points que l'on souhaite classifier en procédant comme suit. Un point sera rattaché à l'agrégat existant le plus proche s'il est à une distance inférieure à un seuil $d1$ du barycentre de celui-ci, sinon un nouvel agrégat est créé, contenant comme seul élément le point considéré. Comme les barycentres des agrégats sont susceptibles de se déplacer au fur et à mesure que l'on y rajoute des points, il peut s'avérer nécessaire de fusionner des agrégats. On définit pour cela un deuxième seuil de décision $d2$: si deux agrégats sont à une distance inférieure à $d2$, ils sont fusionnés.

Ce processus permet de passer de la situation initiale (figure 5.2) à un réseau final plus intelligible

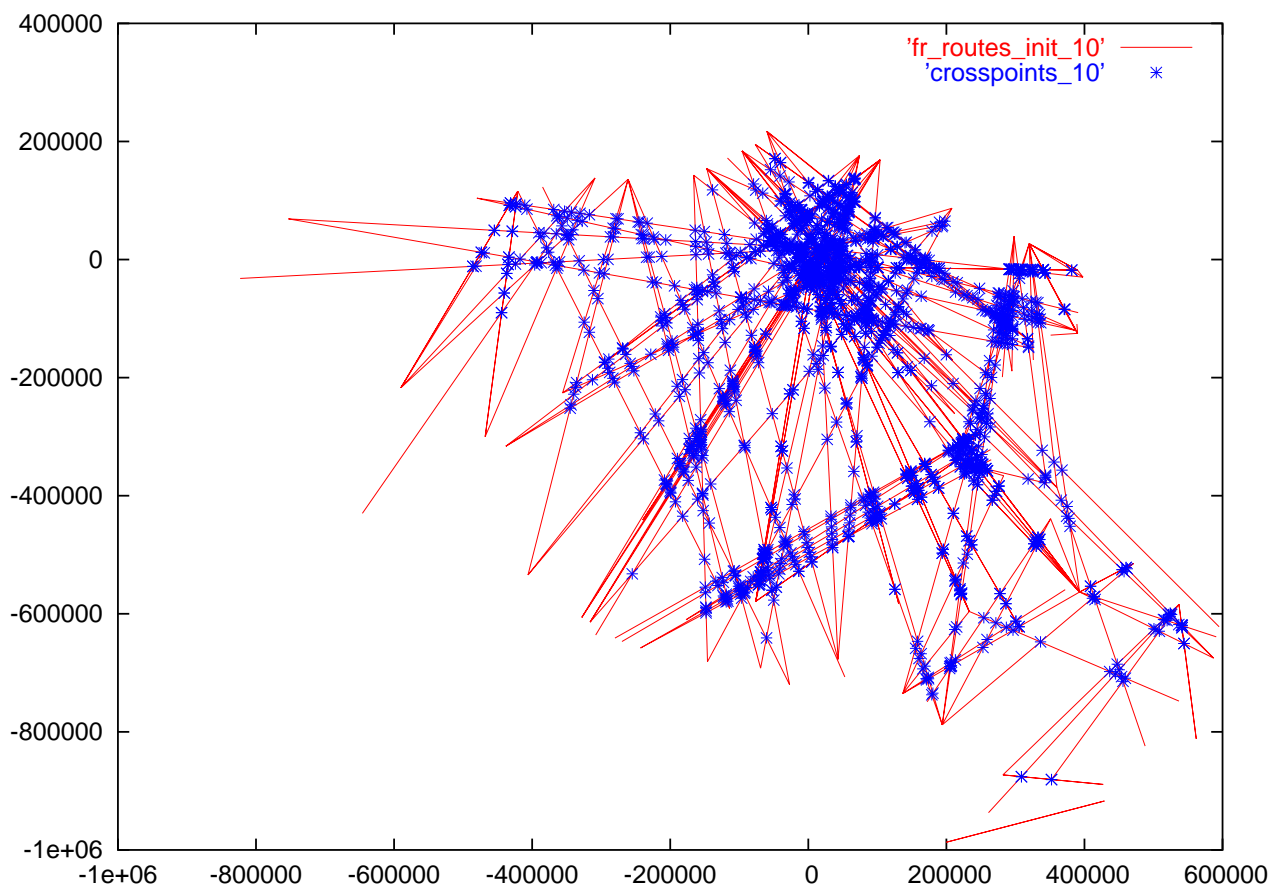


FIG. 5.2 – Routes directes et points de croisement pour les flux de plus de 10 avions (trafic du 21/05/99 ; projection stéréographique centrée sur Paris ; l'unité est le mètre).

pour l'être humain, tel que celui représenté sur la figure 5.4. La figure 5.3 nous montre le nuage des points de croisement initiaux et les barycentres des agrégats, qui deviennent les points de croisement du nouveau réseau de routes.

D'autres procédés de classification existent. Celui-ci a été retenu car son ordre de complexité en $N \log(N)$ (cf [YG93]) est inférieur à celui d'autres types algorithmes. Il présente par contre l'inconvénient d'être dépendant de l'ordre dans lequel les points sont présentés, ce qui ne nous dérange pas outre mesure, puisque l'on choisit systématiquement d'ordonner les points de croisement selon le trafic y transitant.

Toutefois, l'agrégation ne considère que la position géographique des points de croisement, en délaissant des contraintes importantes, comme la limitation des valeurs des angles entre segments de routes consécutifs. En remplaçant, sur chaque route directe, les points de croisement successifs par leurs centres de classe, on obtient une nouvelle route, par déformation de la route initiale. La figure 5.4 montre le tracé d'une telle route, comportant des angles aigus incompatibles avec les performances de navigation des avions. Le réseau obtenu n'est donc pas exploitable opérationnellement.

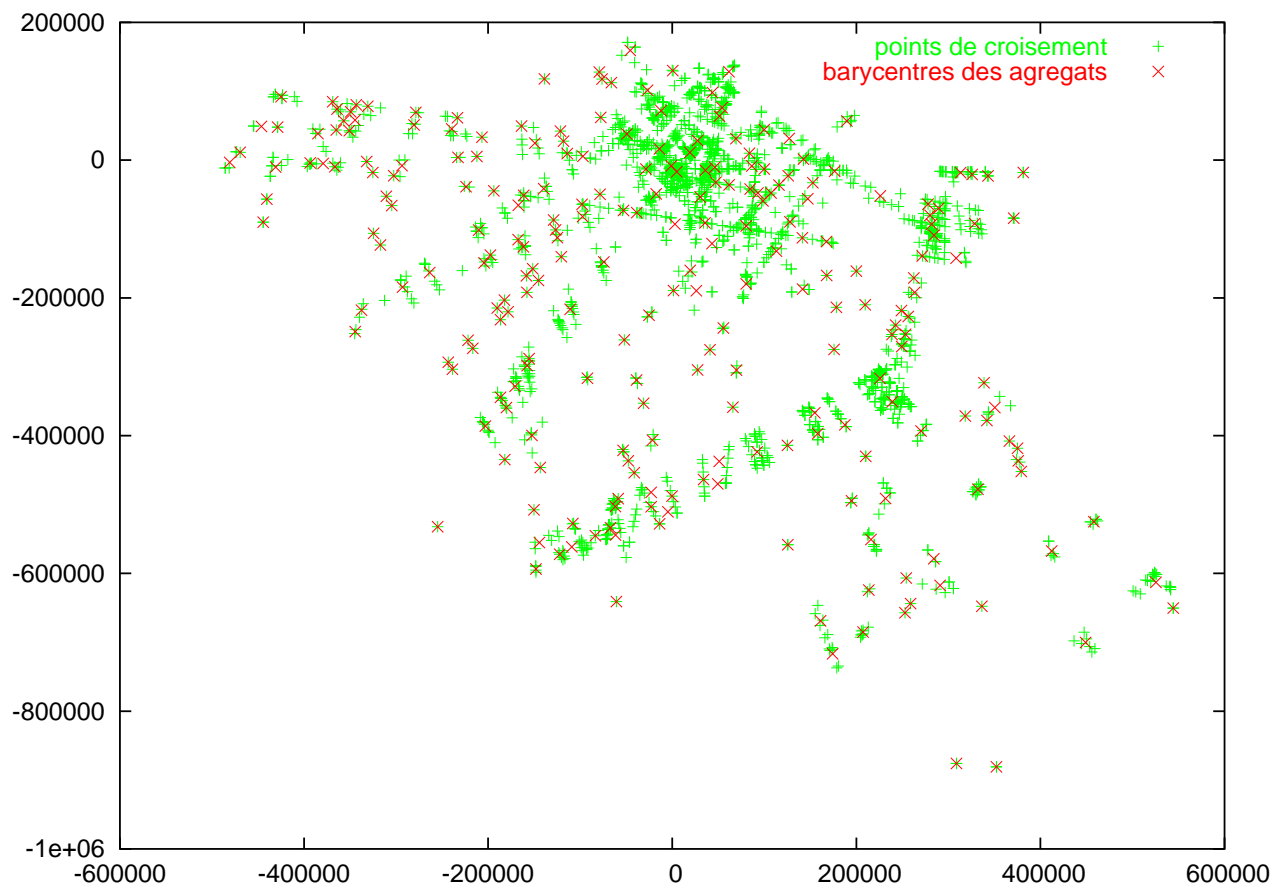


FIG. 5.3 – Agrégation des points de croisement (trafic du 21/05/99 ; projection stéréographique centrée sur Paris ; l'unité est le mètre).

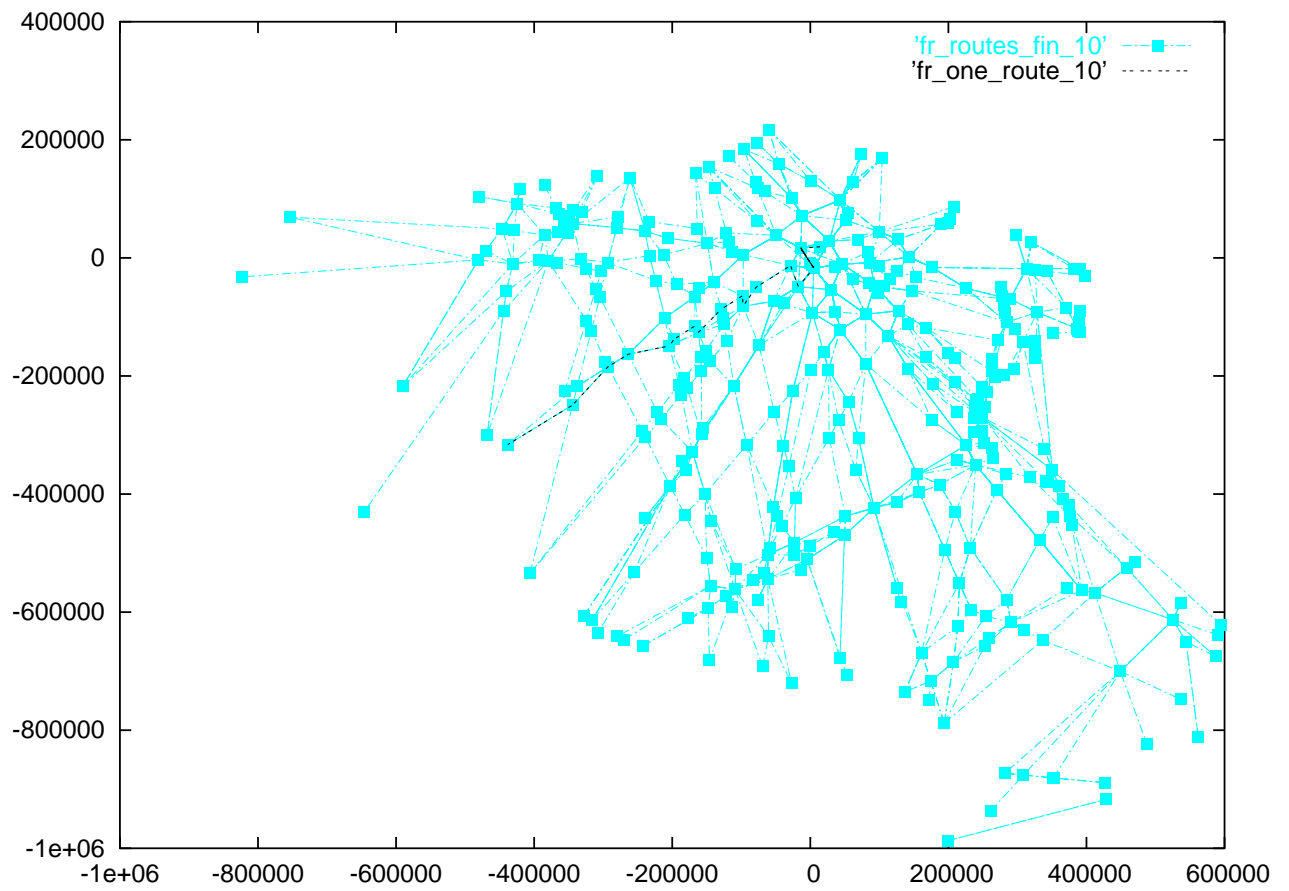


FIG. 5.4 – Routes obtenues par déformation des routes directes (trafic du 21/05/99 ; projection stéréographique centrée sur Paris ; l'unité est le mètre).

5.3 Construction d'un réseau par triangulation de Delaunay

Dans la section précédente, nous avons constaté l'inexploitabilité du réseau de routes obtenu par simple déformation des lignes droites origine-destination lors du processus d'agrégation des points de croisement de ces lignes. Une piste possible pour contourner ce problème pourrait être de revoir ce processus, en agrégeant non pas des points, mais des segments de routes, avec une topologie à définir.

Une solution plus simple consiste à reconstituer un nouveau réseau à partir du maillage de points issus du processus d'agrégation des points de croisement. Il restera alors à affecter une route à chaque flux. La route associée à un flux sera alors obtenue en recherchant le plus court chemin dans le nouveau réseau entre l'origine et la destination du flux, en tenant compte de contraintes d'angle maximum entre tronçons de route successifs.

Pour créer le nouveau réseau, on ne peut pas simplement relier deux à deux tous les points d'agrégation. En effet, chaque point d'agrégation ne doit être relié qu'à ses «voisins». De plus, les segments de route ne doivent pas se croiser ailleurs qu'en leurs extrémités.

La notion de voisinage de points est formalisée par le diagramme de Voronoï (cf chapitre 3, section 3.4) qui permet de définir la zone d'espace constituant le voisinage d'un site donné : tous les points de cette zone sont plus proches du site que de n'importe quel autre site.

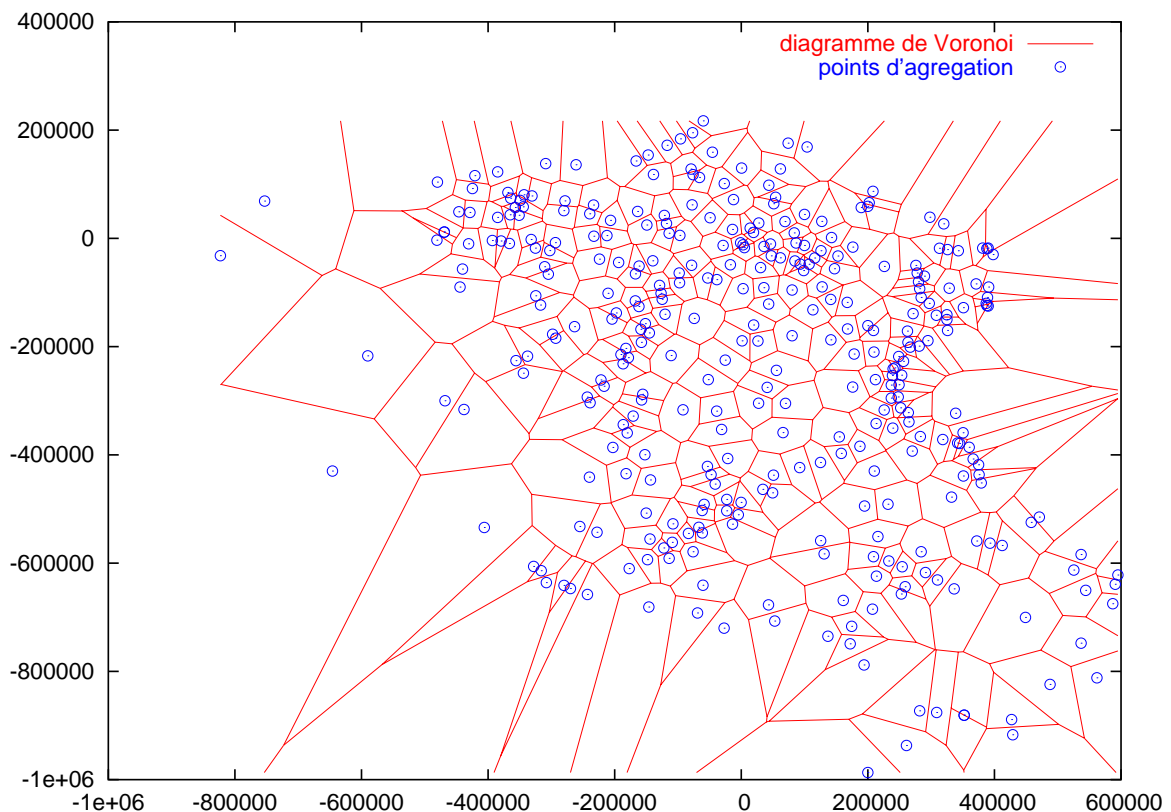


FIG. 5.5 – Diagramme de Voronoï des points d'agrégation (trafic du 21/05/99 ; projection stéréographique centrée sur Paris ; l'unité est le mètre).

La figure 5.5 montre le diagramme de Voronoï associé aux centres de classe issus du processus d'agrégation des points de croisement entre routes directes, pour des flux de plus de 10 avions par jour.

Une triangulation de Delaunay, qui est l'aspect dual du diagramme de Voronoï, permet d'obtenir le réseau de routes : on ne relie que les sites dont les zones de voisinage ont une frontière commune. L'algorithme de Fortune, détaillé au chapitre 3 (section 3.4), permet d'obtenir simultanément les deux diagrammes. La triangulation de Delaunay, pour les flux de 10 avions ou plus, est représentée en figure 5.6.

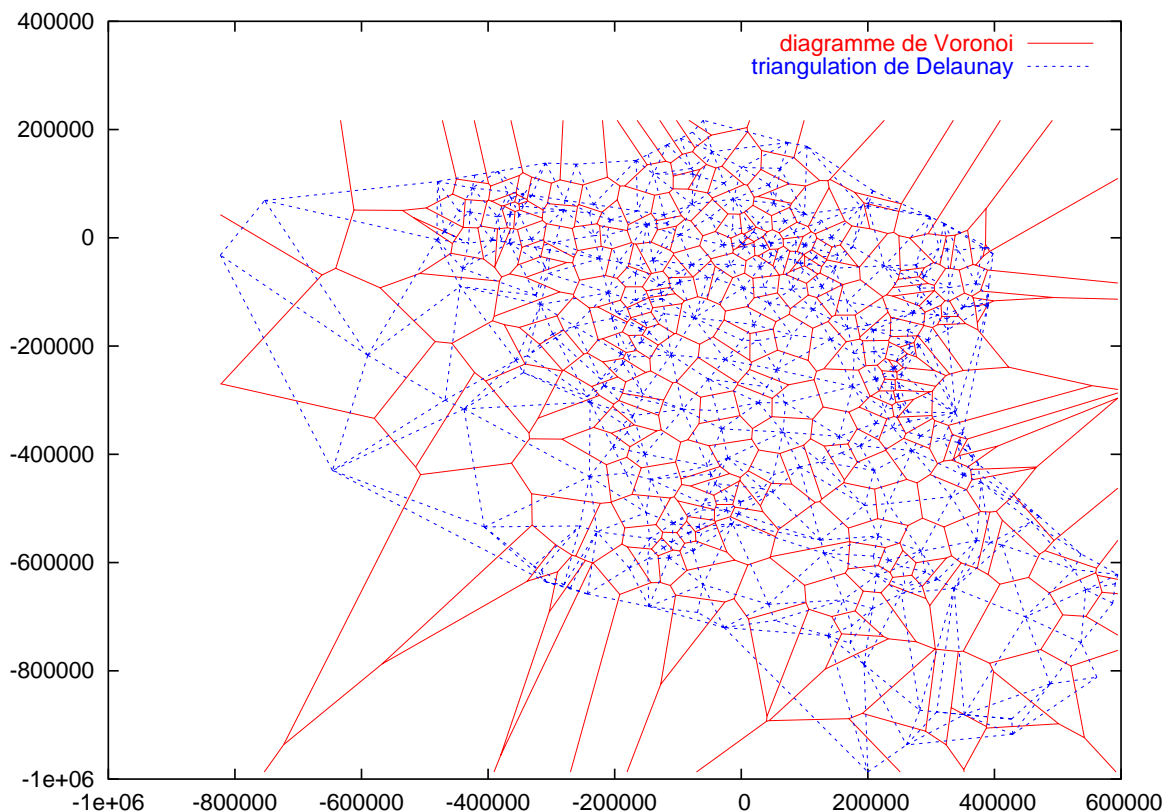


FIG. 5.6 – Triangulation de Delaunay des points d'agrégation (trafic du 21/05/99 ; projection stéréographique centrée sur Paris ; l'unité est le mètre).

On a donc obtenu un réseau de routes respectant les contraintes d'espacement des points de croisement. A chaque point de croisement de ce nouveau réseau est associée une zone du diagramme de Voronoï. On voit aisément l'intérêt de ces zones pour la construction d'une sectorisation de l'espace aérien.

5.4 Considérations sur la sectorisation associée au réseau de routes

L'observation de la figure 5.6 permet de constater que, si le tracé de chaque segment de route se fait bien entre sites dont les zones ont une frontière commune, ce tracé ne passe pas toujours à travers cette frontière. Autrement dit, un segment de route peut traverser d'autres zones que celles associées aux deux points qu'il relie. Cette non-convexité relativement au réseau de routes n'est pas compatible avec une sectorisation opérationnelle : on souhaite que les routes entre deux secteurs traversent la frontière entre ces secteurs.

Par ailleurs, un certain nombre de zones sont trop petites pour pouvoir constituer un secteur de

contrôle viable : la charge de travail de coordination entre un tel secteur et ses voisins serait beaucoup trop importante.

Enfin, le découpage stratégique de l'espace aérien en secteurs doit aussi être équilibré : les secteurs élémentaires doivent représenter en moyenne une charge de travail qui soit ni trop faible ni trop importante pour un contrôleur aérien. Le fait d'avoir espacé les points de croisement devrait répondre en partie à cette exigence, à condition de s'assurer que l'affectation des flux de trafic sur les routes soit également équilibrée.

Notons que la dimension verticale était totalement absente de ce chapitre. Nous avons considéré uniquement la projection horizontale du réseau de routes. Ceci se justifie par le fait qu'actuellement, pour des raisons de simplicité, l'espace est découpé en zones dont la projection horizontale est un polygone, auquel on associe des altitudes plancher et plafond. Un secteur est constitué d'une seule zone de ce type ou bien d'un empilement de telles zones. On définit donc d'une part des tranches d'altitudes correspondant à différentes natures de trafic, et d'autre part un découpage en zones pour chacune de ces tranches.

Pour obtenir une sectorisation opérationnellement exploitable associée au réseau de routes que nous avons défini, il faudrait donc, d'une part, répartir les flux de trafic sur les routes du réseau de façon intelligente, et, d'autre part, constituer des secteurs de contrôle à partir des zones du diagramme de Voronoï, en respectant la propriété de convexité relativement au réseau de routes, et en équilibrant au mieux la charge de travail liée à chaque secteur.

5.5 Plus court chemin à travers le réseau de routes

	Niveaux de vol									
	0-50	50-100	100-150	150-200	200-250	250-300	300-350	350-400	400-450	450-500
0° – 9°	252	1322	1375	4414	2612	6680	8626	9967	258	12
10° – 19°	243	333	957	1362	1070	2215	2173	2822	88	7
20° – 29°	109	334	995	1216	737	1454	1835	1980	44	4
30° – 39°	292	272	859	737	349	1492	1091	1066	42	3
40° – 49°	35	326	422	622	227	473	334	342	9	0
50° – 59°	24	319	284	315	108	347	263	217	5	1
60° – 69°	183	60	205	402	67	119	45	54	2	1
70° – 79°	30	155	340	226	60	56	48	24	1	0
80° – 89°	131	167	199	327	49	52	47	42	1	0
90° – 99°	46	59	49	150	9	26	12	13	0	0
100° – 109°	58	128	95	74	3	4	7	1	0	0
110° – 119°	28	53	63	37	2	0	3	0	0	0
120° – 129°	18	110	66	73	14	18	0	1	0	0
130° – 139°	9	30	19	9	3	2	0	0	0	0
140° – 149°	17	24	9	19	13	4	0	1	0	0
150° – 159°	48	184	55	4	0	1	1	0	0	0
160° – 169°	224	69	7	13	0	1	0	1	0	0
170° – 179°	162	22	100	14	12	0	0	0	0	0
180°	7	34	0	3	1	3	0	0	0	0

TAB. 5.1 – Angles entre segments de route successifs (plans de vol du 21/05/1999).

Jusqu'à présent, nous avons défini un maillage de points et des segments de routes. Nous allons maintenant essayer d'affecter à chaque flux la route la plus directe possible à travers le réseau, sans toutefois chercher à séparer les flux (ce qui sera tenté dans le chapitre 6 et les suivants), en utilisant un algorithme de recherche du plus court chemin.

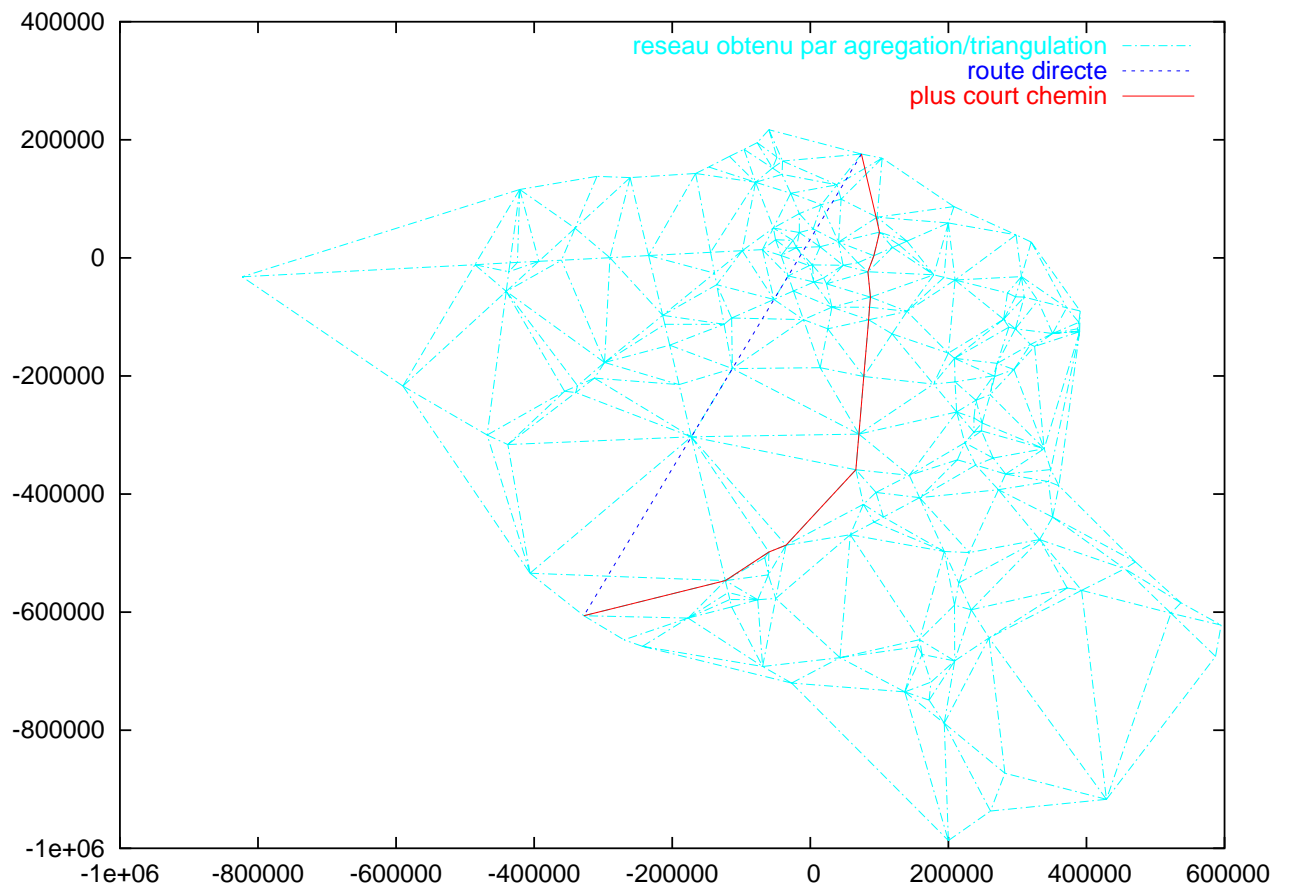


FIG. 5.7 – Plus court chemin à travers un réseau (flux de plus de 20 avions ; trafic du 21/05/99 ; projection stéréographique centrée sur Paris ; l'unité est le mètre).

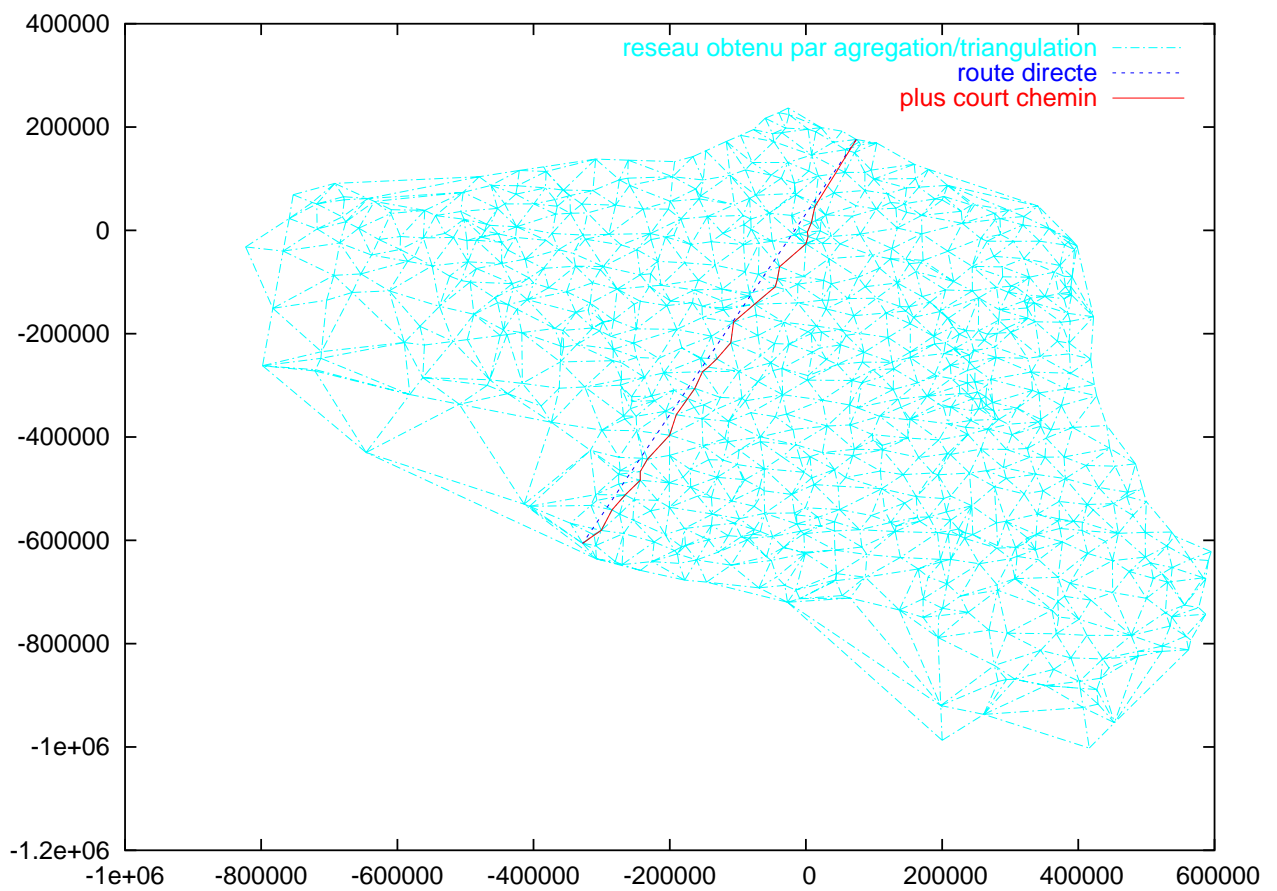


FIG. 5.8 – Plus court chemin, sans contrainte angulaire, à travers un réseau généré avec le trafic total du 21/05/99 (projection stéréographique centrée sur Paris ; l'unité est le mètre).

L'algorithme A^* , présenté au chapitre 3, dans la section 3.5.2, permet de trouver le chemin de coût minimal à travers un réseau. Cet algorithme explore un arbre d'états, avec une fonction de coût associée aux transitions entre états et une fonction heuristique permettant d'estimer (en le minorant) le coût du chemin restant à parcourir entre l'état courant et l'état final. Ici, un état est simplement un point du réseau, et l'arbre d'états se construit en prenant comme fils de l'état courant, les points voisins sur le réseau. Le coût d'une transition entre l'état courant et un état fils est la distance horizontale entre le point courant et le point voisin considéré. L'heuristique sera simplement la distance entre le point considéré et le point d'arrivée du flux. Elle minore bien le coût de n'importe quel parcours sur le réseau permettant de joindre ce point.

Quelques contraintes utiles ont également été ajoutées dans la génération des états fils : on impose un angle maximum entre deux segments consécutifs de la route parcourue, ainsi qu'un angle maximum entre la route directe (entre origine et destination) et une navigation directe entre le point courant et la destination. Ces contraintes d'angles restreignent l'espace d'états. Elles se justifient d'une part par les performances de virage limitées des avions en croisière, et d'autre part du fait qu'on ne désire pas rallonger les trajectoires de façon excessive, ni arriver sur le point de destination selon un cap totalement différent de celui de la route directe.

Afin d'avoir une idée des angles que l'on peut s'autoriser entre segments de routes successifs, on s'intéresse au réseau de routes actuel. Le tableau 5.1 montre la distribution des angles de virage pour les vols du 21/05/1999, suivant les routes déclarées dans leurs plans de vol. Les angles mesurés sont distribués par tranches d'angles et par tranches de niveaux de vol. Chaque avion passe successivement au-dessus des balises de sa route à un niveau de vol calculé selon un profil de vol nominal. Ce sont ces niveaux qui sont utilisés dans la distribution.

Au vu de ces résultats, on se fixera arbitrairement des contraintes d'angle maximum de l'ordre de 70 degrés dans la recherche du plus court chemin dans notre réseau.

L'algorithme A^* est employé successivement pour chacun des flux, dans un ordre qui importe peu, puisqu'on ne cherche pas à contourner les autres flux. La figure 5.7 montre un exemple d'un tel chemin sur le réseau obtenu à partir de l'agrégation des points de croisement des flux de plus de 20 avions. La route directe est également représentée.

Cet exemple, avec des flux de plus de 20 avions, illustre également un problème dans la conception du réseau de routes : le chemin trouvé peut s'écarter sensiblement de la route directe. En effet, les points de croisement du réseau sont trop dispersés dans certaines zones d'espace, et donc le réseau est trop diffus et ne permet pas toujours de trouver un chemin très direct répondant aux contraintes d'angle que l'on s'est fixé.

Afin d'augmenter nos chances de trouver un chemin plus direct, tout en répondant aux contraintes angulaires imposées, on peut chercher à créer un maillage de points plus uniformément répartis. Une solution possible, qui n'a pas été testée, consisterait à ajouter en entrée du processus d'agrégation, en plus des points de croisement entre routes directes, un maillage régulier de points, de façon à imposer une densité minimale de centres d'agrégation. Une autre solution consiste à construire le réseau à partir de tous les flux (cf figure 5.1), le nombre de points de croisement initiaux étant alors beaucoup plus important et mieux réparti que si l'on se restreint à un sous-ensemble des flux. Le résultat est visible sur la figure 5.8, ainsi que le plus court chemin pour le même flux que dans l'exemple de la figure 5.7. Ce plus court chemin est calculé avec une contrainte angulaire maximale de 70 degrés, et l'angle maximum constaté entre deux segments consécutifs du chemin trouvé est alors de 38 degrés, ce qui est satisfaisant. Il n'existe pas de solution avec une contrainte de 30 degrés pour cet avion et ce réseau : il faut relâcher la contrainte à 31 degrés.

Bien que nous ayons densifié le réseau de points, on peut constater des rallongements de trajectoire importants par rapport à la trajectoire directe, si l'on prend des valeurs trop faibles pour la contrainte

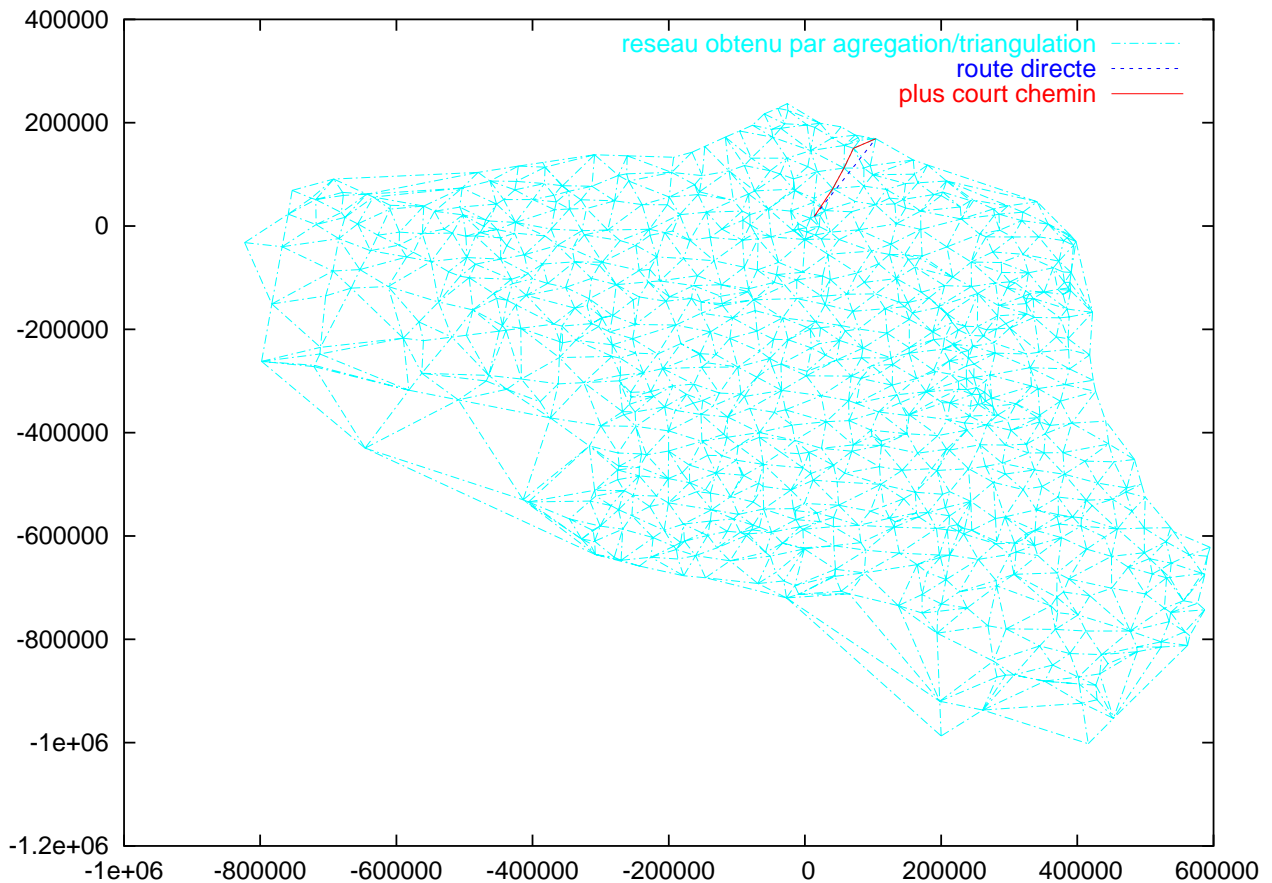


FIG. 5.9 – Exemple 2 de plus court chemin, sans contrainte angulaire (trafic total du 21/05/99 ; projection stéréographique centrée sur Paris ; l'unité est le mètre).

d'angle.

L'exemple des figures 5.9, 5.10, et 5.11 est à ce titre frappant. Le plus court chemin sans contrainte angulaire est représenté sur la figure 5.9. L'angle maximum rencontré est alors de 41 degrés. La figure 5.10 illustre le plus court chemin entre les deux mêmes points, mais avec une contrainte d'angle de 39 degrés, et la figure 5.11 avec une contrainte de 36 degrés. Il n'y a pas de solution avec une contrainte de 35 degrés.

5.6 Conclusion sur la construction du réseau de routes

Commençons par récapituler les différentes phases ayant permis de construire un réseau de routes. Un processus en trois étapes a été appliqué au nuage initial des points de croisement entre routes directes :

- une agrégation des points de croisement, permettant à la fois de diminuer le nombre de points d'intersection du réseau, et de les espacer,
- la redéfinition des segments de routes par une triangulation de Delaunay,
- la recherche (avec un A^*) pour chaque couple origine-destination d'un plus court chemin respectant une contrainte d'angle maximum entre segments successifs de route.

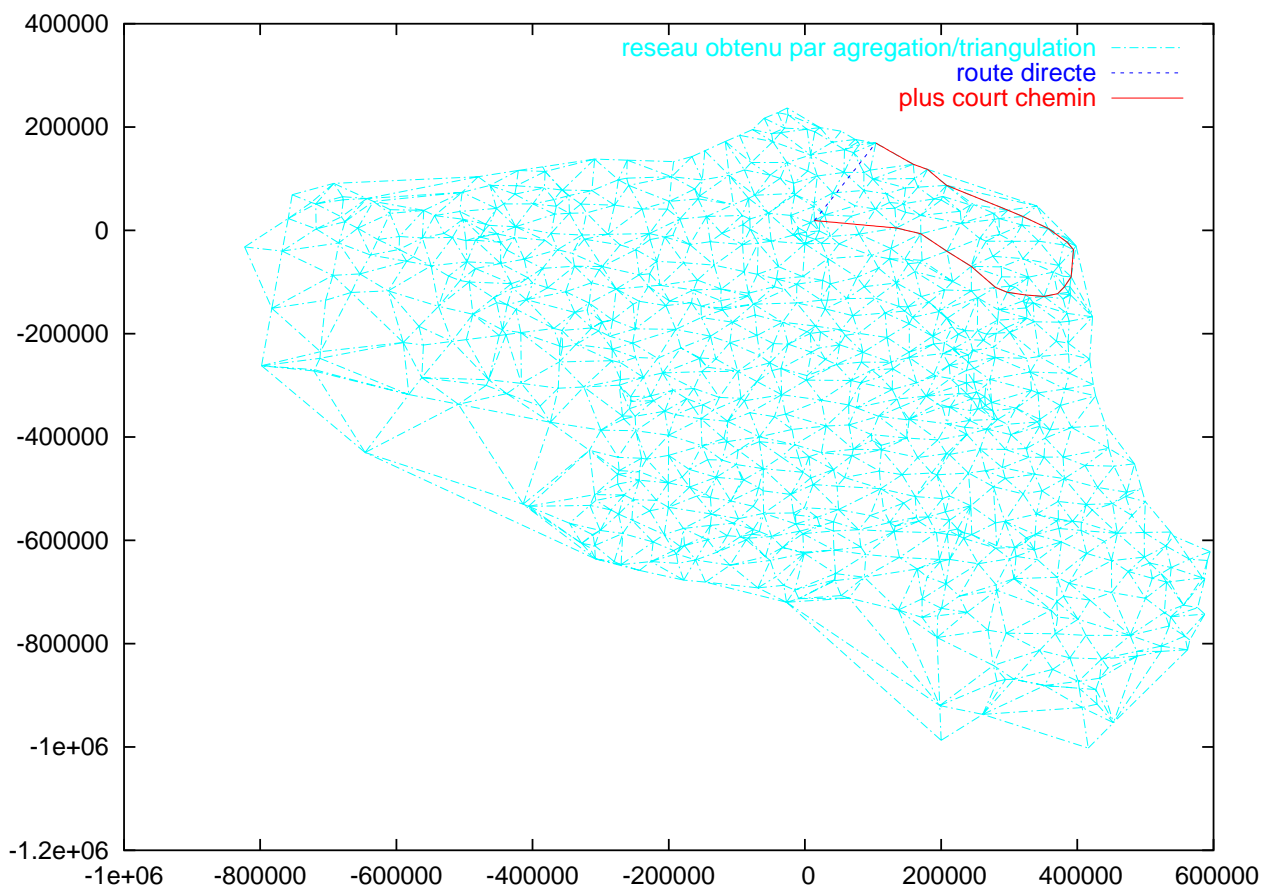


FIG. 5.10 – Exemple 2 de plus court chemin, avec une contrainte de 39 degrés maximum (trafic total du 21/05/99 ; projection stéréographique centrée sur Paris ; l'unité est le mètre).

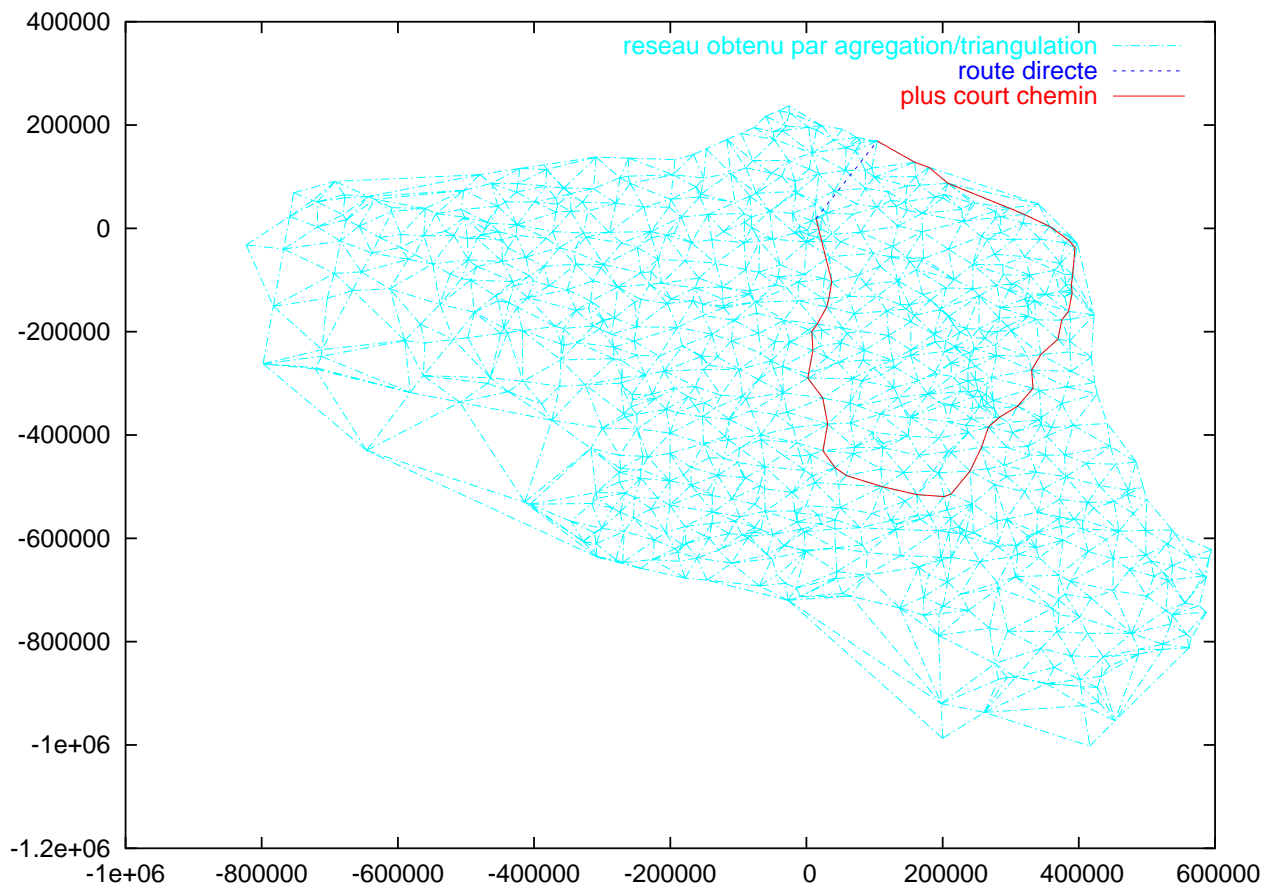


FIG. 5.11 – Exemple 2 de plus court chemin, avec une contrainte de 36 degrés maximum (trafic total du 21/05/99 ; projection stéréographique centrée sur Paris ; l'unité est le mètre).

Au passage, le diagramme de Voronoï, dual de celui de Delaunay de la deuxième étape, a permis de définir des mini-secteurs autour de chaque point d'intersection du réseau. L'algorithme de S. Fortune a été implanté (en langage Ocaml) et utilisé pour obtenir ces deux diagrammes.

Concernant le processus d'agrégation, nous avons vu qu'il était possible d'obtenir un réseau à l'issue de cette première étape, par déformation des routes directes initiales, en remplaçant chaque point de croisement par le barycentre de l'agrégat auquel il appartient. Cependant, les routes ainsi créées peuvent éventuellement se croiser en d'autres points que les barycentres des agrégats. Il faut alors appeler récursivement le processus d'agrégation, en réintroduisant les nouveaux points de croisement, jusqu'à atteindre un réseau sans nouveaux points de croisements. Le problème est que rien n'assure de converger vers un tel réseau, même si cela est le cas dans la pratique, pour les exemples traités. Par ailleurs, les routes trouvées comportent des angles trop aigus, qui ne sont pas compatibles avec les performances de virage des avions. Une bonne solution pour éviter ce problème consiste à tenir compte de ces angles dans le processus d'agrégation, comme dans [Meh00]. Une autre solution a été choisie, consistant à reconstruire le maillage puis à affecter à chaque couple (origine, destination) la route la plus courte à travers ce maillage. Une autre constatation que l'on peut faire sur le résultat de la première étape est la suivante : selon l'ensemble initial de routes directes que l'on choisit, il peut y avoir des zones de très faible densité dans le nuage de points résultant de l'agrégation. La conséquence est un allongement inutile des itinéraires passant par ces zones peu denses, comme sur la figure 5.7 où le réseau est construit à partir de routes directes correspondant aux flux de plus de 20 avions par jour. La solution qui a été adoptée a été de considérer un nuage initial de points plus important et mieux réparti géographiquement, en prenant les points de croisement de toutes les routes directes. Une autre solution à ce problème, et qui reste à explorer, consisterait à ajouter au nuage de points initial un maillage régulier de points.

La définition des segments de routes entre les points d'agrégation, par la triangulation de Delaunay, et celle des mini-secteurs associés du diagramme dual de Voronoï, fournissent les résultats auxquels on s'attend sur le plan mathématique. Toutefois, selon la répartition des points, un segment de route reliant deux points de mini-secteurs adjacents peut éventuellement traverser un troisième secteur adjacent aux deux premiers. Or, du point de vue de l'exploitation opérationnelle, il est impératif qu'un tel segment de route ne traverse que la frontière reliant les deux secteurs concernés. Cette non-convexité relativement aux flux inter-secteurs, ainsi que la taille réduite des secteurs, rend cette sectorisation inexploitable telle quelle. On entrevoit cependant une solution consistant à regrouper ces mini-secteurs pour en faire des secteurs exploitables, en adaptant les méthodes présentées au chapitre 4 sur les regroupements de secteurs.

A propos de la recherche des itinéraires les plus courts, on constate que, pour peu que l'on se fixe un angle maximum autorisé (entre segments successifs de route) un peu trop faible, on peut obtenir des chemins assez détournés pour joindre deux points. Le critère de densité sur les points de croisement, que l'on avait choisi pour rendre le réseau intelligible à des contrôleurs humains, entre ici en conflit avec la nécessité d'avoir des cheminements les plus directs possibles pour les avions.

Pour conclure plus globalement sur la construction du réseau de routes, disons qu'à ce stade, on obtient un réseau planaire et une sectorisation associée, pour lesquels on entrevoit plusieurs possibilités d'améliorations. Cependant, le principal reproche que l'on peut faire à l'approche adoptée dans ce chapitre est qu'elle ne s'intéresse pas du tout à la dimension verticale. Nous laisserons donc inexploitées pour l'instant les pistes entrevues, pour nous focaliser dans le chapitre suivant sur la séparation verticale des flux.

Chapitre 6

Allocation de niveaux de croisière, et séparation des segments évolutifs

6.1 Introduction

Il existe de nombreuses publications sur la régulation du trafic par allocation dynamique de créneaux au décollage ou de routes alternatives ([BP00], [DO97], [BBR01]), ou encore traitant de problèmes de planification des vols à travers un espace congestionné ([Jar03], [vKHHK95]), en utilisant une variété de méthodes déterministes ou stochastiques. La résolution de ces problèmes n'est pas le sujet de ce chapitre, dans lequel on s'intéressera à un réseau statique de trajectoires.

Le chapitre précédent traitait de la construction, au niveau stratégique, d'un réseau de routes compréhensible par l'être humain, et accessoirement de la définition d'une sectorisation associée, ainsi que de la recherche de la route la plus directe entre deux points de ce réseau. Des problèmes analogues avait déjà été abordés dans [Meh00], en utilisant également des diagrammes de Voronoï, et des méthodes d'agrégation différentes. Bien que très intéressante, cette approche essentiellement bi-dimensionnelle ne tient pas compte des évolutions verticales du trafic : le résultat est un réseau dans le plan.

Dans [Let98], ou encore [BB02], des techniques de coloration de graphe sont utilisées pour allouer des niveaux de vol séparés à des avions volant sur des routes directes, afin d'assurer la séparation verticale pendant le vol en croisière. Il paraît intéressant d'évaluer l'intérêt de cette approche au moyen de simulations de trafic, en cherchant à estimer la diminution du nombre de conflits qu'elle peut apporter, ce qui constituera l'objet de la première partie de ce chapitre. Les segments de montée ou de descente des trajectoires ne sont cependant pas pris en compte, mais cette approche constitue déjà un bon début dans la séparation en 3D des flux d'avions.

Dans l'étude TOSCA [MGZK98], un concept intéressant de routes directes entre les principales TMAs européennes est évalué au travers d'analyses statistiques et de simulations de trafic. L'idée est de transférer sur ce réseau de routes prioritaires et non-conflituelles un certain pourcentage du trafic total. Les avions volant sur ces routes seraient prioritaires sur le reste du trafic. Un séquençement temporel au départ permettrait de résoudre les conflits entre avions suivant un même route. Ces vols n'entreraient donc plus dans le processus d'allocation de créneaux au décollage. L'étude montre que le fait de retirer ainsi un pourcentage même modeste du trafic du processus d'allocation de créneaux pourrait diminuer considérablement les retards au décollage, à condition que l'impact du nouveau réseau de routes prioritaires sur la capacité totale du système ne soit pas trop important. Seule la séparation des routes dans le plan est évoquée : les croisements de routes sont soit interdits, soit

autorisés de façon très limitée. Seul un nombre limité de flux peut ainsi être traité, et les gains en terme de réduction du nombre de conflits détectés en simulation ne sont pas très probants.

Dans ce chapitre, nous allons donc tout d'abord évaluer concrètement l'intérêt de séparer verticalement les flux. L'algorithme de coloration de graphe, développé par N. Barnier et P. Brisset ([BB02]), est utilisé pour allouer aux avions des niveaux de vol en croisière séparés verticalement pour les flux qui se croisent dans le plan. Le simulateur de trafic aérien CATS/OPAS (présenté notamment dans les articles [DAB97], [DABM97], [DAG01]) est utilisé pour comparer une journée de trafic de référence avec un trafic modifié, où la croisière de chaque avion se fait au niveau de vol alloué. Le critère retenu pour estimer le gain de l'allocation de niveaux est le nombre de conflits détectés par le simulateur, sur des trajectoires directes. Les phases évolutives des vols ne sont à ce stade toujours pas séparées.

Dans un deuxième temps, on cherchera à séparer également les segments évolutifs des vols, en adaptant l'algorithme A^* utilisé en 5.3. Dans cette section, il était utilisé pour trouver la route la plus directe entre deux points dans un réseau. Ici, il nous faudra trouver des trajectoires 3D qui n'interfèrent pas, tout en minimisant les écarts par rapport aux trajectoires nominales en route directe avec une phase de croisière au niveau alloué par l'algorithme de coloration de graphe.

L'utilisation potentielle des méthodes proposées dans ce chapitre se situe tant au niveau stratégique, pour la définition d'itinéraires 3D alloués à chaque flux de trafic important, qu'au niveau tactique, où ces méthodes pourraient être employées avec un horizon temporel de quelques heures, dans des tâches de pré-planning.

6.2 Allocation de niveaux de croisière, par une méthode de Coloration de Graphe

Le problème traité dans cette partie consiste à allouer un niveau de croisière à chaque flux de trafic, de façon à ce que la phase de croisière de chaque vol soit libre de tout conflit avec les vols en croisière des autres flux principaux. Le nombre de niveaux alloués doit être aussi petit que possible, et le niveau de vol alloué doit être aussi proche que possible du niveau le plus demandé sur le flux considéré.

Notons que le but que l'on se fixe n'est pas de chercher une trajectoire sans conflit pour chaque vol, mais uniquement pour les vols appartenant aux principaux flux de trafic. Par ailleurs, les conflits pouvant survenir dans les phases de montée ou de descente ne pourront pas être évités en séparant uniquement les niveaux de croisière. La séparation des segments évolutifs de trajectoire sera abordée dans la section 6.3, puis dans les chapitres suivants, avec des approches différentes.

6.2.1 Description succincte de l'algorithme de N. Barnier et P. Brisset

L'algorithme développé au LOG par Nicolas Barnier et Pascal Brisset a été utilisé pour allouer des niveaux aux flux au dessus d'une certaine taille. Le problème d'allocation est représenté par un graphe dont les noeuds sont les flux, et dont les arêtes représentent les intersections de flux. Une allocation valide assure que deux flux interférant (dans le plan horizontal) auront des niveaux de vol en croisière différents. L'objectif est également de minimiser le nombre de niveaux alloués. Ce problème est équivalent au problème classique de coloration de graphe, dont on sait qu'il est NP-difficile, ce qui implique souvent l'utilisation de techniques approchées, notamment sur les graphes aléatoires et de grande taille ([Lei79], [MS86]). Cependant, les algorithmes énumératifs complets utilisant le Branch & Bound (B&B) peuvent aussi être efficaces (voir [Bré79]) pour des problèmes plus structurés, issus d'applications réelles ou de taille moindre.

Angle	--	-↑	-↓	↑↑	↓↓	↑↓	Mix	Total
0-9	176	45	56	41	26	1	7	352
10-19	16	3	1	13	3	0	0	36
20-29	34	1	5	4	2	0	0	46
30-39	39	6	1	3	1	0	0	50
40-49	38	0	3	0	0	0	0	41
50-59	35	3	2	0	0	0	0	40
60-69	28	2	2	0	0	0	0	32
70-79	41	1	3	0	0	0	0	45
80-89	24	2	2	0	0	1	0	29
90-99	17	4	2	0	0	0	0	23
100-109	13	2	0	0	0	1	0	16
110-119	16	0	2	0	0	0	0	18
120-129	16	13	0	0	0	0	0	29
130-139	10	7	6	1	0	1	0	25
140-149	25	5	1	0	0	0	0	31
150-159	7	6	4	0	0	3	0	20
160-169	14	12	5	0	0	2	0	33
170-180	60	22	8	0	1	28	0	119
Total	609	134	103	62	33	37	7	985
Percent.	61.8	13.6	10.5	6.3	3.4	3.8	0.7	100

TAB. 6.1 – Distribution des conflits par angle de convergence et par type (Croisière : – ; Montée : ↑ ; Descente : ↓), pour des avions volant sur une route directe à leur niveau de vol initialement demandé (REF_SUBSET)

L’algorithme proposé pour le coloriage exact des noeuds par N. Barnier et P. Brisset dans [BB02] consiste à associer des techniques de programmation par contraintes avec un algorithme glouton de recherche de *cliques*. Pour le problème d’allocation de niveaux, une *clique* sera un ensemble de routes toutes sécantes deux à deux : autant de niveaux différents seront alloués à chacune de ces routes. L’utilisation d’un B&B pour rechercher la *clique maximum* (cf [Cou97]) permet de plus, lorsque la taille du problème l’autorise, d’obtenir une borne inférieure du problème de coloriage de graphe et éventuellement de prouver l’optimalité, lorsque cette borne est atteinte.

Notons r_i la route directe dans le plan horizontal (en projection stéréographique) pour le flux i , et tw_i la fenêtre de temps pendant laquelle cette route est utilisée. Afin de détecter les intersections de flux, on définit un ruban autour de chaque route r_i , d’une largeur S_h (en milles nautiques), où S_h est la valeur de la norme de séparation. Deux routes r_i et r_j sont sécantes lorsque les deux rubans définis autour de ces routes s’intersectent ou sont superposés. Il y a cependant quelques exceptions à cette règle : par exemple, deux flux divergents ayant la même origine n’interfèrent pas si cette origine est un aéroport ou le dernier point d’une procédure de départ d’un aéroport. De même pour les flux convergeant sur un même aéroport d’arrivée. Deux flux i et j interfèrent si leurs routes horizontales r_i et r_j interfèrent, et si leurs fenêtres temporelles tw_i et tw_j se recoupent.

Angle	--	-↑	-↓	↑↑	↓↓	↑↓	Mix	Total
0-9	286	40	51	31	28	2	11	449
10-19	5	5	3	13	7	0	0	33
20-29	1	3	6	4	4	0	0	18
30-39	1	2	2	4	1	0	0	10
40-49	1	4	1	1	3	0	0	10
50-59	0	2	3	0	0	0	0	5
60-69	0	3	1	0	0	0	0	4
70-79	0	1	3	0	0	0	0	4
80-89	2	4	3	0	0	1	0	10
90-99	2	5	2	0	0	0	0	9
100-109	4	0	5	0	0	0	0	9
110-119	0	3	3	0	0	0	0	6
120-129	2	9	1	0	0	0	0	12
130-139	8	6	7	0	0	1	0	22
140-149	19	3	3	0	0	0	0	25
150-159	14	7	2	0	0	1	0	24
160-169	6	8	5	0	0	1	0	20
170-180	40	24	8	0	0	22	0	94
Total	391	129	109	53	43	28	11	764
Percent.	51.2	16.9	14.3	6.9	5.6	3.7	1.4	100

TAB. 6.2 – *Distribution des conflits par angle de convergence et par type (Croisière : – ; Montée : ↑ ; Descente : ↓), pour des avions volant sur une route directe au niveau de croisière alloué par l’algorithme (ALLOC_SUBSET)*

6.2.2 Application aux flux dans l’espace aérien français

L’espace aérien supérieur et l’espace aérien inférieur sont considérés séparément. Pour ce qui concerne l’espace inférieur (vols évoluant à des niveaux de croisière entre 6000 et 19500 pieds au dessus de l’isobare 1013 hPa), l’algorithme a alloué 10 niveaux de vols aux flux comprenant plus de 3 avions dans la journée. Dans l’espace supérieur, le problème d’allocation a été résolu pour des flux de plus de 10 avions, avec un nombre minimal de 17 niveaux de vol. Les tailles retenues pour les deux problèmes (3 ou 10 avions minimum par flux) permettent d’obtenir un nombre réaliste de niveaux de vol. Cependant, l’algorithme est capable de résoudre des problèmes nettement plus difficiles : par exemple, il trouve une solution optimale pour les flux en espace aérien ayant plus de 3 avions dans la journée, avec 22855 intersections de flux, en allouant 34 niveaux. Ce nombre n’est pas compatible avec la réalité de la structure de l’espace aérien : il n’y a pas autant de niveaux de vol exploitables.

Le jour de trafic choisi comprenait 7902 vols répartis sur 1736 flux. 6104 vols (répartis sur 906 flux) demandaient un niveau de vol en espace supérieur, et 1798 vols (941 flux) intéressaient seulement l’espace inférieur. Le problème d’allocation résolu en espace inférieur (flux de plus de 3 avions) concerne 837 vols (171 flux), et celui en espace supérieur 4131 vols (146 flux). L’algorithme utilisé a donc permis d’allouer des segments de trajectoire en croisière n’interférant pas entre eux pour près de 63 % du trafic.

Angle	--	-↑	-↓	↑↑	↓↓	↑↓	Mix	Total
0-9	232	72	75	63	34	1	7	484
10-19	38	13	11	33	11	2	0	108
20-29	60	12	14	10	10	0	0	106
30-39	59	5	8	6	3	0	0	81
40-49	75	6	8	0	1	2	0	92
50-59	70	4	6	0	0	2	0	82
60-69	65	8	7	1	0	0	0	81
70-79	83	6	13	0	0	0	0	102
80-89	55	12	7	0	0	1	0	75
90-99	56	10	5	1	0	0	0	72
100-109	39	13	9	1	0	1	0	63
110-119	50	8	3	2	1	6	0	70
120-129	35	16	8	0	0	3	0	62
130-139	41	15	9	2	0	6	0	73
140-149	39	17	11	0	0	6	0	73
150-159	28	16	7	0	0	8	0	59
160-169	34	21	10	0	0	17	0	82
170-180	119	45	20	1	1	61	0	247
Total	1178	299	231	120	61	116	7	2012
Percent.	58.5	14.9	11.5	6.0	3.0	5.8	0.3	100

TAB. 6.3 – Distribution des conflits par angle de convergence et par type (Croisière : – ; Montée : ↑ ; Descente : ↓), pour TOUS LES AVIONS volant sur une route directe à leur niveau de vol initialement demandé (REF)

6.2.3 Validation par simulation de trafic

Les simulations sur cette journée de trafic avec le simulateur en temps accéléré CATS/OPAS montrent les gains apportés par cette allocation de niveaux. Une présentation de CATS/OPAS et de ces différents modules peut être trouvée dans plusieurs articles ([DAB97], [DABM97], [DAG01]). Ces principales fonctions sont la détection des conflits et leur résolution, pour des avions volant soit en route directe, soit en route standard. Le simulateur utilise les plans de vols enregistrés par les systèmes opérationnels.

Pour notre problème, seule la détection de conflits a été exploitée, avec quatre jeux de trafic. Le trafic de référence est celui du 21 mai 1999, chaque avion volant sur une route directe avec une phase en croisière au niveau de vol demandé par l'opérateur aérien (RFL : *Requested Flight Level*). Le deuxième trafic reprend les caractéristiques du trafic de référence, à ceci près que la phase de croisière se fait au niveau de vol alloué par l'algorithme de coloriage de graphe, sauf si un tel niveau n'existe pas, auquel cas elle se fait au niveau demandé (RFL). Les troisième et quatrième jeux reprennent les deux premiers en éliminant les flux pour lesquels aucun niveau de vol n'a été alloué. Voici une description synthétique des jeux de trafic utilisés :

REF : Trafic de référence : route directe, croisière au niveau de vol demandé (RFL).

ALLOC : Trafic modifié : route directe, croisière au niveau de vol alloué par l'algorithme, ou par défaut au niveau de vol demandé (RFL).

REF_SUBSET : Sous-ensemble du trafic de référence (REF), ne comprenant que les avions appar-

Angle	--	- ↑	- ↓	↑↑	↓↓	↑↓	Mix	Total
0-9	318	79	71	52	41	2	11	574
10-19	17	19	8	31	16	2	0	93
20-29	13	15	20	10	11	0	0	69
30-39	7	7	7	8	2	0	0	31
40-49	12	8	10	1	3	1	0	35
50-59	10	5	10	0	0	1	0	26
60-69	21	7	7	1	0	0	0	36
70-79	22	11	11	0	0	0	0	44
80-89	21	11	9	0	0	1	0	42
90-99	19	13	7	1	0	0	0	40
100-109	22	10	12	1	0	0	0	45
110-119	18	9	9	2	1	4	0	43
120-129	10	22	5	0	0	3	0	40
130-139	24	11	14	1	0	5	0	55
140-149	31	12	16	0	0	5	0	64
150-159	37	17	8	0	0	6	0	68
160-169	34	26	10	0	0	13	0	83
170-180	72	49	32	1	0	50	0	204
Total	708	331	266	109	74	93	11	1592
Percent.	44.5	20.8	16.7	6.8	4.6	5.8	0.7	100

TAB. 6.4 – *Distribution des conflits par angle de convergence et par type (Croisière : – ; Montée : ↑ ; Descente : ↓), pour TOUS LES AVIONS volant sur une route directe à leur niveau de vol alloué (ALLOC)*

tenant aux flux pour lesquels un niveau a été alloué. Les avions volent au niveau demandé dans leur plan de vol (RFL).

ALLOC_SUBSET : Sous ensemble du trafic modifié (ALLOC), ne comprenant que les avions appartenant aux flux pour lesquels un niveau a été alloué.

Nous avons modifié le simulateur de façon à calculer pour chaque jeu le nombre de conflits et la distribution de ces conflits selon l'angle de convergence et la typologie. Le tableau 6.1 montre les résultats pour le jeu REF_SUBSET, avec un nombre total de 985 conflits détectés. Notons (au bas de la deuxième colonne) que plus de 60 % des conflits se produisent entre avions stables à leur niveau de croisière. Ce chiffre souligne l'intérêt de séparer les flux de trafic pendant leur phase de croisière.

Dans le tableau 6.2, qui montre les résultats en utilisant l'allocation de niveaux (pour le jeu ALLOC_SUBSET), le nombre de conflits détectés tombe à 764, soit un gain de 22,4 % par rapport au trafic de référence. Le pourcentage de conflits entre avions en phase de croisière est encore relativement important : 391 conflits en croisière sur les 764, soit 51,2 %. La plupart de ces conflits (286 sur les 391) ont un angle de convergence inférieur à 9 degrés. Il s'agit en fait de conflits *en rattrapage*, ayant lieu entre avions suivant une même trajectoire, au même niveau de vol.

Ces conflits *en rattrapage* entre avions en croisière, pour lesquels on peut faire l'hypothèse raisonnable qu'ils puissent être résolus par des techniques spécifiques de séquençage (dites *miles-in-trail*), représentent à eux seuls 37,4 % du total des 764 conflits détectés pour ALLOC_SUBSET, et 24,6 % des conflits restants, en simulant la totalité du trafic (voir les résultats pour le jeu ALLOC dans le tableau 6.4).

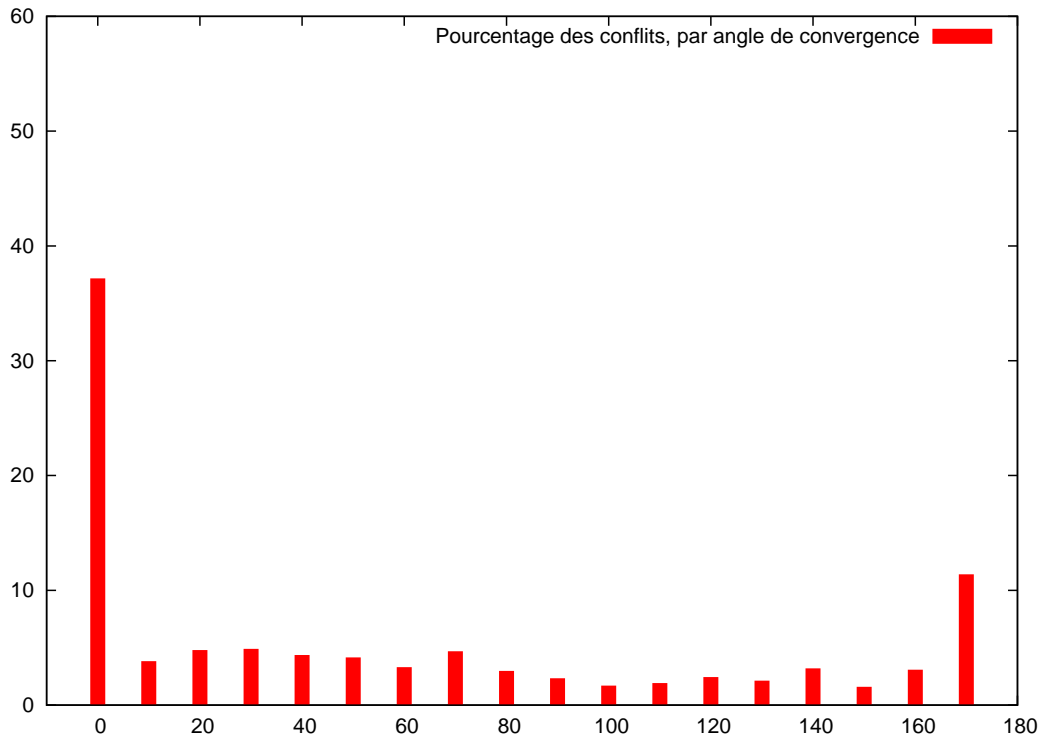


FIG. 6.1 – Distribution des conflits par angle de convergence, pour le jeu REF_SUBSET.

Les figures 6.1 et 6.2 montrent la distribution des conflits par angle de convergence pour les deux jeux de trafic REF_SUBSET et ALLOC_SUBSET. Ces graphes sont directement tirés de la dernière colonne des tableaux 6.1 et 6.2. La proportion des conflits *en face-à-face* reste approximativement constante, aux alentours des 12 %, entre REF_SUBSET et ALLOC_SUBSET. Pour le reste, la distribution est aplatie pour les angles intermédiaires, lorsqu'on utilise l'allocation de niveaux (figure 6.2), mais reste importante pour les angles compris entre 0 et 9 degrés, à cause du phénomène que nous venons d'évoquer sur les conflits en rattrapage.

Les tableaux 6.3 et 6.4 illustrent les résultats pour les configurations REF et ALLOC, prenant en compte la totalité du trafic. Le nombre total de conflits passe de 2012 pour la situation de référence à 1592 avec l'allocation de niveaux. On constate des phénomènes similaires à ceux évoqués ci-dessus pour les configurations SUBSET, mais de façon moins marquée. La diminution du nombre de conflits détectés (20,9 %), obtenue en allouant des niveaux de croisière séparés à 63 % du trafic, est donc tout à fait significative. Elle est potentiellement plus importante encore, si l'on considère que les conflits en rattrapage sur un même flux peuvent être réglés au départ par un séquençage temporel et une régulation en vitesse.

Ces bons résultats sur le trafic global, avec les jeux REF et ALLOC, doivent toutefois être tempérés par la considération suivante. Dans la configuration de trafic de référence (REF) pour le 21 mai 1999, le dispositif RVSM¹ n'était pas encore en place pour tous les avions et dans tout l'espace. Or, sans RVSM, la séparation verticale des niveaux est de 1000 pieds en dessous du FL295 (29500 pieds) et de 2000 pieds au dessus. L'algorithme d'allocation autorise une séparation de 1000 pieds dans tout

¹RVSM : *Reduced Vertical Separation Minima*, dispositif autorisant des séparations verticales de 1000 pieds au dessus du FL295, au lieu des 2000 pieds imposés précédemment.

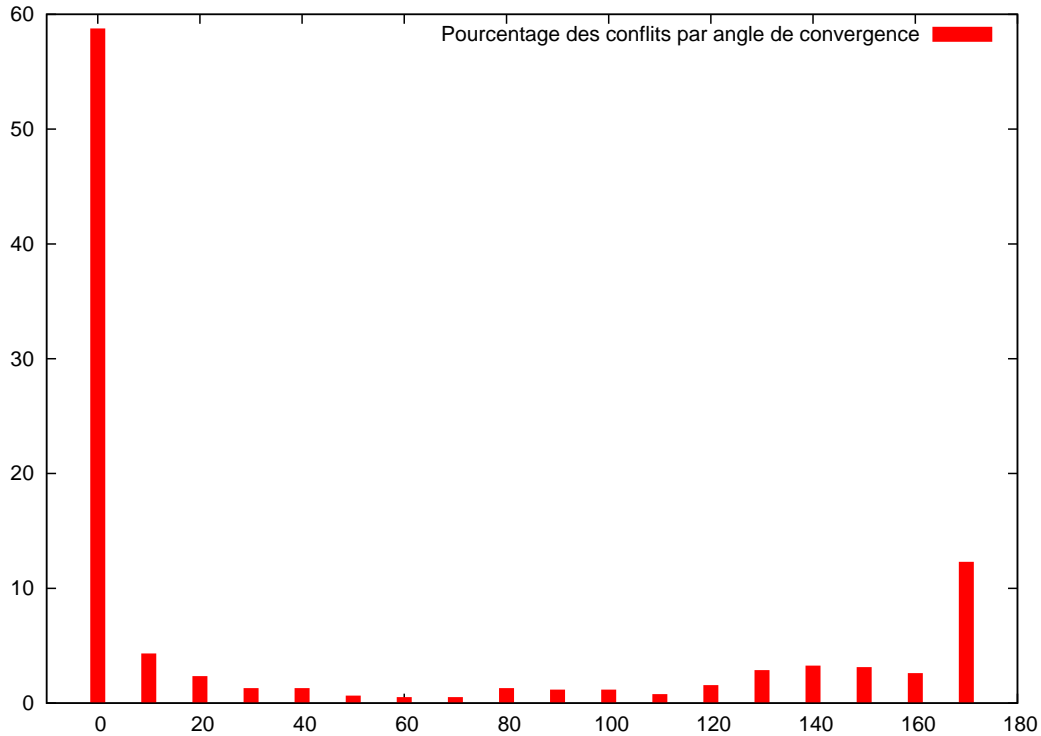


FIG. 6.2 – Distribution des conflits par angle de convergence, pour des avions volant sur une route directe au niveau de croisière alloué par l’algorithme, pour le jeu ALLOC_SUBSET.

l’espace et pour tous les flux. En conséquence, le fait de faire évoluer une partie du trafic (dans le jeu ALLOC) sur des niveaux qui n’étaient pas précédemment utilisés diminue nécessairement les conflits avec le reste du trafic appartenant aux flux non traités. Une partie des gains obtenus avec le trafic global est donc à mettre au compte de cette nouvelle répartition des niveaux.

Les résultats sur les jeux de trafic restreints (REF_SUBSET et ALLOC_SUBSET) restent par contre valables tels quels.

6.3 Séparation des trajectoires 3D, avec un algorithme A^*

Nous venons de voir que l’allocation de niveaux de croisière fournit à elle seule des résultats intéressants. Cependant, seuls les segments en phase de croisière sont séparés de façon à ne pas interférer entre eux. Dans ce qui suit, nous présentons une nouvelle approche, avec seulement des résultats préliminaires. Cette approche consiste à définir des trajectoires 3D pour les principaux flux de trafic, qui soient complètement séparées, dans les phases de croisière mais aussi dans celles de montée ou de descente.

Afin de bénéficier des bons résultats de l’algorithme de coloration de graphe, les niveaux de croisière alloués par celui-ci sont utilisés comme niveaux de vol préférentiels, pour les trajectoires que nous essayons de construire.

6.3.1 Hypothèses de départ

A ce stade, quelques hypothèses simplificatrices ont été adoptées :

- tous les avions sont du même type (A320) ;
- chaque flux est représenté par un seul *avion représentatif*, volant du point d'origine du flux au point de destination, avec comme niveau de croisière préférentiel le niveau alloué par l'algorithme présenté en section 6.2 ;
- chaque *avion représentatif* suit par défaut un profil nominal de vol, selon un modèle tabulé de performances avion ;
- on considère que tous les aéroports sont à l'altitude 0, relativement à l'isobare 1013 hPa ;
- les flux débutent et se terminent aux points d'entrée ou de sortie de l'espace aérien, ou aux points d'entrée ou de sortie des zones d'approche terminale. Les segments de trajectoire à l'intérieur des zones d'approche ne sont pas considérés.

La figure 6.3 illustre le profil vertical idéal d'un avion représentant un flux. Comme indiqué dans les hypothèses, le niveau préférentiel de croisière est produit par l'algorithme d'allocation de niveaux. Seuls les flux demandant des niveaux de vol en espace supérieur sont considérés.

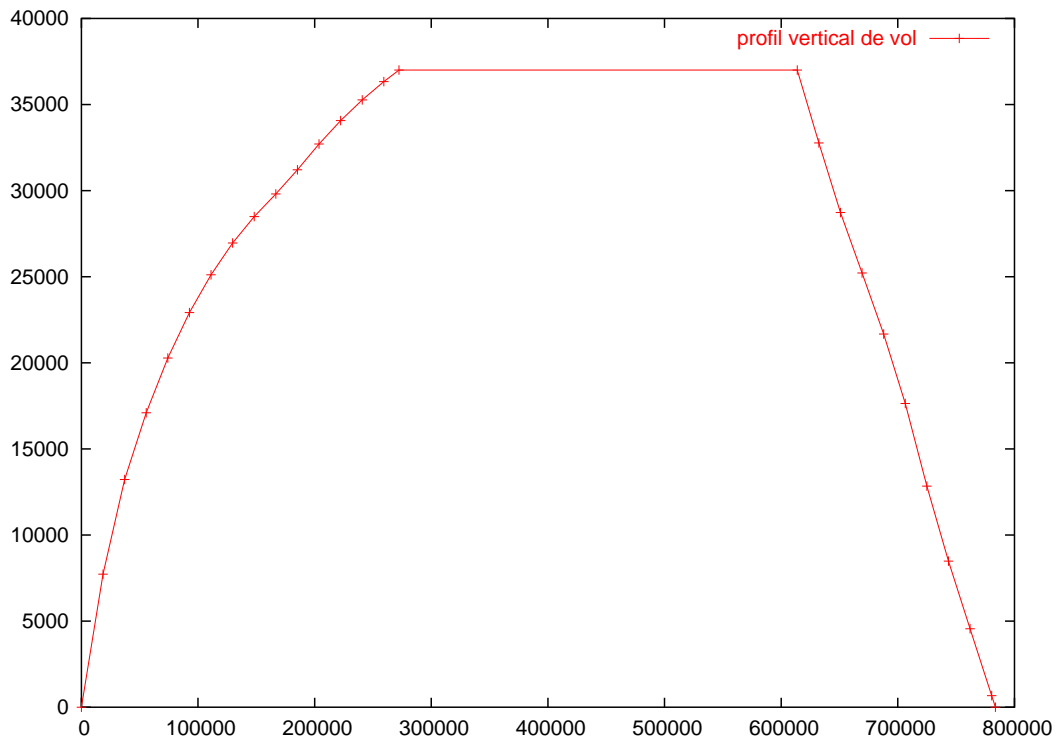


FIG. 6.3 – Exemple de profil vertical de vol pour une trajectoire nominale (non déviée), avec en abscisse la distance parcourue en mètres, et en ordonnée la hauteur, en pieds.

Pour chaque flux, les niveaux de vol en entrée et en sortie de l'espace aérien pour l'avion représentatif sont calculés en considérant l'ensemble des avions appartenant au flux. Prenons l'exemple d'un flux dont l'origine est un point de sortie de la zone d'approche de Paris. Les vols sur ce flux peuvent *a priori* provenir de n'importe lequel des aéroports de la région parisienne (bien que ce ne soit pas toujours le cas, les procédures propres à chaque aéroport se terminant souvent sur des points distincts). Dans un tel cas, nous prendrons comme niveau de vol en entrée le plus petit des niveaux de vols atteints au dessus de ce point par les avions appartenant au flux.

Bien sûr, s'il existe des flux avec des vols entrant dans l'espace français à un niveau de croisière stable, mais aussi avec des vols en montée provenant d'un aéroport proche, l'*avion représentatif* risque

de ne plus être si représentatif que cela du comportement de l'ensemble des avions du flux. Le modèle retenu n'est donc pas, à ce stade de la recherche, très réaliste. Il peut être facilement amélioré en définissant plusieurs avions représentatifs par flux, avec des hypothèses plus réalistes (altitude des aéroports, incertitudes verticales, etc...). Cependant, il faut se souvenir que notre but n'est pas d'éviter tous les conflits, ce qui est le rôle du contrôle aérien tactique, mais seulement de diminuer de façon significative le nombre de conflits, en organisant mieux les flux à un niveau stratégique. Il n'est donc pas utile de compliquer le modèle pour le moment.

6.3.2 Détection des interférences entre trajectoires

Dans le but de séparer les trajectoires des flux principaux, une grille 3D est définie. Cette grille permet de mémoriser l'occupation de l'espace par les trajectoires déjà définies. Lorsqu'on cherche à définir une nouvelle trajectoire, il suffit de regarder si le segment que l'on est en train de construire traverse des zones d'espace déjà occupées, pour savoir s'il faut retenir ce segment ou non.

La figure 6.4 montre un exemple de trajectoire d'avion à travers une telle grille. Les cubes en rouge (ou gris foncé) sont ceux traversés par la trajectoire, et ceux en bleu (gris clair) sont les voisins de ces cubes.

La figure 6.5 est une vue 3D des trajectoires nominales pour les flux de plus de 20 avions. Elle montre également les cubes d'espace pour lesquels les contraintes de séparation entre trajectoires sont violées.

Une définition adéquate de la grille et de la notion de proximité permet d'assurer que les flux sont complètement séparés. Par exemple, en choisissant une grille d'un pas horizontal de 1 mille nautique, et en définissant la proximité par "*tous les cubes dans un ruban de 10 milles nautiques de large autour de la trajectoire directe*", nous pouvons dire que si les cubes à proximité d'un segment de trajectoire ne sont pas contraints, alors ce segment sera toujours éloigné d'au moins 5 milles nautiques (la valeur de la séparation devant être assurée) de n'importe quelle autre trajectoire existante.

6.3.3 Description de l'algorithme

Nous avons utilisé un algorithme A^* pour trouver une trajectoire pour chaque flux, à travers l'espace non occupé par les trajectoires précédemment calculées pour les autres flux. Les flux sont considérés par ordre décroissant de taille (nombre d'avions par flux). De cette manière, les flux les plus importants ont les trajectoires les plus directes. Chaque trajectoire est calculée pas à pas, et peut éventuellement être déviée verticalement ou horizontalement, afin d'éviter des cubes d'espace occupés.

L'algorithme A^* , présenté au chapitre 3, section 3.5.2, fait appel à une fonction de coût pour la transition entre états, et à une fonction heuristique, qui doit estimer au mieux, tout en le minorant, le coût total des étapes restantes entre l'état courant et la solution optimale.

Dans notre problème :

- les noeuds de l'arbre sont des états. Un état est un couple (*position, niveau*) regroupant la position de l'avion dans l'espace et le niveau de vol vers lequel il évolue.
- la racine de l'arbre (l'état initial) sera alors le couple comprenant la position de départ de l'*avion représentatif* du flux considéré, et le niveau vol en croisière alloué à ce flux.
- les feuilles (noeuds terminaux) sont des états solutions, pour lesquels la position de l'avion est celle du point d'arrivée.
- chaque noeud a zéro, un, ou plusieurs fils. Pour calculer un noeud fils, on choisit un point parmi les points admissibles. La partie gauche de la figure 6.7 montre pour la trajectoire 2 les rangées successives de points admissibles définis perpendiculairement à la trajectoire directe. Lorsque

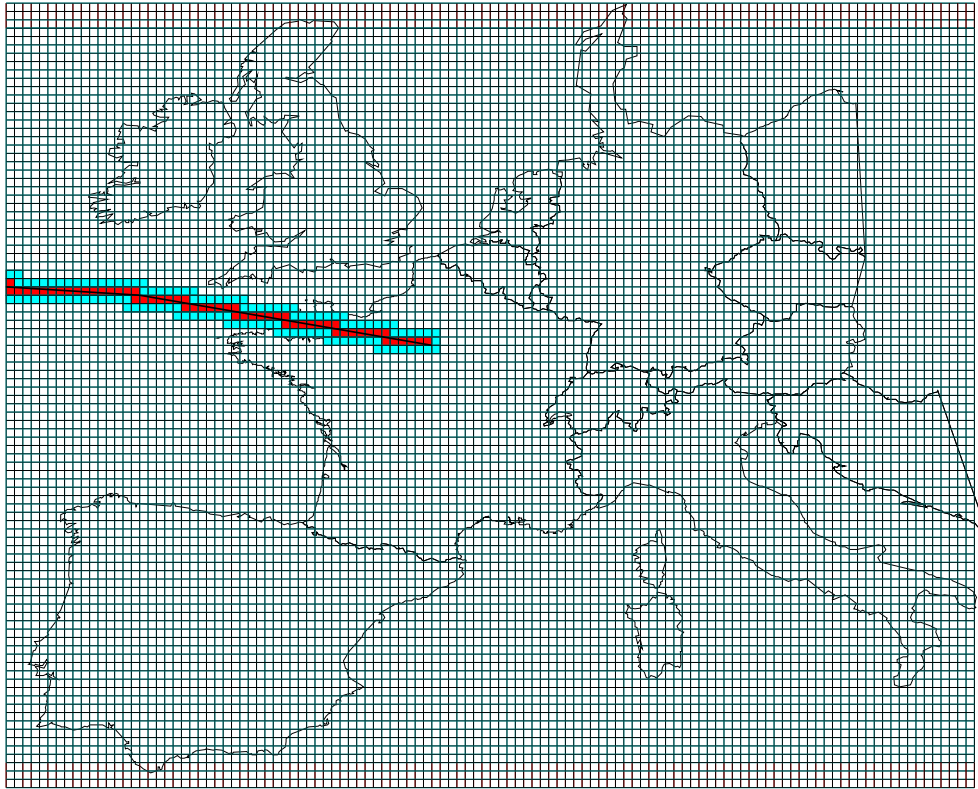


FIG. 6.4 – Exemple de trajectoire (vue en 2D) à travers la grille 3D.

le point courant est sur l'une de ces lignes, on choisit donc un point sur la ligne suivante, dans le sens de progression de l'avion. Il faut également choisir un niveau de vol en croisière si l'avion n'est pas déjà en descente finale. Ce n'est pas forcément le même que celui du noeud père. La position suivante (en 3D) est alors calculée en faisant évoluer l'avion vers ce niveau, sur le nouveau segment de route. On détermine alors les cubes de la grille 3D traversés par le nouveau segment de trajectoire. Si l'un des cubes traversés (rouge/gris foncé), ou si un cube au voisinage de ceux-ci (bleu/gris clair) est déjà occupé par une trajectoire précédemment calculée, le noeud fils potentiel est rejeté, et sinon il est retenu.

- l'arbre est parcouru en partant à chaque étape du meilleur noeud calculé précédemment. Pour ceci, l'A* gère une file d'attente à priorités (*priority queue*) qui mémorise tous les noeuds calculés. Le processus consiste à considérer le meilleur noeud (en tête de la file d'attente), à calculer les noeuds fils et à les insérer dans la file à priorités. L'insertion se fait avec une priorité dépendant du coût et de l'heuristique. Le processus est répété jusqu'à tomber sur une feuille (la solution trouvée est alors optimale) où jusqu'à ce que l'on démontre qu'aucune feuille ne peut être atteinte (il n'y a pas de solution).

Pour instancier les fonctions de coût et d'heuristique, il faut pouvoir associer un coût aux déviations par rapport à la trajectoire idéale (route directe, profil vertical nominal). Un exemple de profil nominal peut être : montée vers le niveau de vol préférentiel, puis phase de croisière, puis descente vers l'aéroport de destination. Considérons la surface entre ce profil nominal et le profil réellement suivi par l'avion. Pour une déviation verticale, avec par exemple une stabilisation pendant un certain temps à un niveau en dessous du niveau préférentiel, on voit immédiatement que plus la déviation sera importante et longue, plus la surface sera importante. C'est également valable pour les déviations

horizontales, du moins lorsqu'il y a une phase de descente : le rallongement de la trajectoire déviée décale le profil de descente par rapport au profil nominal, générant ainsi une surface représentative du rallongement de trajectoire. Toutefois ce n'est pas vrai pour les vols stables en croisière pendant toute la traversée de l'espace aérien. Pour ceux-ci, toutes les déviations horizontales auraient un coût nul. Pour pallier à ces cas particuliers, il suffit de tenir compte également de la distance parcourue.

Le coût d'une transition entre un état et le suivant sera donc une pondération de la distance horizontale entre les deux points successifs, et de la surface entre les portions des deux profils (le nominal et le réel) comprises entre les abscisses curvilignes² correspondant aux deux états.

L'heuristique sera également une pondération, avec les mêmes coefficients, de la distance horizontale du point considéré jusqu'à l'arrivée, et de la surface entre le profil nominal (la portion après le point) et un nouveau profil calculé, *en supposant que l'avion cherchera à rejoindre le profil nominal à partir du point considéré.*

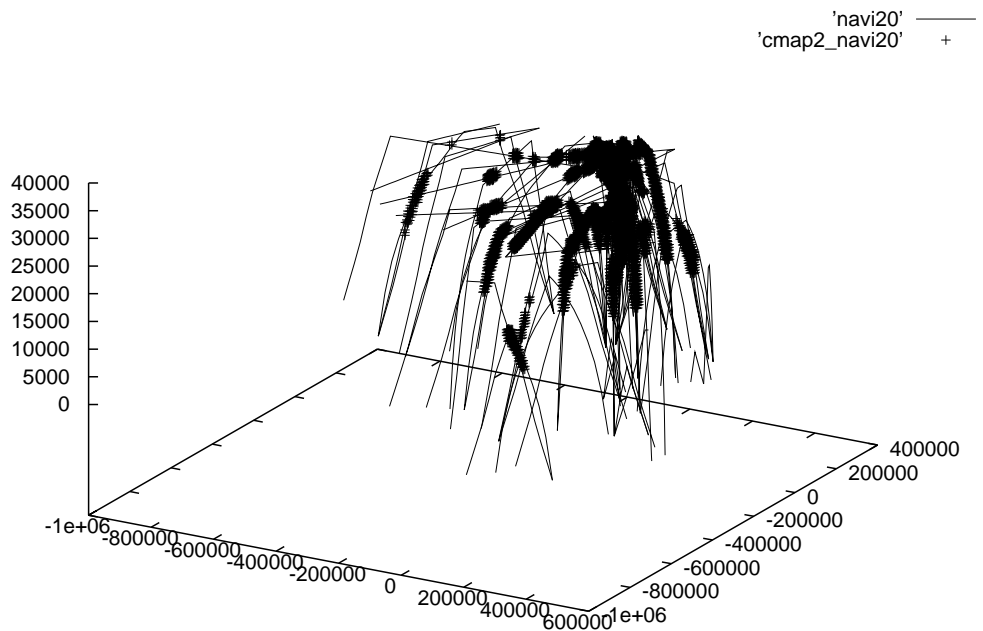


FIG. 6.5 – Trajectoires 3D directes, et violations de séparation, pour les flux de plus de 20 avions (trafic du 21 mai 1999 ; distances horizontales en mètres ; hauteurs en pieds).

Une fois que la trajectoire complète a été calculée, le chemin complet à travers la grille 3D (les cubes en rouge/gris foncé traversés par la trajectoire) est stocké dans une structure de données spécifique contenant les contraintes d'espace (c'est-à-dire les cubes traversés par les trajectoires précédemment calculées).

L'utilisation de la grille 3D et de la structure de données stockant les contraintes accroît l'efficacité de l'algorithme : il n'est pas besoin, à chaque pas de calcul, de tester la séparation entre le nouveau segment de trajectoire et chacune des trajectoires déjà calculées. Il n'y a qu'à tester si le nouveau segment traverse ou est à proximité d'un cube déjà occupé.

² L'abscisse curviligne est ici la distance depuis le point de départ.

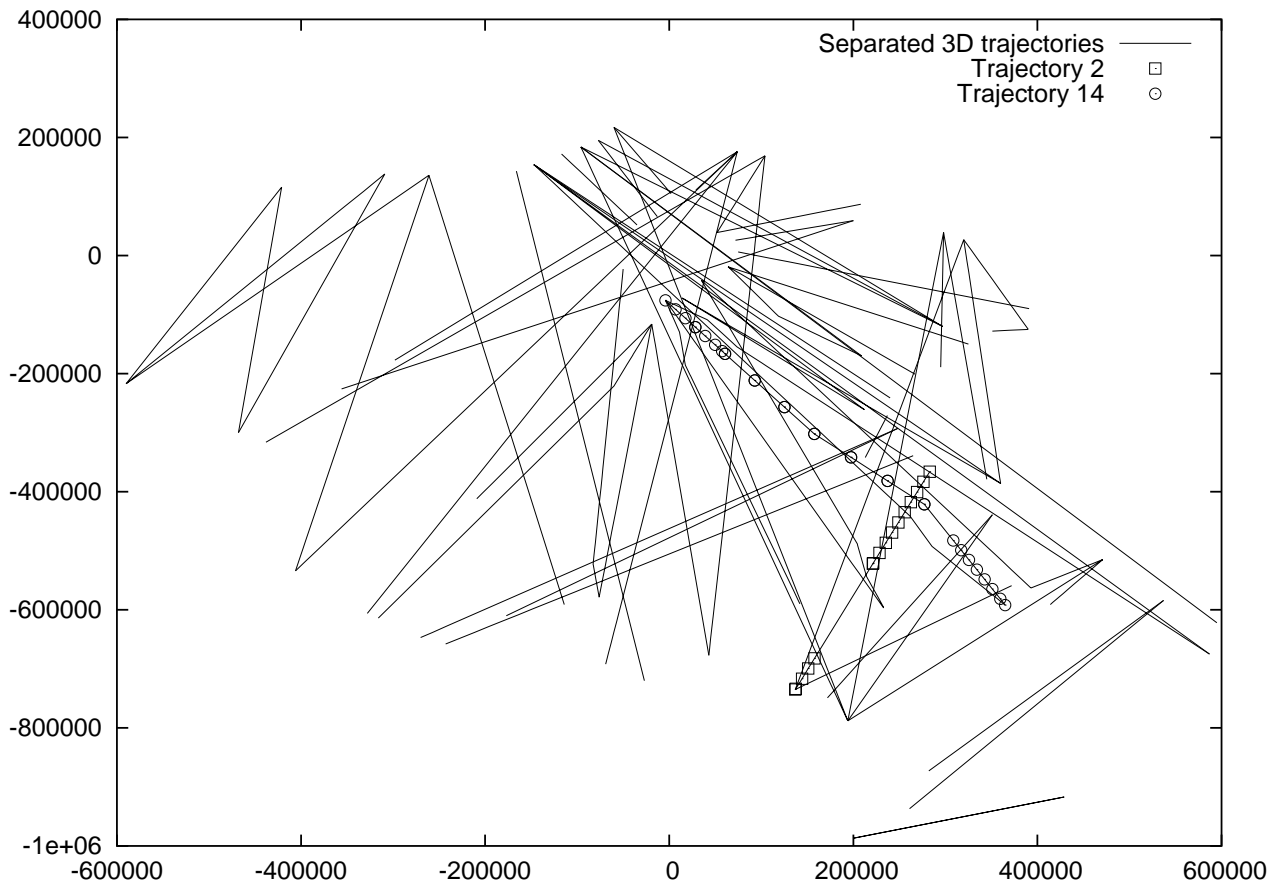


FIG. 6.6 – Vue en plan des trajectoires 3D séparées, pour les flux de plus de 20 avions au dessus de la France (trafic du 21 mai 1999 ; projection stéréographique centrée sur Paris ; l'unité est le mètre).

6.3.4 Résultats

L'algorithme a été testé avec en entrée les plans de vol enregistrés par les centres de contrôle aérien français, pour la journée du 21 mai 1999. La figure 6.6 montre les trajectoires résultantes pour les flux de plus de 20 avions dans l'espace supérieur. Les figures 6.7 et 6.8 détaillent les trajectoires pour les flux 2 et 14, et montrent comment la trajectoire 14 évite d'interférer avec le segment en montée de la trajectoire 2 en étant légèrement déviée horizontalement. La route dans le plan est choisie parmi les points calculés que l'on voit sur la vue 2D (figure 6.7).

Les résultats obtenus semblent valables, en terme de résolution d'interférences entre trajectoires. A la fin du processus, on ne détecte aucune violation des contraintes d'occupation de l'espace pour les trajectoires associées à chacun des avions représentatifs. Rappelons cependant que ces trajectoires sont calculées sur la base d'un modèle simplifié des flux de trafic (un seul type d'avions, profil nominal, etc...), et seulement pour les flux dans l'espace aérien supérieur. Nous sommes donc encore loin d'un contexte réaliste, où les avions appartenant à un même flux origine-destination peuvent avoir des performances très disparates.

Le fait d'utiliser les niveaux de vol préalablement obtenus par l'algorithme d'allocation facilite grandement la recherche de trajectoires sans interférences. Les quelques tests effectués en utilisant un autre niveau préférentiel pour chaque avion représentatif (en l'occurrence le niveau de croisière le

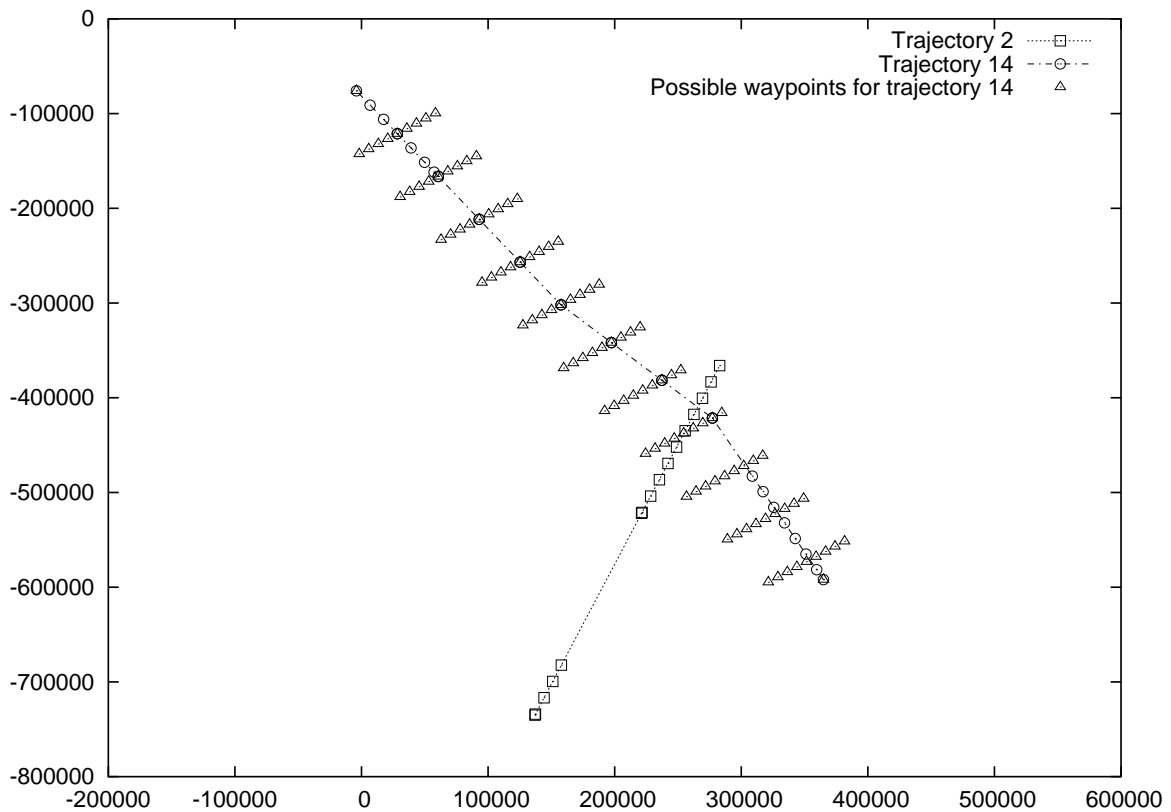


FIG. 6.7 – Vue en plan de deux trajectoires séparées (2 et 14 ; distances horizontales en mètres).

plus demandé dans le flux) ont montré des temps de calcul plus importants pour les flux de plus de 20 avions, voire rédhibitoires dès que l'on considère un nombre plus important de flux. L'explication est simple : au fur et à mesure que l'on ajoute des trajectoires, l'espace disponible diminue, et à chaque nouvelle trajectoire l'algorithme A^* doit explorer un nombre sans cesse croissant d'états pour éviter les précédentes, ou pour déterminer qu'il n'y a pas de solution. Avec la modélisation choisie, il y a plusieurs choix possibles de déviations de route et de changements de niveau de vol, à chaque pas de calcul de la trajectoire. Le nombre de combinaisons possibles pour ces choix est donc particulièrement important. Le fait de disposer d'une bonne heuristique permet certes d'orienter et d'accélérer le parcours dans l'arbre des choix possibles, mais la taille de cet arbre rend malgré tout difficile la recherche d'une trajectoire sans interférences, lorsque l'espace devient trop encombré.

6.4 Conclusion sur l'allocation des niveaux et la séparation des segments évolutifs

Nous avons démontré, dans la section 6.2, une amélioration significative du nombre et de la nature des conflits détectés lorsque l'on utilise l'allocation de niveaux de croisière. L'algorithme fournit des niveaux de croisière séparés, mais ne tient pas compte des phases de montée ou de descente. Cette méthode à elle seule réduit de 22,4 % le nombre de conflits détectés par le simulateur CATS/OPAS, on considérant les 63 % du trafic total, appartenant aux flux auxquels un niveau a été alloué. Parmi les conflits restants, 391 d'entre eux (soit plus de 37 %) sont des conflits *en rattrapage* entre avions

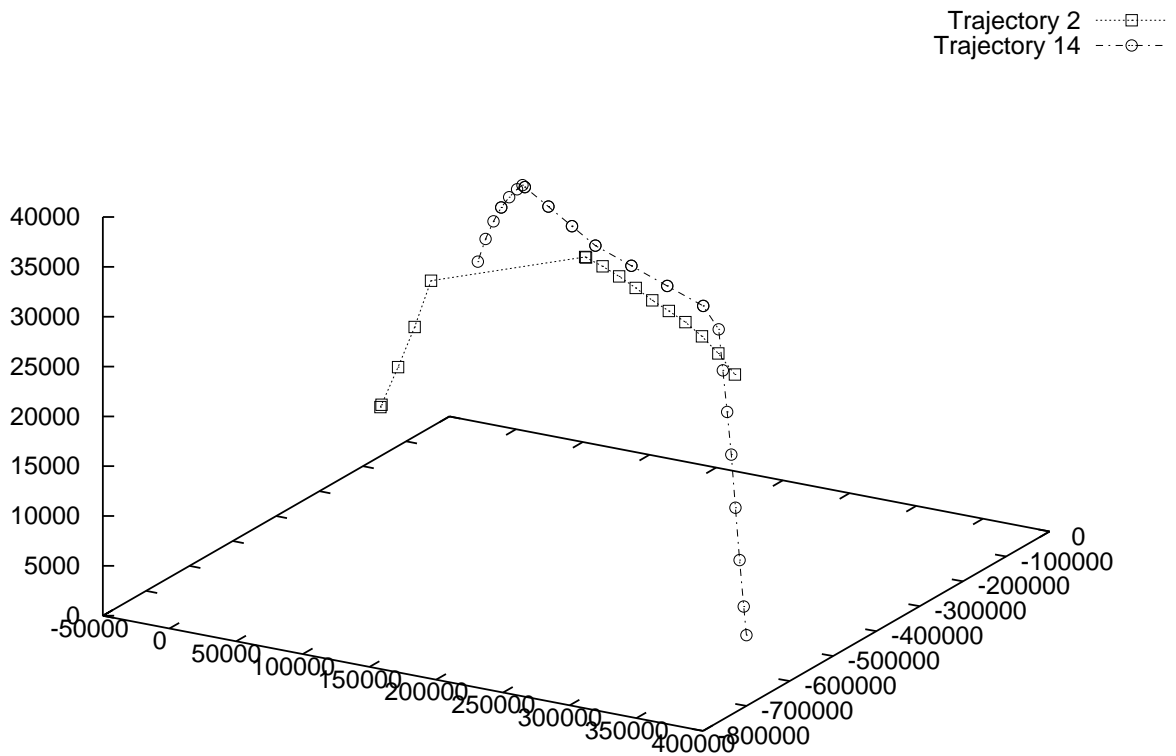


FIG. 6.8 – Vue 3D de deux trajectoires séparées (2 et 14 ; distances horizontales en mètres ; hauteurs en pieds).

appartenant au même flux et suivant donc la même trajectoire. On peut raisonnablement imaginer que les conflits de ce type soient résolus dans la phase tactique du contrôle aérien, par des techniques de séquençage et de régulation en vitesse (dites *miles-in-trail*), assurant une séparation longitudinale des avions sur la même trajectoire.

L'allocation de niveaux permet seulement d'éviter des conflits entre avions appartenant à des flux différents, et uniquement sur les segments en croisière. Les segments évolutifs ne sont pas du tout pris en compte.

Dans la section 6.3, les niveaux de croisière alloués précédemment sont utilisés dans le modèle de l'*avion représentatif*, où chaque flux est représenté par un seul avion. Un algorithme A^* est utilisé pour séparer les trajectoires dans les trois dimensions, et dans toutes les phases de vol (montée, croisière, descente). La détection d'interférences se fait en comptabilisant les trajectoires traversant les cubes élémentaires d'espace au voisinage de ceux traversés par la trajectoire en cours de construction. La taille des cubes est choisie en fonction des normes de séparation. Seuls les flux en espace aérien supérieurs sont traités. Les flux en espace inférieur n'ont pas encore été pris en compte. Le cas de deux flux ayant la même origine et la même destination, l'un en espace supérieur et l'autre en espace inférieur, devrait alors être examiné avec attention : ils devraient alors soit suivre la même route et n'être séparés que dans la phase de croisière, soit suivre des routes différentes et avoir alors des

trajectoires complètement séparées. Cet aspect (plusieurs trajectoires par flux) sera traité au chapitre 8.

Quoi qu'il en soit, la séparation complète des trajectoires 3D pour les flux en espace aérien supérieur semble correcte. L' A^* trouve des solutions dans des temps de calcul acceptables (quelques minutes sur un Pentium IV 1.8 GHz) dans la mesure où l'on utilise les niveaux de croisière préalablement fournis par l'algorithme d'allocation. Il reste à adapter le simulateur CATS/OPAS afin de réaliser les simulations permettant de quantifier les bénéfices apportés par cette méthode. Mais avant cela, nous allons reconsidérer dans le chapitre suivant la modélisation des trajectoires et des interférences, et chercher à appliquer l'algorithme A^* seul, sans allocation préalable de niveaux, ainsi qu'un algorithme génétique, sur un cas-test simplifié.

Après avoir exploré dans les chapitres précédents un certain nombre de problèmes concrets d'optimisation des flux de trafic aérien, nous allons donc, dans les chapitres suivants, nous intéresser uniquement à la séparation en 3D ou 4D des flux de trafic, de manière plus détaillée et selon d'autres angles d'approche que ceux évoqués dans ce chapitre.

Chapitre 7

Étude d'un problème simplifié de séparation de flux

7.1 Introduction

Considérons un problème purement hypothétique, consistant à trouver des trajectoires non conflictuelles dans l'espace, pour un nombre n de flux que l'on définit arbitrairement.

A titre d'exemple, on prendra des origines et destinations sur le bord d'un cercle, et des flux se croisant en un même point (le centre du cercle).

En utilisant une modélisation très simplifiée des profils de vol et des conflits de trajectoires, nous allons tester quelques techniques algorithmiques sur ce problème, après avoir estimé sa complexité.

7.2 Modélisation des trajectoires dans l'espace

7.2.1 Trajectoire préférentielle associée à un flux

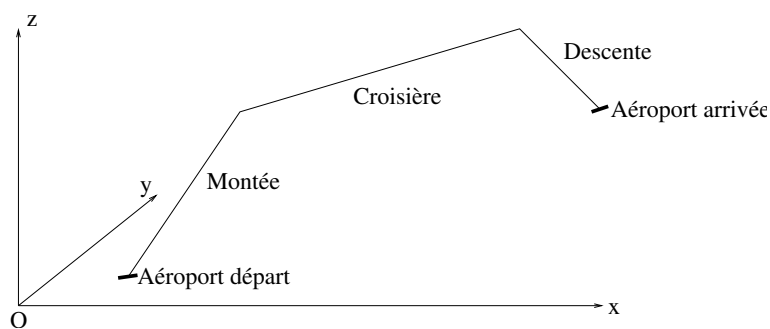


FIG. 7.1 – Trajectoire simplifiée associée à un flux.

Nous allons pour commencer faire quelques hypothèses simplificatrices nous permettant d'associer une seule trajectoire à un flux (*origine, destination*) donné :

- On se place dans un référentiel euclidien, avec des coordonnées en x, y, z . L'utilisation d'un référentiel adapté à la géométrie de la terre n'est pas envisagée à ce stade. Notons que l'on pourra toujours se ramener à un référentiel euclidien par une projection stéréographique, moyennant

des marges d'erreur supplémentaires dans les normes de séparation entre trajectoires, et à condition de se limiter à une région terrestre pas trop importante (à l'échelle de la France ou de l'Europe).

- Tous les aéroports de départ et d'arrivée sont à l'altitude 0 dans ce référentiel.
- Tous les avions ont des performances identiques et suivent en montée ou en descente des pentes linéaires.
- Pour un flux donné i , tous les avions demandent un même niveau RFL_i ¹, et souhaitent suivre préférentiellement une route directe entre départ et destination.

Une trajectoire pourra donc être représentée par une suite de segments de droites dans l'espace euclidien à trois dimensions. La figure 7.1 illustre une trajectoire préférentielle, la plus directe possible, avec une montée vers un palier au niveau demandé RFL , puis une descente vers l'aéroport de destination.

7.2.2 Déviations latérales et verticales

Afin d'éviter les interférences entre trajectoires, il nous faudra introduire certaines déviations, latérales ou verticales, par rapport aux trajectoires préférentielles. Nous allons maintenant détailler la modélisation choisie pour décrire une trajectoire, déviée ou non, en faisant abstraction de l'indice i du flux, afin d'alléger les notations.

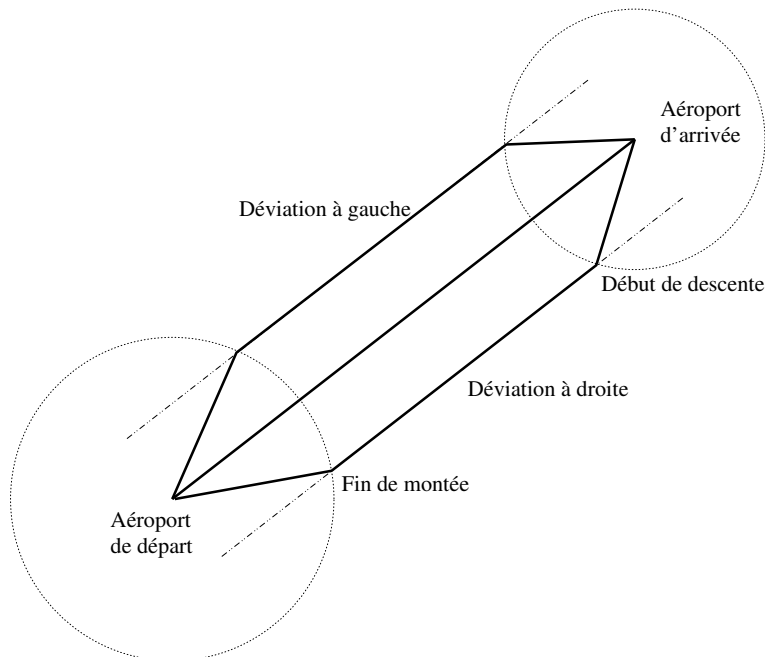


FIG. 7.2 – Déviations horizontales possibles.

Horizontalement, on autorisera pour l'instant trois possibilités seulement, illustrées sur la figure 7.2 : la ligne droite, une parallèle à gauche, ou une parallèle à droite. Les points tournants des déviations droite ou gauche sont les points de début et de fin de croisière selon un profil idéal (croisière à RFL sur la route considérée), mais ce choix est paramétrable et l'on peut en fait choisir à la fois la valeur de l'écart latéral et celles des rayons autour des aéroports de départ et de destination.

¹RFL (*Requested Flight Level*) : niveau de vol demandé pour la croisière

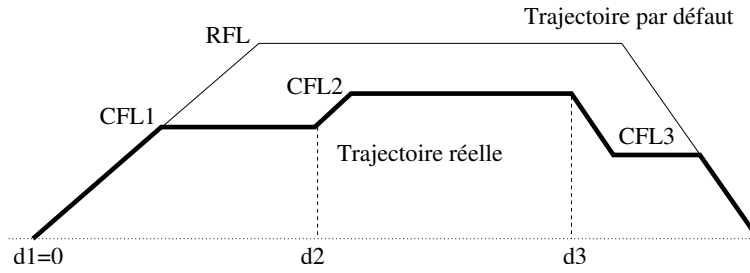


FIG. 7.3 – Cas typique de profil vertical autorisé.

Verticalement, on s'autorisera une succession de paliers à des niveaux de vol différents, comme illustré sur la figure 7.3. Chaque niveau de vol devra prendre sa valeur entre un niveau minimum FL_{min} et le niveau de vol demandé RFL . Les déviations verticales sont donc caractérisées par une séquence de couples (d_j, CFL_j) , où d_j est la distance (le long de la route) à laquelle l'évolution verticale vers le niveau de vol CFL_j commence².

7.2.3 Modélisation d'une trajectoire

Pour récapituler, sachant que l'on se fixe les pentes de montée et de descente, une trajectoire \mathcal{T}_i associée à un flux i est totalement déterminée par les éléments suivants, en faisant abstraction de l'index i pour des besoins de clarté :

- le choix r de la route, qui vaudra 0 quand la route directe sera choisie, 1 pour une déviation sur la droite, et -1 pour la route déviée à gauche.
- la séquence $\{(d_j, CFL_j) / j \in [0, p - 1]\}$, où chaque couple (d_j, CFL_j) se traduit par : à la distance d_j , commencer une évolution verticale vers le niveau CFL_j . On aura au maximum p couples, car les valeurs de d_j seront choisies dans un ensemble fini (cf ci-dessous).

Les limitations imposées aux évolutions verticales sont les suivantes, sachant que RFL est le niveau de croisière demandé et l la longueur de la route suivie :

$$\forall j \quad FL_{min} \leq CFL_j \leq RFL \quad (7.1)$$

$$\exists k \in [0, p - 1] \text{ tel que } d_j = k \times \frac{l}{p} \quad (7.2)$$

7.2.4 Coût associé à une trajectoire

Pour chaque trajectoire i , le coût sera directement lié aux écarts verticaux et horizontaux par rapport à la trajectoire préférentielle.

Le coût d'une déviation latérale dépend de l'élongation $\frac{(l_i - l_{ref_i})}{l_{ref_i}}$ de la route suivie par rapport à la route directe (de longueur l_{ref_i}).

Le coût d'une déviation verticale dépend de la surface entre le profil de vol effectivement suivi et le profil par défaut (au niveau RFL_i) : $l_i \times RFL_i - \text{surface}(\text{profil}_i)$, où l_i est la longueur de la route suivie. On pourrait s'en tenir à cette définition, le coût des déviations verticales dépendant alors linéairement de la distance entre l'origine et la destination. Par souci d'équité entre les vols, et afin de pouvoir équilibrer correctement les coûts des déviations latérales et verticales dans le coût total, on choisit de diviser cette expression par la longueur de la route l_i .

²CFL (Cleared Flight Level) : niveau autorisé par le contrôle aérien.

Finalement, le coût total associé à une trajectoire est une combinaison des deux :

$$cost(i) = RFL_i - \frac{surface(profil_i)}{l_i} + K \times \frac{(l_i - lref_i)}{lref_i} \quad (7.3)$$

où RFL_i est le niveau de vol initialement demandé pour la croisière, et K est un coefficient choisi de façon à équilibrer au mieux le coût d'une déviation horizontale par rapport à une déviation verticale.

Notons que si l'on ne divisait pas la surface de l'écart vertical par la longueur de la route, le poids relatif des déviations verticales serait beaucoup plus important pour les vols long-courriers que pour les court-courriers. Minimiser le coût total conduirait alors à privilégier les déviations latérales pour les long-courriers et les déviations horizontales pour les court-courriers. En soi ce ne serait pas forcément une mauvaise chose, sauf que cela rend problématique le choix d'une valeur pour K . On peut de plus perdre du temps à explorer les déviations d'un certain type, alors que la solution naturelle passe par une déviation de l'autre type.

7.3 Modélisation des interférences entre trajectoires

Nous éviterons délibérément d'utiliser le terme de *conflit* entre trajectoires qui se réfère usuellement au fait que les positions de deux avions dans l'espace sont (ou seront dans un proche futur) plus proches que la norme de séparation autorisée. Soient N_h la norme de séparation dans le plan horizontal et N_v la norme de séparation verticale. Deux trajectoires \mathcal{T}_1 et \mathcal{T}_2 interfèrent s'il existe au moins un couple de points (p_1, p_2) tels que $p_1 \in \mathcal{T}_1, p_2 \in \mathcal{T}_2$ et que l'on ait simultanément $d_h(p_1, p_2) \leq N_h$ et $d_v(p_1, p_2) \leq N_v$. Cette définition permet d'assurer que deux avions empruntant des trajectoires qui n'interfèrent pas n'ont aucune chance de se retrouver en conflit (s'ils suivent parfaitement leur trajectoire, bien sûr).

Il est facile de trouver un algorithme détectant les interférences entre trajectoires stables en niveau : soit les deux trajectoires considérées sont verticalement séparées de plus de N_v , soit elles ne le sont pas et on calcule alors facilement la distance horizontale entre les deux points de rapprochement maximal, pour la comparer à N_h . C'est plus compliqué dès que l'on a des segments en montée ou en descente : les deux trajectoires pourraient parfaitement être séparées verticalement aux points de rapprochement maximal (dans le plan horizontal), mais ne pas l'être en d'autres endroits. On ne peut donc plus considérer indépendamment la dimension verticale et le plan horizontal et il faut déterminer les portions de trajectoires qui violent simultanément les deux normes, ce qui, géométriquement, n'est pas simple.

C'est pourquoi nous allons légèrement modifier la définition de l'interférence entre trajectoires, en introduisant une nouvelle distance. Soit le produit scalaire défini par :

$$\begin{pmatrix} x1 \\ y1 \\ z1 \end{pmatrix} \cdot \begin{pmatrix} x2 \\ y2 \\ z2 \end{pmatrix} = \frac{x1 \cdot x2 + y1 \cdot y2}{N_h^2} + \frac{z1 \cdot z2}{N_v^2}$$

Soit d la distance associée, définie comme la racine carrée du produit scalaire du vecteur reliant les deux points par lui-même. L'algorithme calculant cette distance ne présente aucune difficulté à implanter, et on montre que pour tout couple de trajectoires $(\mathcal{T}_1, \mathcal{T}_2)$, si p_1 et p_2 sont les points de rapprochement maximal de ces deux trajectoires, avec la distance d , alors :

$$d(p_1, p_2) > \sqrt{2} \quad \Rightarrow \quad \forall (q_1, q_2) \in \mathcal{T}_1 \times \mathcal{T}_2 \quad d_h(q_1, q_2) > N_h \quad \text{et} \quad d_v(q_1, q_2) > N_v \quad (7.4)$$

Considérons l'équation $d(p_1, p_2) = \sqrt{2}$. Elle se ramène à l'équation suivante, x, y, z étant les coordonnées du vecteur reliant p_1 et p_2 :

$$\frac{x^2 + y^2}{N_h^2} + \frac{z^2}{N_v^2} = 2$$

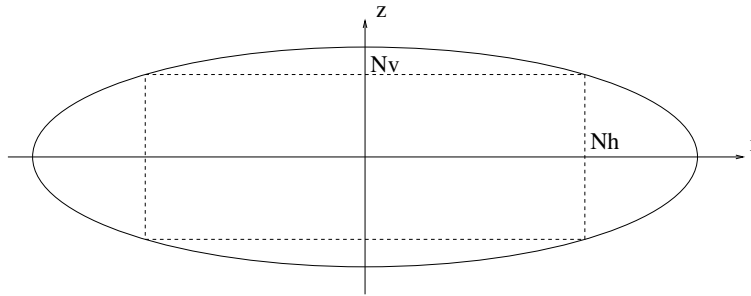


FIG. 7.4 – Vue en coupe de l'ellipsoïde.

Il s'agit d'une surface ellipsoïde en forme de sphère aplatie, dont une vue en coupe est représentée en figure 7.4. On constate, sur cette figure, que tout point situé à l'extérieur de la surface est nécessairement situé à une distance horizontale supérieure à N_h du centre, et à une distance verticale supérieure à N_v , ce qui nous convainc de la véracité de la proposition 7.4. La réciproque de cette proposition n'est pas vraie : il suffit de considérer tous les points à l'intérieur de l'ellipsoïde et à l'extérieur du cylindre de rayon N_h et de demi-hauteur N_v .

Dans la suite de ce chapitre, nous allons considérer que deux segments de trajectoires $s_1 \subset \mathcal{T}_1$ et $s_2 \subset \mathcal{T}_2$ dont l'un au moins est en montée ou en descente, interfèrent lorsque les points de rapprochement maximal (avec la distance d) sont à une distance inférieure à $\sqrt{2}$, et sont séparés dans le cas contraire. Pour deux segments à altitude constante, nous prendrons la définition classique : il interféreront si la différence des altitudes est inférieure à N_v et si la distance horizontale (euclidienne cette fois) les séparant est inférieure à N_h . L'utilisation de la distance d revient à s'assurer une marge supplémentaire par rapport à la définition initiale donnée au début de cette section. Les flux d'avions volant sur des trajectoires séparées selon le critère de distance choisi n'ont donc aucune chance de provoquer des conflits, du moins entre avions appartenant à des flux différents.

7.4 Description du problème, stratégies possibles

Considérons N flux d'avions, à chaque flux étant associée une trajectoire préférentielle entre son aéroport de départ et sa destination. Ces trajectoires sont susceptibles d'interférer entre elles, au sens défini dans la section précédente. L'objectif que l'on se fixe est de trouver un ensemble de N nouvelles trajectoires séparées selon le critère de distance exposé précédemment, et les plus proches possibles des trajectoires préférentielles. Il s'agit d'un problème de minimisation, pour lequel on peut choisir parmi deux types de stratégies.

Une première stratégie, que nous baptiserons *1 contre n*, consiste à parcourir les flux en séquence avec l'objectif de trouver pour chacun une trajectoire optimale, minimisant l'écart à la trajectoire préférentielle tout en évitant les n trajectoires (avec $n < N$) déjà calculées pour les flux précédents. On voit que l'ordre dans lequel sont traités les flux n'est pas neutre. Un choix sensé pour notre problème consistera à parcourir ceux-ci par ordre décroissant d'importance. De cette façon, les flux les plus importants bénéficieront des trajectoires les plus directes.

Dans une deuxième approche, que l'on qualifiera dans la suite de *stratégie globale*, on cherchera à minimiser un coût associé à l'ensemble des trajectoires, plutôt que de calculer celles-ci séquentiellement en minimisant un coût individuel. Ce coût global, qui sera détaillé dans la section 7.7, est bien sûr dépendant des déviations par rapport aux trajectoires préférentielles.

C'est donc en fait à deux problèmes de minimisation bien distincts que l'on a affaire, selon que l'on choisit une stratégie globale ou *1 contre n*.

7.5 Difficulté des problèmes, choix des algorithmes

La difficulté du problème de minimisation en séquence selon la stratégie *1 contre n* est liée d'une part au nombre de contraintes (zones d'espace occupées par les flux précédents) imposées à la trajectoire que l'on cherche à construire, et d'autre part à la grande variété de choix possibles dans la construction de la trajectoire. Ces difficultés ne semblent toutefois pas rédhibitoires, et permettent d'envisager l'utilisation de méthodes déterministes de parcours d'arbres. En l'occurrence, s'agissant de la détermination du plus court chemin dans un espace saturé, l'algorithme A^* paraît tout à fait indiqué. La section 7.6 détaille l'adaptation de cet algorithme à notre problème.

La difficulté du problème global est quant à elle analogue à celle de la résolution de conflits aériens, discutée dans [Dur96] (chapitre 2.3). Considérons deux flux d'avions suivant des trajectoires préférentielles T_1 et T_2 qui interfèrent. Pour obtenir deux autres trajectoires séparées T'_1 et T'_2 les plus proches possibles de T_1 et T_2 , on a deux options possibles : soit le tube de rayon $\sqrt{2}$ défini (avec la distance d) autour de T'_1 traverse la surface définie par la trajectoire et la ligne entre départ et arrivée de T'_2 en étant complètement à l'intérieur (comme illustré en figure 7.5), soit ce tube est complètement à l'extérieur de cette surface.

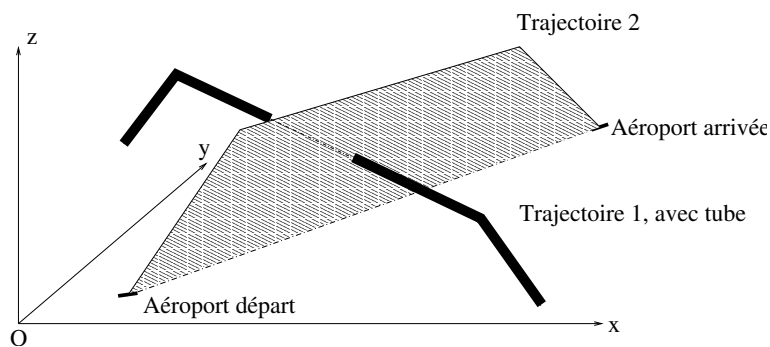


FIG. 7.5 – Exemple de solution de croisement de deux trajectoires.

Sauf à faire voler les avions sous terre, il est impossible de passer continûment d'une solution du type *à l'intérieur* à une solution *à l'extérieur*, sans tomber dans un domaine où les deux trajectoires interfèrent. L'ensemble des solutions admissibles pour deux trajectoires comprend donc deux composantes connexes.

Pour un nombre n quelconque de trajectoires interférentes, il faut considérer les $\frac{n(n-1)}{2}$ couples possibles de trajectoires, et le nombre de composantes connexes s'élève alors à $2^{\frac{n(n-1)}{2}}$. L'existence d'un tel nombre de composantes connexes (plus de trente mille milliards pour 10 trajectoires) et le fait que l'on ne sait pas *a priori* dans laquelle se trouve la solution optimale, rendent le problème fortement combinatoire.

Cette constatation conduit à penser que l'utilisation de méthodes d'optimisation locales et déterministes (gradient, BFGS, Branch & Bound, etc...) n'est pas adaptée. De telles méthodes, dont le principe général est de parcourir l'espace d'états à partir d'un point initial choisi et en cherchant à améliorer un critère de coût, ont de grandes chances de mener soit à l'échec, soit à une solution qui ne sera qu'un optimum local appartenant à la même composante connexe que le point initial choisi.

On s'orientera donc pour le problème global vers l'utilisation de méthodes stochastiques. La section 7.7 présente la résolution du problème global avec des profils de vol simplifiés par un algorithme génétique.

7.6 Résolution du problème 1 contre n par un algorithme A*

7.6.1 Mise en oeuvre de l'algorithme A*

L'algorithme A* et son principe de recherche dans un arbre d'états sont présentés au chapitre 3, section 3.5.2. Dans notre problème, les états représenteront des choix lors des étapes successives de la construction de la trajectoire. Les différents choix possibles sont illustrés dans la section suivante. La fonction de coût et l'heuristique seront choisies de façon à tenir compte de l'écart par rapport à la trajectoire préférentielle.

L'algorithme sera appliqué successivement à chacune des trajectoires, de façon à éviter les trajectoires précédemment calculées.

7.6.2 Description du parcours de l'arbre

Pour chacune des trajectoires que l'on cherche, l'algorithme A* parcourt un arbre dont les noeuds sont des états représentant une étape (un choix de route ou de niveau de vol) de la construction de la trajectoire. Cette dernière est définie par le choix r de la route, et par les contraintes sur les niveaux de vol :

$$cfls = \{(d_j, CFL_j) / j \in [0, p - 1]\}$$

A titre d'exemple, pour la trajectoire par défaut, c'est-à-dire la plus directe avec une croisière au RFL, nous aurons $r = 0$ et $cfls = [(0, RFL)]$.

La trajectoire est une suite d'étapes : montée (ou descente) vers le niveau CFL_j , suivie d'une stabilisation à ce niveau jusqu'à la prochaine évolution vers le niveau CFL_{j+1} . Donc chaque contrainte (d_j, CFL_j) se traduira par un segment de vol évolutif, suivi d'un segment de vol stabilisé. Une étape de la trajectoire sera alors caractérisée par l'indice j de la contrainte en niveau et par le choix seg du segment : *Évolutif* ou *Croisière* et un noeud de l'arbre sera donc un état :

$$(r, cfls, j, seg)$$

La racine de l'arbre, à partir de laquelle commence la recherche, est l'état initial spécifique, noté *Départ*, pour lequel aucun choix de route ou de trajectoire n'a encore été effectué. Pour chaque noeud parcouru, les noeuds fils sont calculés selon les principes suivants, en fonction de la nature de l'étape courante :

Racine de l'arbre (départ) : les trois noeuds fils correspondent chacun à l'une des trois routes possibles (on choisit la valeur de r), avec une phase de montée initiale sans conflit. On choisira donc pour chacun d'entre eux la première valeur pour laquelle il n'y a pas d'interférence entre la montée initiale et les trajectoires précédemment définies. Les valeurs possibles sont testées

par décrets de 1000 pieds à partir de RFL , et doivent rester supérieures à un minimum FLM_{min} . Si aucun fils valide n'est trouvé, la recherche est terminée : il n'y a pas de solution.

Étape évolutive $(r, \{(d_0, CFL_0); \dots; (d_j, CFL_j)\}, j, Evolutif)$:

- un premier fils correspondant au passage à la croisière :

$$(r, \{(d_0, CFL_0); \dots; (d_j, CFL_j); (d_{j+1}, CFL_j)\}, j, Croisiere)$$

La contrainte supplémentaire (d_{j+1}, CFL_j) est ajoutée, en choisissant d_j de façon à avoir un segment de vol en croisière le plus long possible sans interférences avec les trajectoires précédentes. Au mieux, d_{j+1} correspondra au début de la descente finale vers l'aéroport de destination. Si un tel palier n'existe pas, ce fils n'est pas retenu.

- un deuxième fils représentant une évolution verticale alternative vers un niveau inférieur :

$$(r, \{(d_0, CFL_0); \dots; (d_j, CFL_j - 1000 \text{ pieds})\}, j, Evolutif)$$

Ce fils n'est pas retenu lorsque le nouveau niveau de stabilisation $CFL_i - 1000$ pieds est inférieur à FL_{min} :

Étape de croisière $(r, \{(d_0, CFL_0); \dots; (d_j, CFL_j); (d_{j+1}, CFL_j)\}, j, Croisiere)$:

- un premier fils correspondant au passage à l'étape évolutive suivante :

$$(r, \{(d_0, CFL_0); \dots; (d_j, CFL_j); (d_{j+1}, CFL_{j+1})\}, j + 1, Evolutif)$$

Le niveau cible CFL_{j+1} est le premier niveau de vol inférieur ou égal à RFL , et différent de CFL_j , pour lequel le segment évolutif entre CFL_j et CFL_{j+1} est libre de toute interférence.

- un deuxième fils correspondant à une étape j avec un palier de croisière de longueur inférieure :

$$(r, \{(d_0, CFL_0); \dots; (d_j, CFL_j); (d_{j+1} - \delta, CFL_j)\}, j, Croisiere)$$

où δ est le pas qu'on s'est fixé pour le choix des distances d_j .

- si d_{j+1} correspond au début de la descente finale, on teste si la descente finale est libre de toute interférence, auquel cas la trajectoire $(r, \{(d_0, CFL_0); \dots; (d_j, CFL_j)\})$ est la solution. La recherche est alors terminée, on a atteint une feuille de l'arbre.

L'arbre est parcouru en partant, à chaque pas de calcul, du meilleur noeud calculé précédemment. Pour ceci, l' A^* gère une file d'attente à priorités (*priority queue*), qui mémorise tous les noeuds parcourus. Le processus consiste à prendre le meilleur noeud (en tête de la file), à calculer les noeuds fils et à les insérer à leur tour dans la file à priorités. L'insertion se fait avec une priorité liée à la somme du coût des transitions passées et de l'heuristique. Le processus est répété jusqu'à ce que l'on tombe sur une feuille (la solution trouvée est alors optimale) où jusqu'à ce que l'on démontre qu'aucune feuille ne peut être atteinte (il n'y a pas de solution).

7.6.3 Coût de transition entre états

A chaque pas de l'algorithme, il faut calculer le coût des transitions entre le noeud père extrait de la file d'attente et chacun des noeuds fils. Pour notre problème, le coût d'une transition entre un noeud père s_1 et un noeud fils s_2 est défini en fonction de l'étape du noeud père et de celle du noeud fils de la façon suivante :

- $K \times \frac{l-lref}{lref}$, l'allongement relatif de la route, quand s_1 est l'état initial (la racine de l'arbre), l étant la longueur de la route suivie et $lref$ celle de la route directe,

- $\frac{surf_j}{l}$, si s_1 est le segment *Croisière* d'une étape j , et s_2 le segment *Evolutif* de l'étape $j + 1$, où $surf_j$ désigne la surface comprise entre les distances d_j et d_{j+1} délimitée par le niveau *RFL* et la portion de trajectoire associée à la contrainte j . La figure 7.6 donne un exemple d'une telle surface,
- 0 dans tout autre cas.

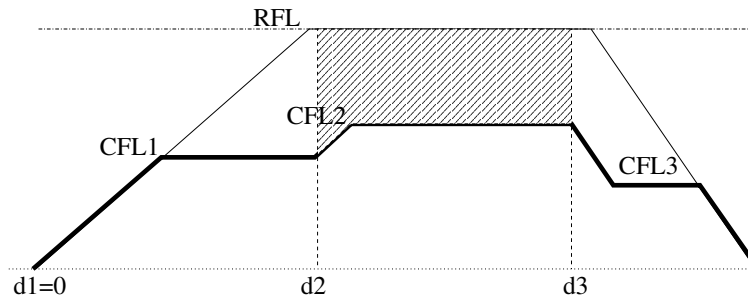


FIG. 7.6 – Exemple de surface permettant le calcul du coût de transition.

Nous constatons que lorsqu'on atteint la solution, on retrouve bien le coût total associé à une trajectoire, décrit en section 7.2.4.

7.6.4 Choix de l'heuristique

Après avoir calculé les noeuds fils et le coût des transitions entre le noeud père et chacun d'entre eux, il nous faut une estimation minorante du coût du parcours restant entre chaque noeud fils et les solutions atteignables à partir de celui-ci. En ajoutant cette estimation, appelée heuristique, au coût des transitions passées, on obtient la valeur que l'on associera au noeud fils au moment de l'insérer dans la file à priorités. Cette valeur représente en fait le coût estimé des solutions que l'on peut atteindre en ayant suivi d'abord le parcours menant au noeud fils. Au pas de calcul suivant, on prendra dans la file d'attente le noeud dont le coût estimé pour atteindre la solution est le plus faible. On voit que l'heuristique doit absolument minorer le coût du parcours restant, faute de quoi le coût estimé d'un noeud menant à la solution optimale pourrait être surévalué, empêchant d'explorer plus avant le sous-arbre correspondant.

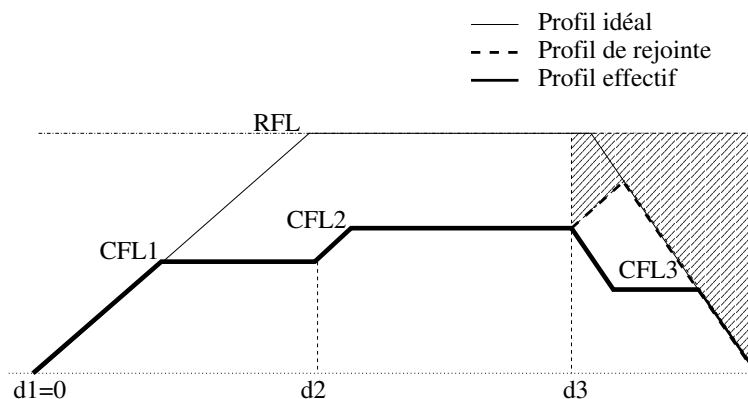


FIG. 7.7 – Exemple de surface permettant le calcul du coût de l'heuristique.

Pour notre problème, nous allons calculer un profil vertical partant du point courant (c'est-à-dire

le dernier point de la portion de trajectoire correspondant à l'étape courante) et rejoignant le profil vertical idéal pour la route choisie (avec une seule phase de croisière au niveau *RFL*). L'heuristique sera alors tout simplement la surface délimitée par ce profil de rejointe et par le *RFL* (cf figure 7.7), divisée par la longueur de la route. Cette heuristique est bien une borne inférieure du coût de toutes les transitions possibles permettant d'atteindre une feuille de l'arbre à partir de l'état courant.

7.6.5 Résultats de l' A^*

L'algorithme a été testé avec différentes configurations comprenant dix flux :

- sur des trajectoires convergentes en demi-cercle de rayon 350 milles nautiques,
- sur des trajectoires en cercle, avec un flux aller et un flux retour sur chaque diamètre.
- sur un quadrillage 5x5 de 700 milles nautiques de large,

Demi-cercle

Pour cette configuration de 10 flux avec des points de départ sur un demi-cercle, convergeant vers un même point et demandant tous le même niveau de vol (35000 pieds, ou FL350), l'algorithme A^* trouve une solution illustrée sur les figures 7.8, et 7.9, avec uniquement des déviations verticales.

Le coût cumulé de ces déviations est de 52.068, et le temps de calcul sur un CPU Xeon 2.8 GHz est de 5.76 secondes. Le coût est calculé selon la formule 7.3 (section 7.2.4), avec un coefficient K fixé à 10000. Comme il cumule les coûts des déviations latérales et verticales, il n'est pas aisé de lui donner une interprétation plus parlante que sa définition. Il s'agit donc essentiellement d'un indicateur permettant de comparer des solutions entre elles, ce que nous ferons un peu plus loin avec les solutions de l'algorithme génétique.

Cercle

Ici les points de départ des 10 flux sont répartis symétriquement sur un cercle. Comme précédemment, ils convergent tous vers un même point, et demandent tous le même niveau de vol (FL350). Une difficulté supplémentaire apparaît : chacun des flux rencontre un flux allant dans le sens inverse, ce qui pose un problème pour la détection des interférences.

En effet, pour deux flux entre deux mêmes points mais de directions opposées, les trajectoires 3D aboutissent nécessairement sur les mêmes points (les aéroports). Avec la détection d'interférences décrite en 7.3, les segments des trajectoires ayant pour extrémité un même aéroport seront nécessairement interférents, et l'algorithme sera donc dans l'impossibilité de trouver une solution pour la moitié des flux.

Afin de pallier cet inconvénient, une zone de non-détection est définie autour de chaque aéroport : la détection est inhibée pour les segments à l'intérieur d'un cercle d'un certain rayon. Celui-ci est choisi de façon que les trajectoires empruntant une route parallèle à la route directe soient séparées latéralement de cette dernière, à l'extérieur du cercle. En pratique, un rayon de 15 milles nautiques a été choisi pour l'exemple illustré ici.

Dans la réalité, les avions évoluant autour des aéroports sont gérés par le contrôle d'approche, qui assure leur séparation. La zone de non-détection peut être considérée comme cette zone terminale d'approche, et notre objectif est de trouver des trajectoires 3D séparées entre ces zones.

La solution trouvée par l'algorithme A^* est illustrée sur les figures 7.10, et 7.11. On voit les évitements du point de croisement au centre, avec des déviations verticales occupant cinq niveaux de

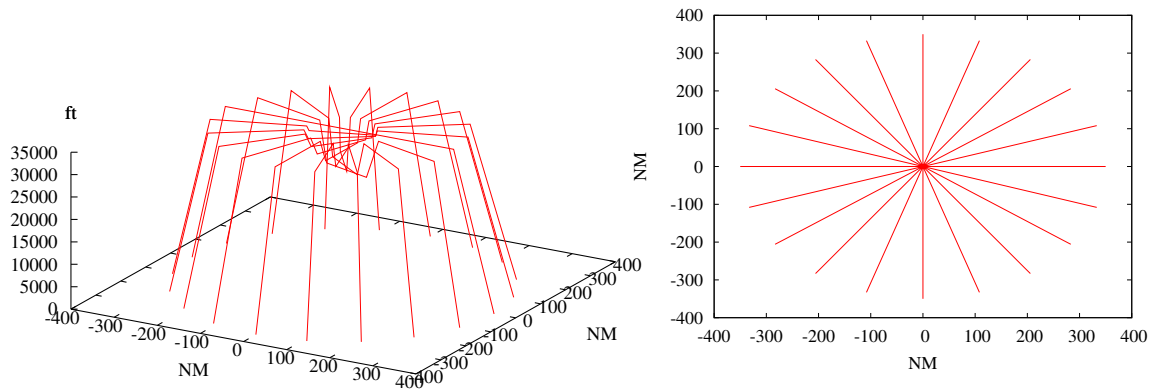


FIG. 7.8 – Résultats de l’A* sur une configuration en demi-cercle (distances horizontales en milles nautiques ; altitude en pieds).

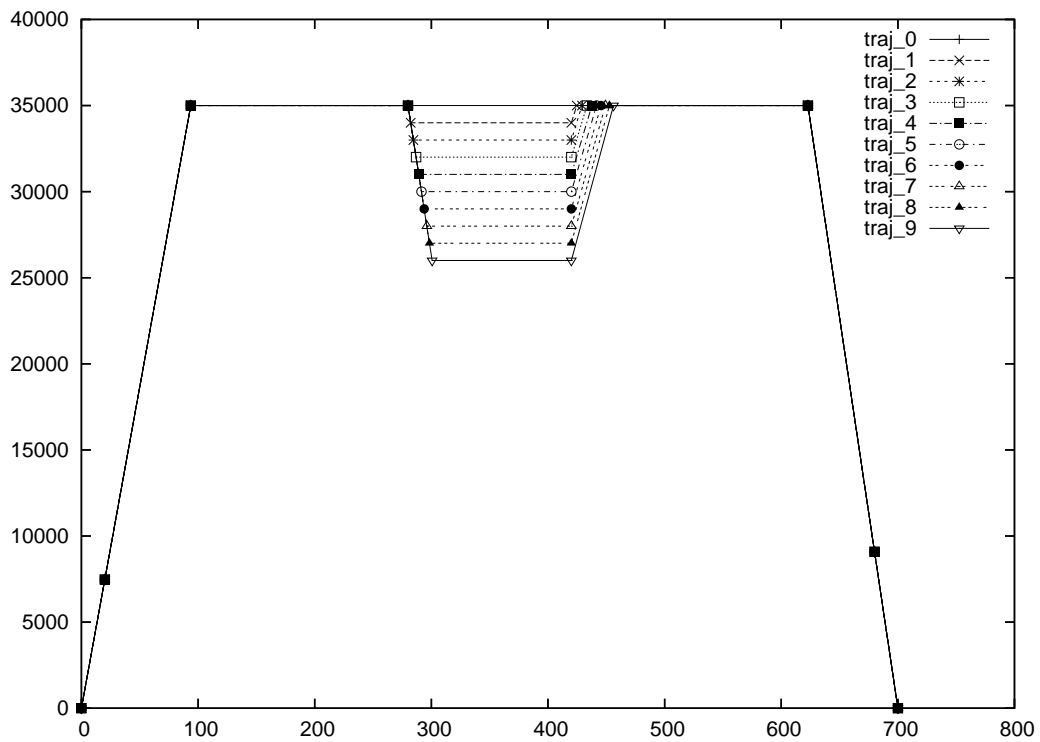


FIG. 7.9 – Vue des profils verticaux des résultats de l’A* sur une configuration en demi-cercle (en abscisse : distance parcourue, en milles nautiques ; en ordonnée : altitude en pieds).

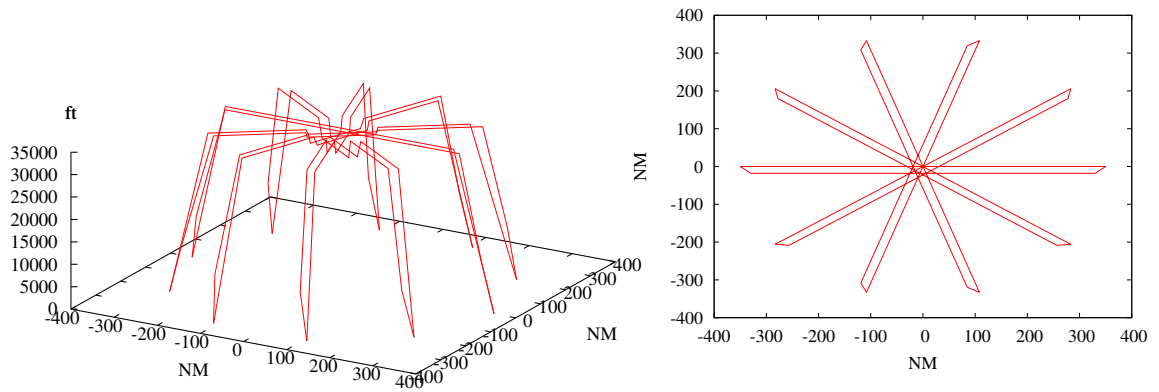


FIG. 7.10 – Résultats de l' A^* sur une configuration en cercle (distances horizontales en milles nautiques ; altitudes en pieds).

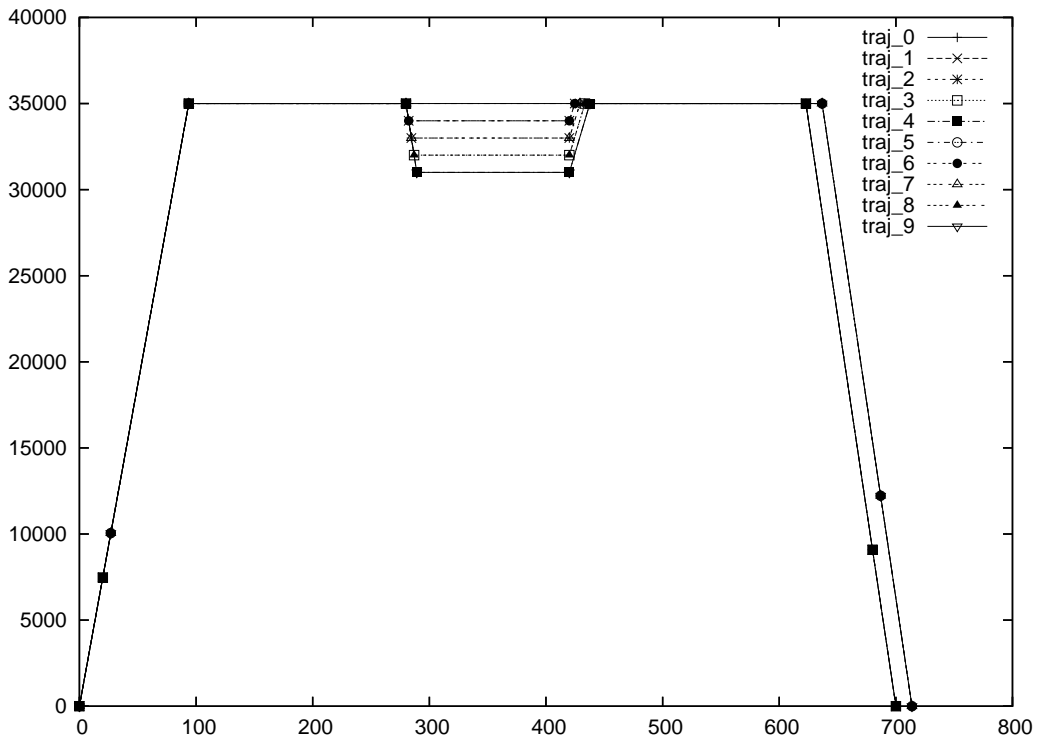


FIG. 7.11 – Vue des profils verticaux des résultats de l' A^* sur une configuration en cercle (en abscisse : distance parcourue, en milles nautiques ; en ordonnée : altitude en pieds).

vol, et aussi l'évitement des flux en direction opposée, avec des déviations latérales empruntant des routes parallèles.

Le coût cumulé des déviations de trajectoires est de 56.195, avec un temps de calcul (CPU Xeon 2.8 GHz) de 1.94 secondes.

Quadrillage

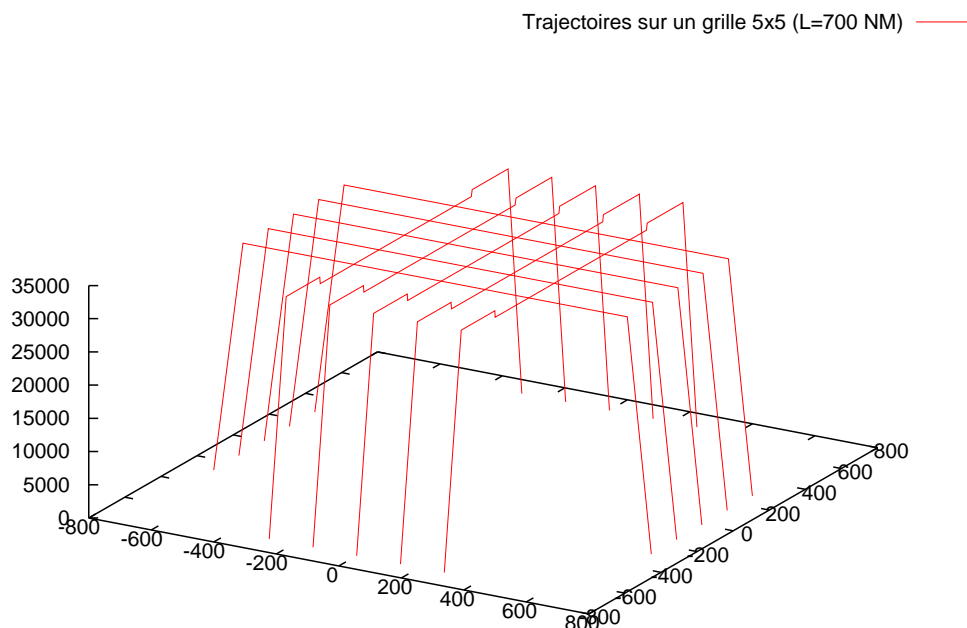


FIG. 7.12 – Résultat de l'A* pour 10 flux en quadrillage, demandant tous le niveau 350, considérés dans l'ordre des x croissants, puis des y croissants (distances horizontales en milles nautiques ; altitudes en pieds).

La figure 7.12 illustre l'affectation de trajectoire pour le problème de 10 flux sur un quadrillage 5X5. La séquence choisie pour traiter ces flux consiste à prendre d'abord cinq premiers flux allant dans une direction, puis les cinq autres flux allant dans l'autre direction. Les flux allant dans une même direction ne se croisent pas. En conséquence, les cinq premiers flux suivent leur trajectoire idéale : route directe et niveau de vol demandé (FL350), et les cinq autres croisent horizontalement les cinq premiers en passant en-dessous, occupant le niveau de vol immédiatement inférieur (FL340). Le coût cumulé des déviations de trajectoire pour cette solution est de 24.34851, et le temps de calcul de 1.46 secondes (CPU Xeon 2.8 GHz).

La figure 7.13 représente la solution de l'A* lorsque les flux sont considérés dans un ordre différent (alternance de flux dans une direction, puis dans l'autre). Cette solution est plus coûteuse qu'avec l'ordre précédent (27.96444), pour un temps de calcul de 4.6 secondes.

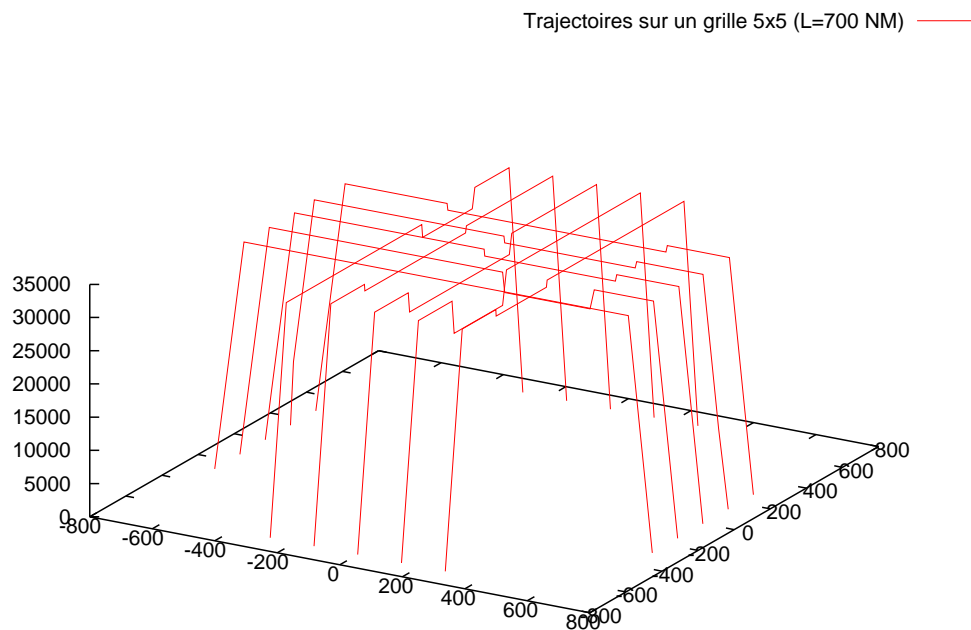


FIG. 7.13 – Résultat de l' A^* pour 10 flux en quadrillage, demandant tous le niveau 350, considérés dans un ordre alternatif (distances horizontales en milles nautiques ; altitudes en pieds).

7.7 Résolution du problème global par un algorithme génétique

7.7.1 Représentation du problème

Au vu de la difficulté du problème global, nous avons choisi d'utiliser une méthode d'optimisation stochastique, en l'occurrence un algorithme génétique.

Dans notre cas, un individu est un ensemble de n trajectoires, la mutation s'opère par modification aléatoire d'une ou plusieurs trajectoires de cet ensemble, et le croisement entre deux individus se fait par échange ou modification d'une partie des trajectoires. Le critère d'adaptation permettant la sélection est directement dépendant du coût global associé aux déviations par rapport aux trajectoires préférentielles, avec de plus une forte pénalisation lorsque les trajectoires ne sont pas séparées.

7.7.2 Choix de la population initiale

Chaque élément de la population initiale est créé en choisissant de manière aléatoire les éléments déterminants des n trajectoires qui composeront cet élément, à l'intérieur toutefois de certaines limites.

Les parallèles droite et gauche sont distantes de 15 milles nautiques de la route directe. Le niveau de croisière doit être entre le FL290 et le FL350. Les niveaux de stabilisation en montée ou en descente doivent se situer entre le FL200 et le niveau de croisière. Les segments de vol stabilisé lors d'une montée interrompue ou d'une descente anticipée ne doivent pas représenter chacun, ou à eux deux, plus de 80% de la longueur totale des segments en vol stable.

7.7.3 Le critère d'adaptation

Afin de construire le critère d'adaptation selon lequel seront sélectionnés les meilleurs individus d'une population, nous allons calculer une matrice triangulaire C dont la diagonale contiendra, pour chaque trajectoire i , le coût de l'écart par rapport à la trajectoire préférentielle. Les éléments non diagonaux, d'indices (i, j) avec $i \neq j$, permettront de comptabiliser les interférences entre segments de trajectoires (notés s). Voici l'expression des coefficients de cette matrice, avec les notations introduites précédemment :

$$\forall i \neq j \quad \begin{cases} C_{ii} = \text{cost}(i) \\ C_{ij} = \sum_{(s_p, s_q) \in \mathcal{T}_i \times \mathcal{T}_j} \delta(s_p, s_q) \end{cases}$$

où $\text{cost}(i)$ est le coût associé à la trajectoire \mathcal{T}_i défini en section 7.2.4, et où :

$$\delta(s_p, s_q) = \begin{cases} 1 & \text{si } d(s_p, s_q) \leq \sqrt{2} \\ 0 & \text{sinon.} \end{cases}$$

Notons $f(i)$ la somme $\sum_{j \in [1, n], j \leq i} C_{ij}$. Cette somme sera nulle lorsque la trajectoire \mathcal{T}_i n'interfère avec aucune trajectoire. Les trajectoires sont totalement séparées les unes des autres lorsque $\sum_{i \in [1, n]} f(i) = 0$.

Le critère d'adaptation \mathcal{F} associé à un individu (c'est-à-dire un ensemble de n trajectoires) est défini par :

$$\mathcal{F} = \begin{cases} 1 + \frac{n}{1 + \sum_i C_{ii}} & \text{si } \sum_i f(i) = 0 \\ \frac{1}{\sum_i f(i)} & \text{si } \sum_i f(i) > 0 \end{cases}$$

Cette *fitness* sera donc inférieure à 1 si les trajectoires ne sont pas séparées. Elle sera supérieure à 1 lorsque les trajectoires n'interfèrent pas entre elles, et sera dans ce cas d'autant plus grande que les écarts aux trajectoires préférentielles seront faibles. Notre objectif sera donc de maximiser cette *fitness*, minimisant ainsi le nombre d'interférences et le coût cumulé des déviations.

7.7.4 L'opérateur de croisement

Soient deux individus a et b , composés chacun de n trajectoires. L'opérateur de croisement retenu consiste à choisir, par tirage au sort, une des deux stratégies suivantes :

- Examiner pour chaque indice i inférieur à n si la trajectoire de a comporte moins d'interférences que celle de b , auquel cas on la copie dans b . Inversement, si b_i est de meilleure qualité que a_i , on la copie dans a_i . Pour comparer les deux trajectoires, on utilise un critère local d'adaptation (propre à chaque trajectoire) : la somme $f(i) = \sum_{j \in [1, n], j \leq i} C_{ij}$, que l'on a déjà vue précédemment. Si le critère local est identique, on ne change rien.
- Effectuer un croisement barycentrique des deux individus (cf chapitre 3, section 3.6 pour le détail de ce type de croisement).

Pour chaque croisement, on se donne une certaine probabilité d'adopter la première stratégie, plus élitiste que la deuxième, puisqu'on cherche délibérément à produire des descendants plus performants, ou alors de choisir la stratégie introduisant plus de variations aléatoires. Le choix parmi la population des couples d'éléments que l'on choisit de croiser est lui-même aléatoire, la probabilité de croisement étant un paramètre du programme.

7.7.5 L'opérateur de mutation

La probabilité de mutation d'un élément de population est également un paramètre du programme. L'opérateur de mutation choisi fonctionne de la manière suivante :

1. une trajectoire est choisie aléatoirement parmi les n de l'individu muté. Il est possible éventuellement de choisir parmi celles dont le critère local d'adaptation est le plus mauvais.
2. on tente de calculer une trajectoire séparée des autres, la plus proche possible de la trajectoire préférentielle. En cas de réussite, cette trajectoire vient remplacer celle choisie.
3. en cas d'échec de la tentative précédente, des variations aléatoires (bruit) sont introduites dans la trajectoire choisie.

L'étape consistant à calculer une trajectoire séparée des autres est réalisée au moyen de l'algorithme A^* présenté précédemment.

7.7.6 Résultats de l'algorithme génétique

Les mêmes configurations que dans la section 7.6.5 ont été testées avec l'algorithme génétique

Demi-cercle

Num. test	1	2	3	4	5	6	7	8	9	10
\mathcal{F}_{best}	1.1897	1.1897	1.1901	1.1897	1.1885	1.1885	1.1897	1.1893	1.1901	1.1885
Coût	51.735	51.735	51.620	51.735	52.068	52.068	51.735	51.842	51.620	52.068
Num. Gén.	71	125	116	89	75	78	109	98	84	78
Temps (s)	591.03	1307.13	1033.93	848.83	742.89	894.82	1124.39	1107.01	714.64	728.49

TAB. 7.1 – Résultats numériques de l'algorithme génétique pour 10 flux en demi-cercle (DC).

L'algorithme a été lancé 10 fois de suite avec des valeurs différentes pour la racine du générateur aléatoire. La population de 250 éléments évolue sur 150 générations, avec une probabilité de croisement de 0.6 et une probabilité de mutation de 0.05. Les résultats sont détaillés dans le tableau 7.1, qui donne les différentes valeurs du critère d'adaptation, le coût, le numéro de génération à laquelle la meilleure solution a été trouvée, ainsi que le temps de calcul (CPU Xeon 2.8 GHz).

On voit que, dans tous les cas, l'algorithme génétique fournit des résultats meilleurs que ceux de l' A^* (cf section 7.6.5) ou équivalents, ce à quoi on pouvait s'attendre, étant donné que le but de l' A^* n'est pas de résoudre le problème global. Les temps de calcul sont bien sûr plus longs.

Les figures 7.15 et 7.14 illustrent la meilleure solution trouvée. On voit que les déviations latérales des trajectoires 8 et 9, combinées avec des déviations verticales décalées, permet aux trajectoires 7, 8 et 9 de se croiser sur seulement 2 niveaux de vol. Ces trois trajectoires sont détaillées sur la figure 7.16. Cette solution permet de séparer toutes les trajectoires en occupant seulement 8 niveaux de vol au lieu des 10 de la solution trouvée par l' A^* .

Cercle

Pour le problème des 10 flux en cercle, la même démarche de test a été utilisée, avec une dizaine d'essais avec des valeurs différentes pour la racine du générateur aléatoire. On constate sur le tableau 7.2 que les solutions trouvées sont, comme pour le problème en demi-cercle, meilleures ou équivalentes à celle trouvée par l' A^* , dont la *fitness* est de 1.174859, avec un coût de 56.19546

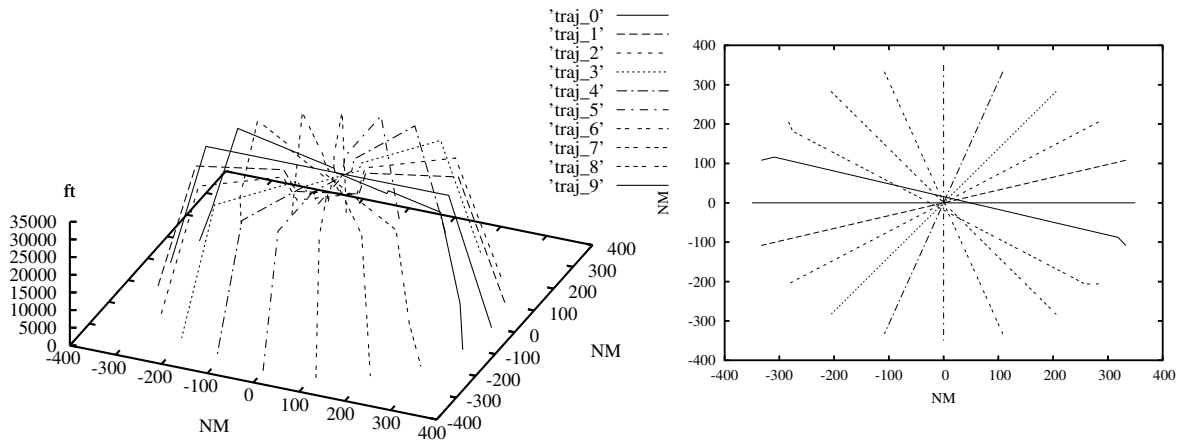


FIG. 7.14 – Résultats de l’algorithme génétique pour 10 flux en demi-cercle (distances horizontales en milles nautiques ; altitudes en pieds).

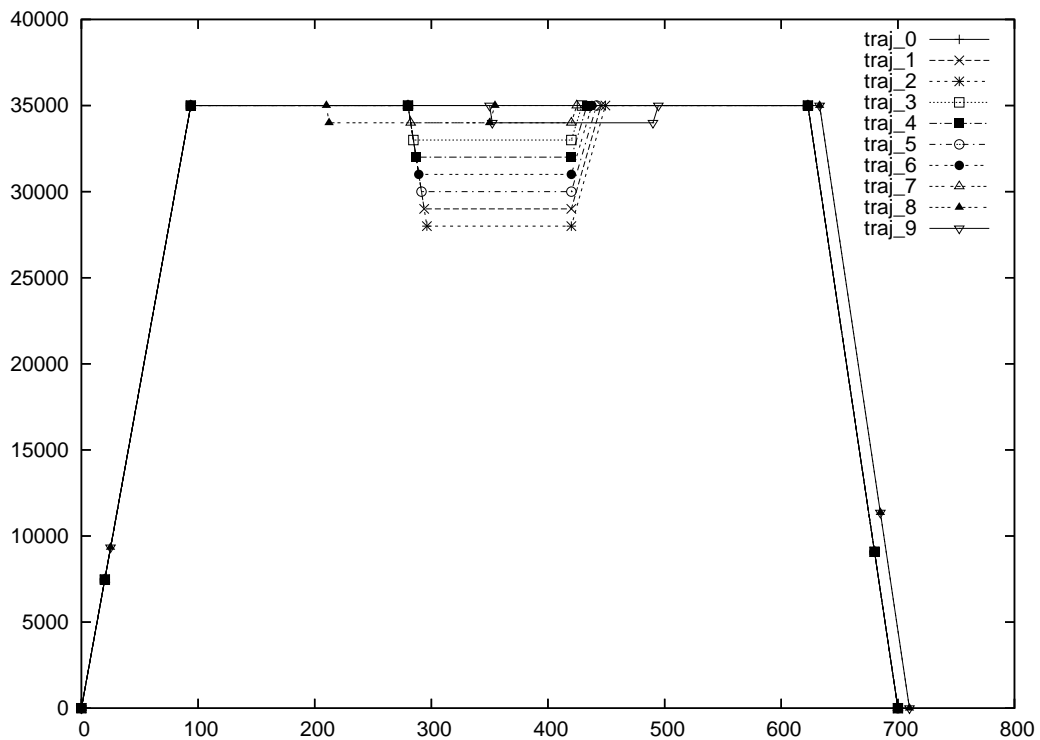


FIG. 7.15 – Profils verticaux des trajectoires trouvées par l’algorithme génétique pour 10 flux en demi-cercle (en abscisse : distance parcourue, en milles nautiques ; en ordonnée : altitude en pieds).

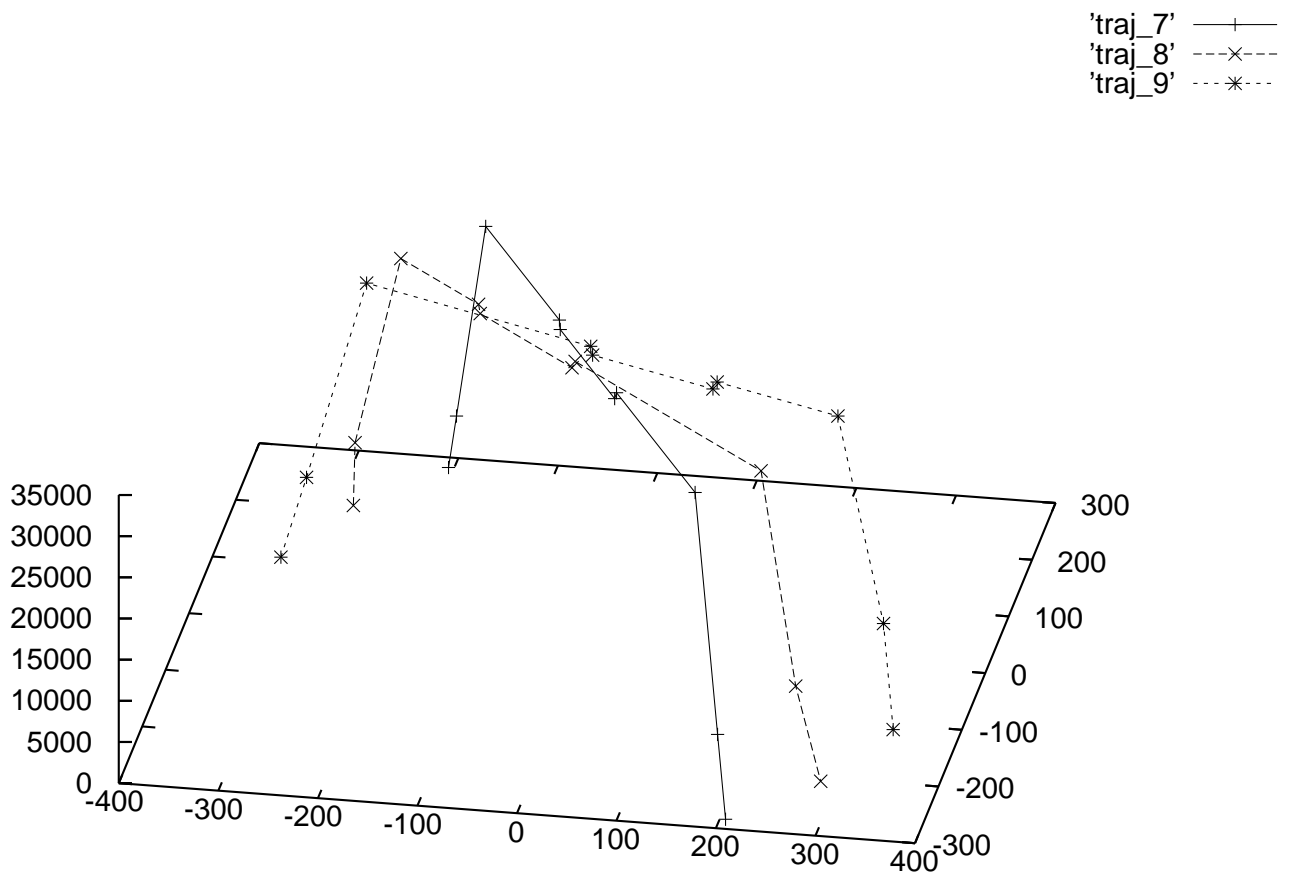


FIG. 7.16 – Détail des trajectoires 7, 8 et 9 du meilleur résultat de l'algorithme génétique pour 10 flux en demi-cercle (distances horizontales en milles nautiques ; altitude en pieds).

Num. test	1	2	3	4	5	6	7	8	9	10
\mathcal{F}_{best}	1.1774	1.1755	1.1755	1.1749	1.1761	1.1767	1.1761	1.1774	1.1749	1.1755
Coût	55.367	55.978	55.973	56.195	55.781	55.585	55.791	55.367	56.195	55.995
Num. Gén.	96	61	85	70	110	84	86	108	51	83
Temps (s)	485.05	323.84	474.72	382.39	572.96	370.00	463.49	629.37	268.60	452.61

TAB. 7.2 – Résultats numériques de l'algorithme génétique pour 10 flux sur un cercle (C).

Quadrillage

Num. test	1	2	3	4	5	6	7	8	9	10
\mathcal{F}_{best}	1.3743	1.3946	1.3946	1.3664	1.3874	1.3650	1.3946	1.3946	1.3806	1.3946
Coût	25.718	24.349	24.349	26.230	24.814	26.400	24.349	24.349	25.280	24.349
Num. Gén.	68	53	79	76	67	99	92	150	69	74
Temps (s)	409.97	333.94	541.95	501.30	389.23	701.95	420.57	819.22	527.32	463.06

TAB. 7.3 – Résultats numériques de l'algorithme génétique pour 10 flux sur un quadrillage (Q).

Pour le problème du quadrillage, les solutions trouvées par l'algorithme génétique (voir tableau 7.3) sont au mieux équivalentes (cinq fois sur dix) à celle trouvées par l' A^* avec le premier ordre considéré (tous les flux dans une direction, puis tous ceux dans l'autre), qui semble être une solution optimale. Pour cet ordre, rappelons que le coût de la solution trouvée était de 24.349, ce qui correspond à un critère d'évaluation de 1.3946.

Toutes les solutions de l'algorithme génétique sont meilleures que celle de l' A^* avec l'ordre alternatif (coût 27.964, fitness 1.3453).

7.8 Conclusion sur l'étude du problème simplifié

Les deux algorithmes donnent des résultats satisfaisants sur les configurations testées. Les solutions trouvées par l'algorithme génétique sont de meilleure qualité que celles trouvées par l' A^* , ce qui est normal puisque ce dernier est appliqué avec une stratégie *1 contre n* minimisant successivement les coûts individuels des trajectoires, alors que le premier cherche à minimiser un coût global.

Pour l' A^* , l'ordre dans lequel sont calculées les trajectoires n'est pas neutre, et il faudra donc s'intéresser à cet aspect lors de l'application au trafic réel : dans quel ordre faudra-t-il parcourir les trajectoires ; quel(s) critère(s) choisir pour déterminer cet ordre ?

Les résultats des algorithmes sont bons, mais il reste encore du chemin à parcourir avant de les appliquer au trafic réel. Notre modèle simplifié doit être enrichi, notamment :

- en prenant en compte le cas des flux entrants ou sortants qui n'aboutissent pas sur un aéroport de la zone géographique considérée.
- en utilisant les performances réelles des avions,
- en prenant en compte les incertitudes liées à la disparité des performances des avions appartenant à un même flux.

Le chapitre suivant traite de ces problèmes, et de l'application aux données réelles.

Chapitre 8

Définition d'un réseau statique de trajectoires 3D séparées, en France et en Europe

8.1 Introduction

Dans ce chapitre, c'est un problème proche du sujet de l'étude TOSCA ([MGZK98]) que nous étudions. Cette étude cherchait à évaluer l'intérêt et la faisabilité d'un concept de réseau de routes prioritaires directes entre les principales TMAs européennes. Ce nouveau réseau s'ajouterait au réseau de routes actuelles, mais serait géré indépendamment du système ATC classique, par des outils de séquençement spécifiques (DMAN et AMAN).

L'intérêt du concept réside dans le fait que, même si l'on ne peut gérer ainsi qu'une faible proportion du trafic, on peut espérer un gain considérable en termes de délais aux décollage pour le reste du trafic, à condition que l'impact du nouveau réseau sur la capacité globale du système ne soit pas trop important.

Cependant, dans cette étude, seule la séparation dans le plan horizontal de ces routes prioritaires est abordée : le croisement des routes du nouveau réseau est soit interdit, soit limité au maximum sans que l'on précise dans ce cas comment seraient gérés les conflits entre avions sur des routes sécantes.

Dans la suite de ce chapitre, les algorithmes présentés dans le chapitre précédent sont utilisés pour construire des trajectoires 3D reliant les origines et destinations des principaux flux en France et en Europe, de façon à ce qu'elles soient les plus directes possibles, sans qu'elles interfèrent entre elles. Concernant l'Europe, c'est en pratique à une zone géographique plus étendue que nous allons nous intéresser : celle pour laquelle il existe une description des secteurs et balises du réseau aérien dans les données européennes. Après la définition et l'analyse des flux de trafic, l'enrichissement de la modélisation simplifiée du chapitre précédent est abordé à travers les points suivants :

- la prise en compte des performances réelles des avions dans les phases de montée ou de descente,
- une détection d'interférences entre trajectoires plus précise, et pouvant tenir compte de la disparité des performances des avions,
- l'adaptation des algorithmes au cas des flux n'aboutissant pas à un aéroport situé dans la zone géographique considérée.

Enfin ce chapitre se terminera sur une présentation des résultats obtenus pour les flux de trafic à l'échelle de l'espace aérien français, puis européen.

8.2 Définition des flux origine-destination

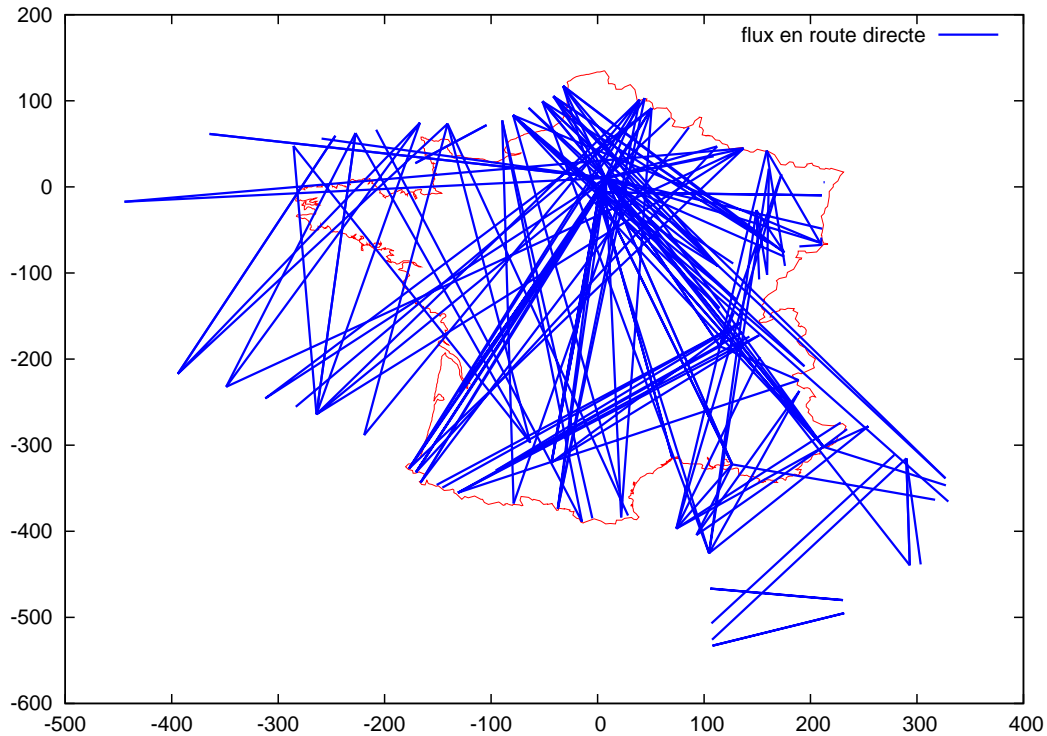


FIG. 8.1 – Flux de plus de 10 avions dans l'espace aérien français (trafic du 21/06/2002 ; projection stéréographique centrée sur Paris ; distances en milles nautiques) .

Dans un premier temps, nous définirons simplement un flux de trafic par la donnée de deux points : l'origine et la destination. Le flux est alors l'ensemble des avions volant entre ces deux points pendant une période de temps donnée. Les points d'origine et de destination peuvent être soit les aéroports de départ ou d'arrivée, ou encore les points d'entrée ou de sortie de l'espace aérien considéré.

Notons que cette définition est plus restrictive que les définitions informelles couramment utilisées dans le monde de l'*Air Traffic Flow Management*, où un flux pourra être défini comme un flot d'avions entre une zone géographique et une autre et passant par un point ou un secteur donné par exemple, ou encore un segment de route. Ce sont les débits de ces flux que l'ATFM cherche à réguler en imposant des délais au décollage, afin de protéger les secteurs de contrôle potentiellement surchargés. Mais ce n'est pas le problème que nous aborderons dans ce qui suit, où nous ferons complètement abstraction de la sectorisation, voire dans un premier temps du réseau de routes.

Dans la suite de ce chapitre, les flux (*origine, destination*) sont calculés à partir des données réelles de trafic, pour la zone France ou la zone Europe. Les figures 8.1 et 8.2 montrent les flux de 10 avions en France et en Europe respectivement (les distances sont en milles nautiques).

Pour résumer, un flux sera donc défini par un couple de points (*origine, destination*), chaque point étant caractérisé par :

- ses coordonnées géographiques,
- sa nature : aéroport, ou balise d'entrée ou de sortie

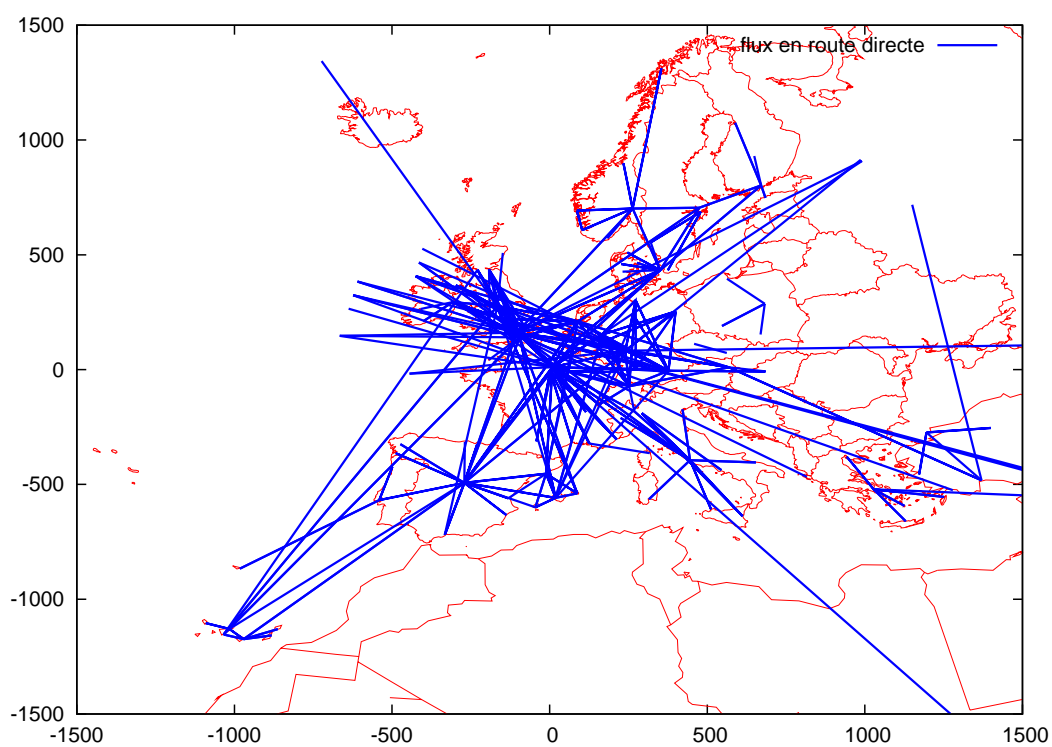


FIG. 8.2 – Flux de plus de 10 avions dans l'espace aérien européen (trafic du 21/06/2002 ; projection stéréographique centrée sur Paris ; distances en milles nautiques).

8.3 Analyse des flux origine-destination

Avant de tenter d'allouer des trajectoires 3D séparées aux différents flux, commençons par une rapide analyse de ceux-ci au-dessus de la France et de l'Europe. Une seule journée de trafic est ici analysée, et on s'intéressera uniquement à la demande initiale de trafic, c'est-à-dire aux plans de vols répétitifs, ou déposés par les compagnies aériennes au plus tard l'avant-veille du jour du vol, avant régulation du trafic (les éventuels délais au décollage n'ont pas encore été décidés). Le lecteur pourra se reporter à l'étude TOSCA ([MGZK98]) pour une analyse statistique plus complète sur l'Europe, portant sur plusieurs mois de trafic.

Les flux sont calculés selon les points d'entrée ou de sortie dans l'espace aérien (des aéroports ou des balises frontalières), puis ordonnés par taille décroissante. La taille d'un flux est tout simplement le nombre d'avions appartenant au flux. Dans cette analyse, nous commencerons tout d'abord par regarder quelle quantité d'avions représentent les n premiers flux de la liste ordonnée, ce qui nous donnera une idée du nombre de flux à prendre en compte pour espérer avoir un impact significatif sur l'écoulement du trafic.

Nous examinerons également la distribution des flux en fonction de leur longueur (la distance parcourue), et surtout la distribution des niveaux de croisière demandés en fonction de la distance parcourue. Ce dernier point est particulièrement intéressant. Jusqu'à présent nous n'avons alloué dans le chapitre 7 qu'une seule trajectoire par flux, pour le problème simplifié, où chaque flux n'était représenté que par un seul *avion représentatif*. Pour le problème réel, il nous faudra tenir compte à terme de la disparité des types d'avions, et donc en particulier des performances de montée et de descente. Cette disparité se traduit entre autres par le fait qu'un avion aux performances médiocres ne demandera pas un niveau de croisière trop haut, qu'il ne pourra jamais atteindre.

En n'allouant qu'une seule trajectoire 3D par flux, on court le risque soit d'allouer des niveaux de vols que certains types d'avions ne pourront pas atteindre, soit de s'imposer volontairement des niveaux de vol suffisamment bas pour que tous les avions du flux les atteignent, soit encore de définir des zones d'incertitude autour de chaque trajectoire, prenant en compte la diversité des performances des avions sur chaque flux. Dans le premier cas, les avions insuffisamment performants risquent d'entrer en conflit avec les avions d'autres flux. Dans le deuxième, on pénalise les avions les plus performants pour qui il serait plus économique de voler plus haut. Dans le dernier cas, on peut être amené à définir des zones d'incertitude englobant plusieurs niveaux de croisière. Une solution plus efficace serait éventuellement de définir pour chaque flux plusieurs trajectoires 3D, avec des niveaux préférentiels de croisière différents. Chaque avion du flux emprunterait alors l'une ou l'autre de ces trajectoires, en fonction de ses performances.

Pour estimer le nombre idéal de trajectoires 3D qu'il faudrait allouer par flux origine-destination, on fait l'hypothèse que chaque avion demande *a priori* le niveau de croisière le plus adapté à ses performances et à la distance qu'il a à parcourir, et on examine sur le trafic réel la distribution des niveaux demandés pour chacun des flux. Etant donné le nombre de flux, on synthétise cette information en calculant la distribution des niveaux demandés en fonction de la distance parcourue, sachant que celle-ci est certainement l'autre facteur déterminant, en dehors des performances, dans le choix d'un niveau de croisière.

8.3.1 Analyse des flux en Europe

La journée du 21 juin 2002 comportait 29687 plans de vol dans l'espace aérien européen, répartis en 11313 flux origine-destination. Ces flux sont triés par taille décroissante, ce qui permet ensuite de calculer la quantité cumulée de trafic pour un flux comprenant au moins n avions. La courbe de

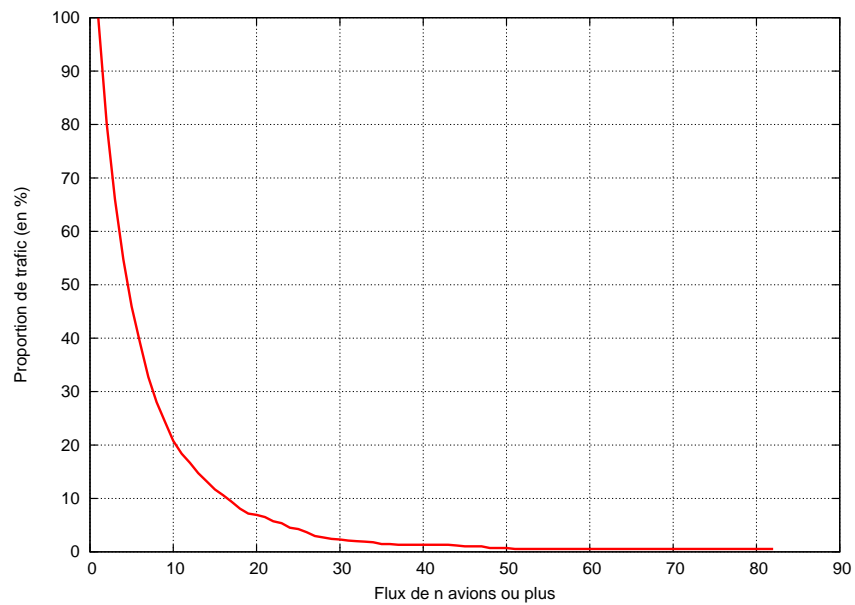
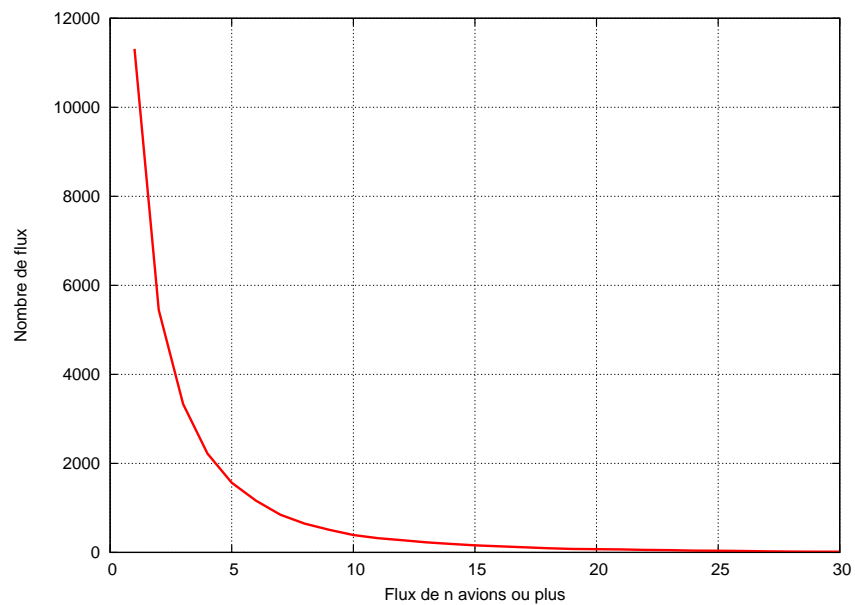


FIG. 8.3 – Europe : pourcentage de trafic en fonction de la taille du flux.

FIG. 8.4 – Europe : nombre de flux en fonction de la taille choisie (n avions ou plus par flux).

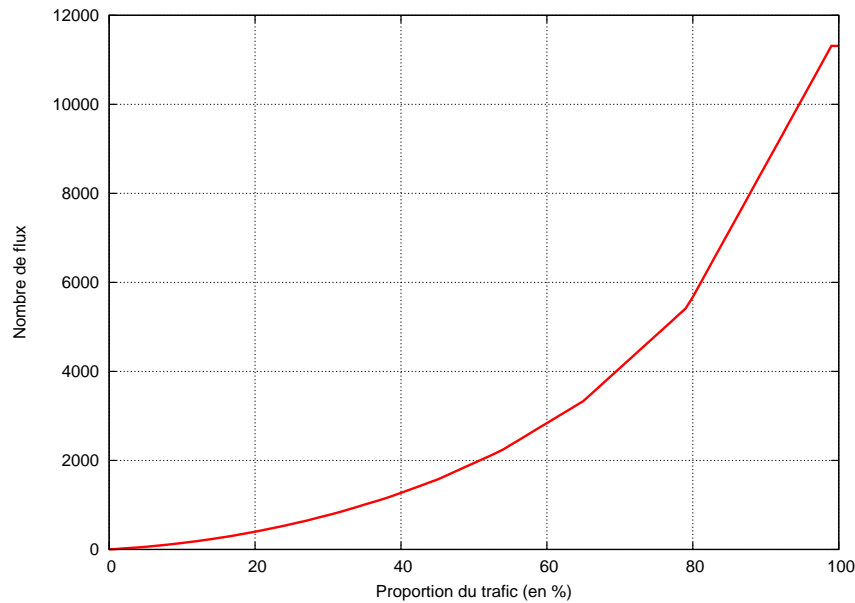


FIG. 8.5 – Europe : nombre de flux à considérer, en fonction du pourcentage de trafic que l'on souhaite traiter.

répartition ainsi obtenue est représentée sur la figure 8.3, avec en abscisse les valeurs de n , et en ordonnée la proportion de trafic, en pourcentage, que représentent les flux d'au moins n avions. Les flux de 5 avions ou plus représentent 46% du trafic total, ceux de 10 avions ou plus en représentent 20,8%, et pour les flux de 20 avions ou plus on est seulement à 6,9% du trafic total.

Le nombre de flux en fonction de la taille de flux choisie est représenté sur la figure 8.4. Il y a 74 flux de 20 avions ou plus, 392 flux de 10 avions ou plus, 1567 flux de 5 avions ou plus. Sur le total des 11313 flux, 5865, soit plus de la moitié, ne contiennent qu'un seul avion.

La figure 8.5 présente directement le nombre de flux à prendre en compte (dans notre liste triée par ordre décroissant) en fonction de la quantité de trafic que l'on souhaite traiter. On voit que pour traiter 50% du trafic européen, il faut considérer près de 2000 flux les plus importants. Ces considérations, que l'on peut retrouver dans l'étude TOSCA ([MGZK98]), justifient la démarche adoptée dans cette étude consistant à regrouper les aéroports dans des TMAs d'assez grande taille, avant de chercher des routes directes entre celles-ci. Ce regroupement permet de définir moins de routes prioritaires, en traitant plus de trafic, mais nécessite en contrepartie d'organiser le trafic dans les TMAs. Dans l'immédiat, nous nous en tiendrons aux flux entre aéroports (ou balises d'entrée/sortie), sachant que même si un pourcentage relativement faible du trafic total est concerné, cela peut se traduire par une forte diminution des retards au décollage, comme l'affirme cette même étude. A terme il faudra toutefois certainement considérer les flux entre TMAs si l'on veut pouvoir affecter des trajectoires 3D séparées à une proportion significative de trafic.

La figure 8.6 montre le nombre de flux en fonction de la distance parcourue. Les flux sont comptabilisés par tranches de distance de 50 milles nautiques. On constate un pic entre 200 et 300 milles nautiques, puis une lente décroissance, les flux de longueur supérieure à 2000 milles étant quantité négligeable. La figure 8.7 présente la courbe du nombre de vols en fonction de la distance parcourue.

Concernant la répartition des flux en Europe selon la nature des points d'origine ou de destination (aéroport de **Départ** ou d'**Arrivée**, ou **Balise** d'entrée ou de sortie), on constate sur le tableau 8.1 qu'une écrasante majorité (plus de 85%) des flux et des vols est purement intra-européen.

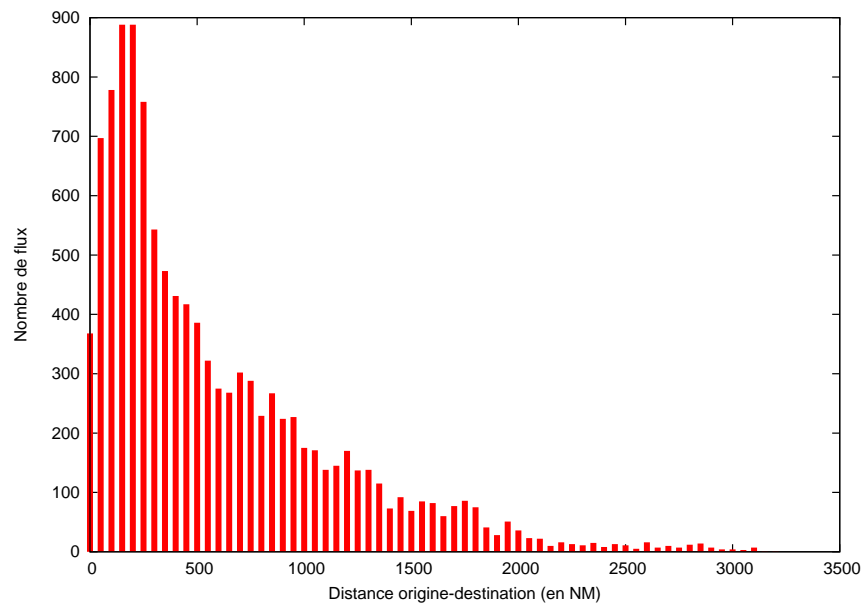


FIG. 8.6 – Europe : distribution du nombre de flux en fonction de la distance parcourue.

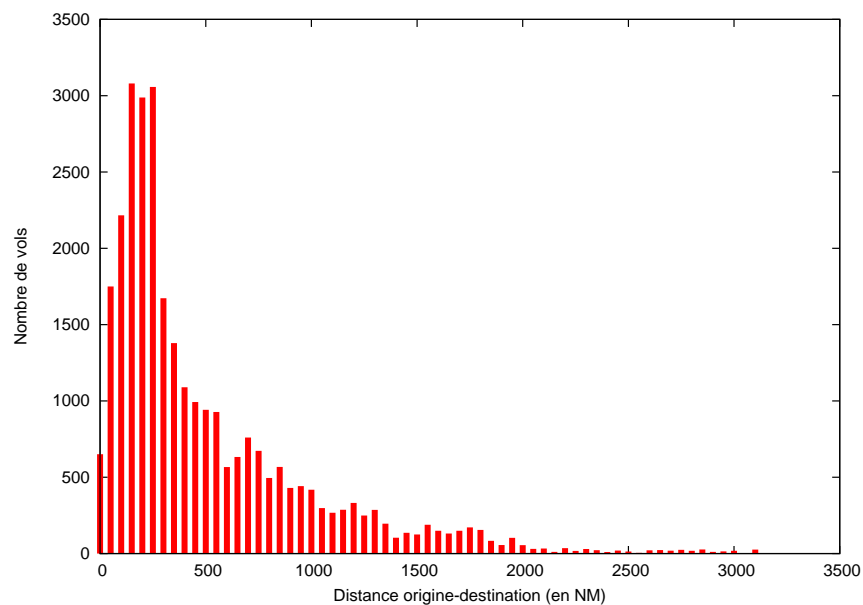


FIG. 8.7 – Europe : distribution du nombre de vols en fonction de la distance parcourue.

	TOUS FLUX				
	Total	Dép -> Arr	Dép -> Bal	Bal -> Arr	Bal -> Bal
Flux	11313	10017	631	656	9
% des flux	100 %	88.5 %	5.6 %	5.8 %	0.1 %
Vols	29687	26700	1446	1531	10
% des vols	100 %	89.9 %	4.9 %	5.2 %	0.03 %

TAB. 8.1 – Répartition du trafic en Europe selon la nature de l'origine et de la destination.

Voyons maintenant la distribution des niveaux de croisière demandés. Pour l'obtenir, on comptabilise le nombre d'occurrences pour chaque valeur de niveau de vol et pour chaque tranche de distance (0 à 50 NM, puis 50 à 100 NM, etc...). Sur la figure 8.8 sont représentés les niveaux de croisière en fonction de la distance parcourue. Un point est affiché chaque fois que le nombre d'occurrences est supérieur à 5% du total des vols dans la tranche de distance. On constate que pour les faibles distances, les niveaux de croisière les plus demandés semblent s'élever en fonction de la distance à parcourir. Pour les longues distances, on a un nuage de points relativement difficile à exploiter. Il nous manque en fait la représentation du nombre d'occurrences pour chaque point.

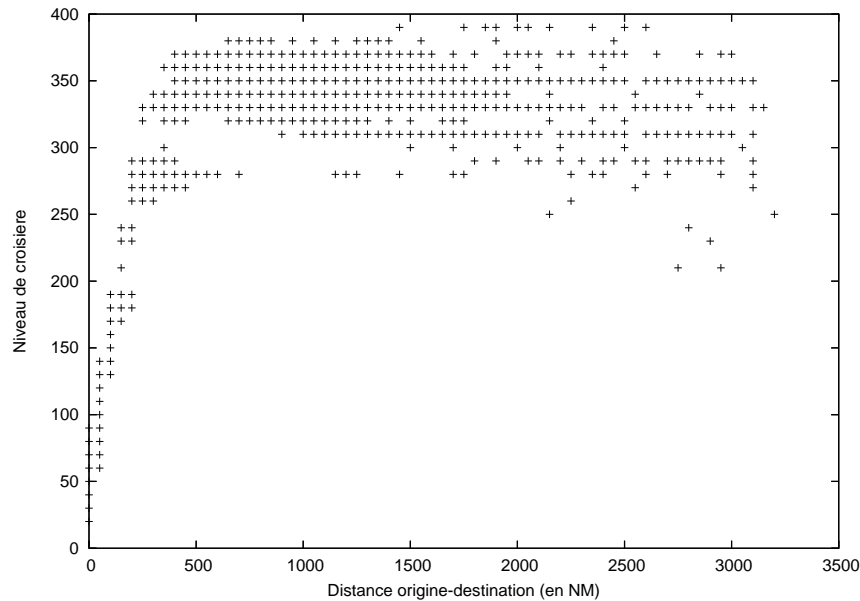


FIG. 8.8 – Europe : niveaux de croisière demandés (seuil de représentation à 5% du total de la classe), en fonction de la distance parcourue.

La figure 8.9 présente pour chaque tranche de distance le nombre de demandes pour chaque niveau de vol. Cette vue tridimensionnelle nous permet de mieux nous rendre compte des hauteurs relatives des différents pics de demandes, mais elle reste relativement peu lisible. Plutôt que de représenter le nombre de demandes, visualisons le pourcentage du nombre de demandes par rapport au total des demandes dans chaque tranche de distances. La figure 8.10 superpose ces courbes de pourcentages pour les quinze premières tranches de distances, et permet d'identifier clairement les pics, en particulier dans l'espace aérien supérieur. Les trois premières classes de distances sont un peu à part des autres. Pour les flux d'une longueur maximale de 50 milles nautiques, la quasi-totalité des demandes est entre les niveaux FL20 et FL120, avec soit un seul pic assez large autour du FL50, soit deux pics assez proches, l'un au FL50 et l'autre au FL80. Les courbes des tranches 50-100 NM et 100-150 NM semblent confirmer la deuxième hypothèse, car elles présentent toutes deux un pic aux alentours du FL80. Entre 50 et 100 milles nautiques, on a donc un pic au FL80 et un autre au FL120, presque confondus en un seul dont la base est assez large. Disons que le gros de la demande est entre le FL50 et le FL150. Sur la courbe de la tranche 100-150 NM, on peut identifier un petit pic au FL80, deux pics plus importants au FL140 et au FL180, et un dernier plus modeste au FL230.

Pour les autres classes de distances, on identifie cinq pics de demandes, autour des niveaux FL180, FL230, FL280, FL330, et FL370. Pour les moyennes distances, l'importance relative de ces pics dépend de la distance : plus on va loin, plus les niveaux de croisière demandés sont élevés, comme

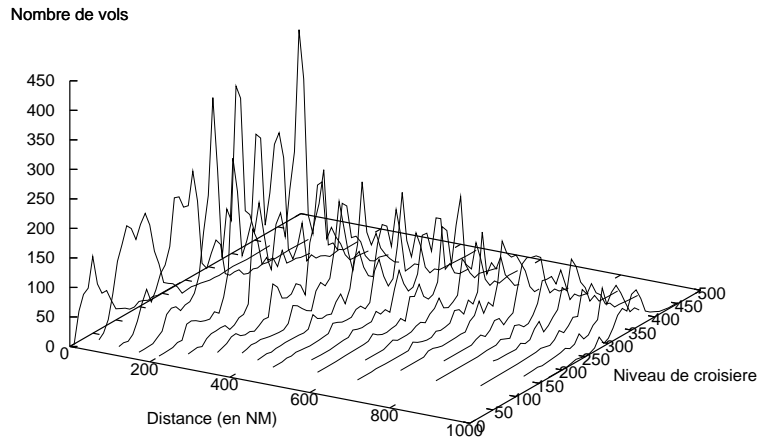


FIG. 8.9 – Europe : nombre d’occurrences des niveaux de croisières demandés en fonction de la distance parcourue.

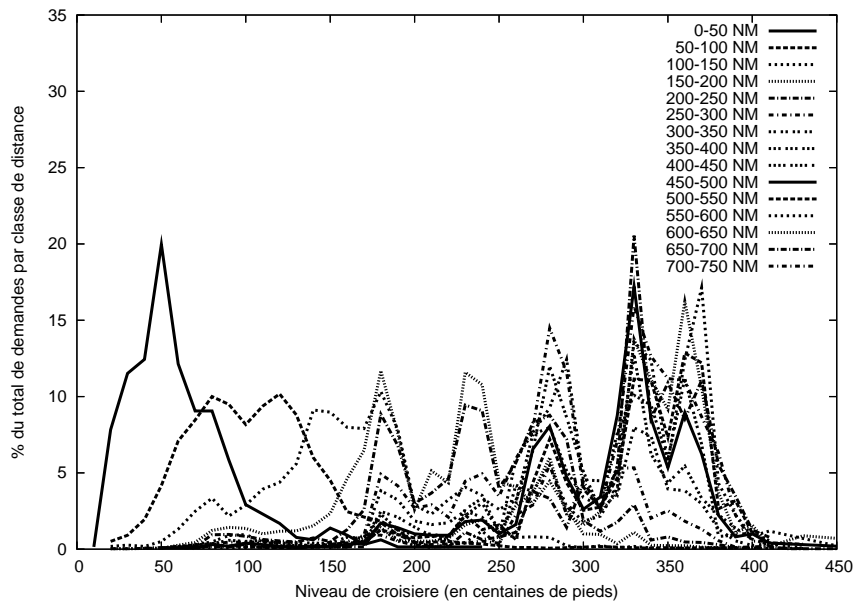


FIG. 8.10 – Europe : pourcentage des demandes (par rapport au total de chaque classe) par niveau de croisière, pour les 15 premières tranches de distances.

on pouvait s'y attendre. Ainsi on a deux pics majeurs aux FL180 et FL230 pour la tranche 150-200 NM, trois pics majeurs aux FL180, FL230, et FL280 pour la tranche 200-250 NM, un pic majeur au FL280 et trois plus modestes aux FL180, FL230, et FL330 pour la tranche 250-300 NM, deux pics majeurs aux FL280 et FL330 pour les tranches 300-350 NM et 350-400 NM. Au-delà des 400 milles nautiques, ce sont les niveaux autour des FL280, FL330 et FL370 qui sont les plus demandés, le pic autour du FL280 diminuant progressivement avec la distance. Remarquons l'existence d'une certaine demande, assez faible certes, autour du FL400.

8.3.2 Analyse des flux en France

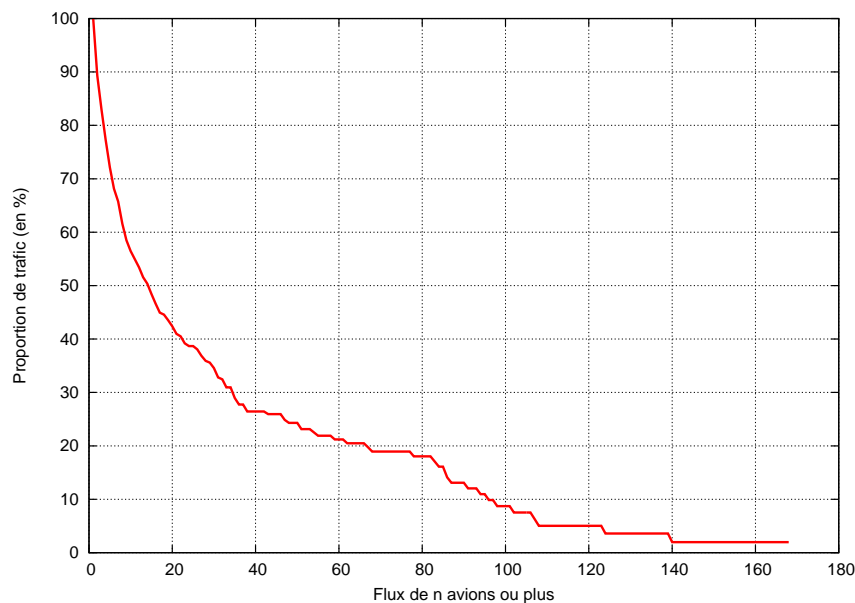


FIG. 8.11 – France : pourcentage de trafic en fonction de la taille du flux.

Pour l'analyse des flux dans l'espace aérien français, nous nous intéresserons seulement à l'aspect quantitatif du nombre de flux et du pourcentage de trafic traité. En effet, la distribution des niveaux de croisière en fonction de la distance est biaisée par la taille limitée de la zone géographique considérée, et il faudrait donc plutôt considérer les distances entre aéroports, ce qui nous ramènerait à une analyse similaire à celle du trafic européen, avec des conclusions identiques.

La journée du 21 juin 2002 comprenait 8540 vols au-dessus de la France, répartis en 1826 flux origine-destination. Ces flux sont triés par taille décroissante, comme précédemment, et on calcule successivement, pour toutes les valeurs de n , la proportion de trafic que représentent les flux à n avions ou plus. La courbe correspondante est représentée en figure 8.11. La figure 8.12 donne quant à elle le nombre de flux pour chaque taille de flux (n avions ou plus). Enfin, la figure 8.13 montre le nombre de flux en fonction de la proportion de trafic qu'ils représentent.

On constate sur ces trois figures que les flux dans l'espace français concentrent beaucoup plus le trafic que les flux dans la zone Europe. Les 77 flux de 20 avions ou plus représentent déjà 42,4% du trafic total. Les 166 flux de 10 avions ou plus en représentent 56%, et les 371 de 5 avions ou plus 72,1%. L'explication tient au fait que l'on a pris comme points d'entrée ou de sortie les balises existantes, utilisées pour le trafic réel déjà organisé. Si l'on avait recalculé les points d'entrée/sortie en prenant l'intersection des routes directes entre aéroports de départ et d'arrivée avec les limites de

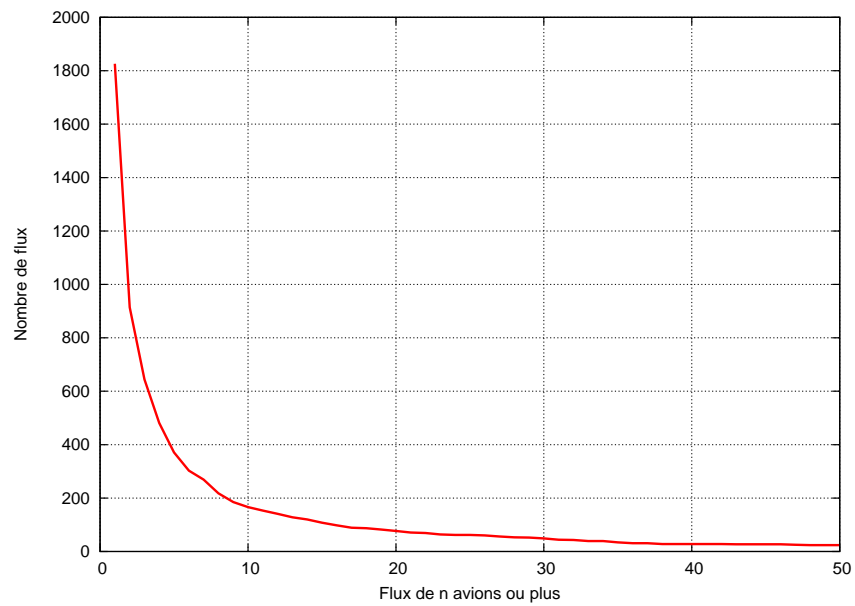
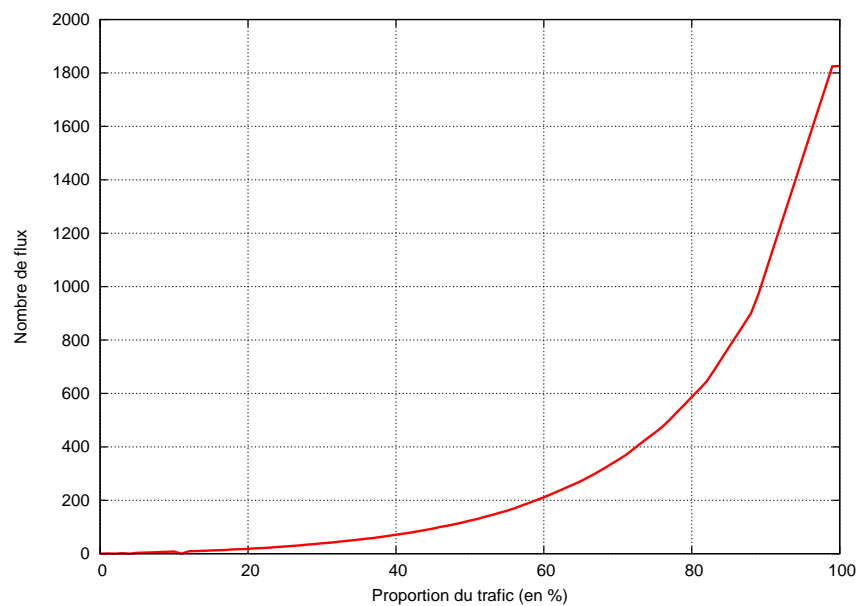
FIG. 8.12 – France : nombre de flux en fonction de la taille choisie (n avions ou plus par flux).

FIG. 8.13 – France : nombre de flux à considérer, en fonction du pourcentage de trafic que l'on souhaite traiter.

l'espace aérien français, on obtiendrait probablement un nombre beaucoup plus important de points, et donc de flux, avec une plus grande dispersion du trafic.

	TOUS FLUX				
	Total	Dép -> Arr	Dép -> Bal	Bal -> Arr	Bal -> Bal
Flux	1826	832	328	323	343
% des flux	100 %	45,6 %	18,0 %	17,7 %	18,8 %
Vols	8540	2028	1563	1601	3348
% des vols	100 %	23,7 %	18,3 %	18,7 %	39,2 %

TAB. 8.2 – Répartition du trafic en France selon la nature de l'origine et de la destination, pour tous les flux.

	FLUX DE 10 AVIONS OU PLUS				
	Total	Dép -> Arr	Dép -> Bal	Bal -> Arr	Bal -> Bal
Flux	166	21	30	28	87
% des flux	100 %	12,7 %	18,1 %	16,9 %	52,4 %
Vols	4823	379	875	917	2652
% des vols	100 %	7,9 %	18,1 %	19,0 %	55,0 %

TAB. 8.3 – Répartition du trafic en France selon la nature de l'origine et de la destination, pour les flux de 10 avions ou plus.

	FLUX DE 20 AVIONS OU PLUS				
	Total	Dép -> Arr	Dép -> Bal	Bal -> Arr	Bal -> Bal
Flux	77	6	11	12	48
% des flux	100 %	7,8 %	14,3 %	15,6 %	62,3 %
Vols	3620	181	629	700	2110
% des vols	100 %	5,0 %	17,4 %	19,3 %	58,3 %

TAB. 8.4 – Répartition du trafic en France selon la nature de l'origine et de la destination, pour les flux de 20 avions ou plus.

Concernant la nature des points d'origine ou de destination (aéroport de **Départ** ou d'**Arrivée**, ou **Balise** d'entrée ou de sortie), la répartition des flux selon ce critère est décrite dans les tableaux 8.2 pour tous les flux, 8.3 pour les flux de 10 avions ou plus, et 8.4 pour les flux de 20 avions ou plus. On constate que le trafic purement domestique (départ et arrivée en France) représente près de la moitié des flux (45,5%), mais seulement 23,7% des vols. Les flux concernés contiennent donc un nombre d'avions peu élevé en moyenne. A l'inverse, les survols purs (colonne Bal -> Bal, à droite) représentent 39,2% des vols mais seulement 18,8% des flux. Ce sont donc plutôt des gros flux. Ces impressions se confirment lorsque l'on se restreint aux flux de 10 avions ou plus, ou de 20 avions ou plus, pour lesquels plus de la moitié des flux et des vols sont des survols purs.

8.3.3 Conséquences sur la problématique

Au vu de la rapide analyse que nous venons de faire sur une seule journée de trafic, on constate finalement que considérer l'ensemble de la zone Europe ou seulement la zone France mène à des

configurations de flux totalement différentes. Dans le premier cas, c'est à un trafic essentiellement intra-européen que l'on a affaire, et l'on sera dans une problématique finalement assez proche de celle du problème simplifié du chapitre 7. Par contre, lorsque l'on se restreint à un espace géographique plus petit, la France en l'occurrence, la proportion du trafic entrant ou sortant de l'espace aérien via des balises frontalières augmente considérablement. Il faut donc modifier les algorithmes du chapitre 7 : comme on s'interdit de changer de point d'entrée ou de sortie, il faut pouvoir allouer des niveaux de vol séparés en entrée ou en sortie, pour les flux se terminant sur une même balise.

Un autre point intéressant, qui ressort de l'étude de la distribution des niveaux de vol demandés pour la croisière, est qu'il apparaît judicieux de considérer séparément les flux en espace aérien inférieur et ceux en espace aérien supérieur, voire même de ne tenter d'allouer des trajectoires 3D séparées que dans le supérieur. On n'invente là rien de nouveau, puisque dans la réalité opérationnelle le réseau de routes comme la sectorisation sont déjà fortement différenciés selon que l'on se trouve au-dessous ou au-dessus du FL195 (19500 pieds), niveau de coupure entre les espaces inférieur et supérieur. La séparation de chaque problème selon un niveau de coupure se justifie par l'analyse des flux (cf figure 8.10 et les commentaires associés) : on retrouve en espace inférieur des vols sur de courtes ou moyennes distances, avec des valeurs plutôt disparates pour les niveaux de croisière demandés (cf l'allure des distributions pour les distances inférieures à 150 milles nautiques), à l'exception d'un pic de demandes bien identifié autour du FL180. Par contre, en espace aérien supérieur, on trouve des vols sur de plus longues distances, avec une distribution des niveaux de croisière pour laquelle on identifie clairement des pics de demandes (FL230, FL280, FL330, et FL370 en espace supérieur). Notons que dans la définition des sous-problèmes inférieur et supérieur, on peut aussi choisir de placer le niveau de coupure au-dessous du FL 180, qui est un pic de demandes bien identifié que l'on peut prendre en compte comme les autres dans l'espace supérieur. Une bonne valeur de coupure dans ce cas semble être le FL145.

	Coupure au FL195		Coupure au FL145	
	Sol -> FL195	Au-dessus du FL195	Sol -> FL145	Au-dessus du FL145
Flux	805	1133	394	1477
Vols	1376	7164	579	7961
% des vols	16,11 %	83,89 %	6,78 %	93,22 %

TAB. 8.5 – Répartition des flux et du trafic en France selon le niveau de coupure entre espace inférieur et espace supérieur.

	Coupure au FL195		Coupure au FL145	
	Sol -> FL195	Au-dessus du FL195	Sol -> FL145	Au-dessus du FL145
Flux	3443	8515	1920	9674
Vols	7246	22441	3656	26031
% des vols	24.41 %	75.59 %	12.32 %	87.68 %

TAB. 8.6 – Répartition des flux et du trafic en Europe selon le niveau de coupure entre espace inférieur et espace supérieur.

Les tableaux 8.5 et 8.6 montrent, en France et en Europe, la répartition des flux et des vols en fonction du niveau de coupure choisi entre espaces inférieur et supérieur. Constatons qu'une écrasante majorité des vols (entre 75% et plus de 90%) demandent des niveaux de croisière au-dessus du niveau de coupure, ce qui incite à considérer en priorité l'espace supérieur dans notre optimisation des flux.

Pour le problème en espace inférieur (si l'on choisit de le traiter) on peut s'attendre à avoir quelques difficultés à identifier un ou plusieurs niveaux de vol préférentiels pour chaque flux origine-destination, au vu de l'allure des courbes de distribution des niveaux pour les courtes ou moyennes distances. D'un autre côté, les flux étant plus courts, ils ont statistiquement moins de chances de se croiser dans le plan, ce qui devrait faciliter la recherche de trajectoires séparées.

Pour le problème en espace supérieur, on a essentiellement deux options possibles pour le choix des niveaux préférentiels et la recherche de trajectoires 3D séparées :

- allouer une seule trajectoire 3D par flux, en prenant comme niveau préférentiel le niveau de croisière le plus demandé, par exemple (mais on pourrait aussi choisir un niveau parmi les plus hauts).
- allouer plusieurs trajectoires 3D par flux, en choisissant les niveaux préférentiels parmi les pics identifiés lors de l'étude des distributions (ou en refaisant une étude personnalisée pour chaque flux).

Dans le premier cas, si l'on veut rester réaliste, il faudra envisager de définir autour de la trajectoire de chaque *avion représentatif* une zone d'incertitude englobant les trajectoires réelles de tous les avions appartenant au flux considéré. La détection d'interférences entre trajectoires devra alors tenir compte des ces *tubes*, dont l'épaisseur verticale dépendra, dans les phases évolutives, des performances de montée ou de descente des différents types d'avions, et, dans les phases de croisière, des niveaux maximum atteignable par chaque type d'avion.

La deuxième option est potentiellement plus efficace, dans la mesure où on peut envisager d'intercaler des trajectoires appartenant à d'autres flux entre deux niveaux préférentiels d'un flux donné, mais elle nécessite de régler le problème des interférences entre deux trajectoires associées à un même flux : l'idéal serait d'inhiber la détection d'interférences pour les phases de montée initiale ou de descente finale, tout en s'assurant que les segments en croisière ou les évolutions intermédiaires restent séparées.

8.4 Modélisation des flux selon les évolutions verticales

La donnée des points d'origine et de destination des flux n'est pas suffisante pour calculer une trajectoire dans l'espace. Il faut donc affiner notre modélisation avant de chercher des trajectoires 3D non interférentes pour les flux les plus importants, en tenant compte dans notre problème de la dimension verticale. Concrètement, avant de tenter de calculer des trajectoires, il faudra définir des *niveaux de vols préférentiels* pour la croisière et, le cas échéant, pour l'entrée et/ou la sortie de l'espace aérien.

Plusieurs modélisations sont possibles en fonction du niveau de réalisme souhaité :

UNIC : définir une seule trajectoire 3D par flux origine-destination. Dans ce cas, on pourra choisir comme niveau de croisière le RFL le plus demandé. Les niveaux de vol en entrée ou en sortie sont alors fonction de la nature des points d'origine et de destination. Pour les balises d'entrée/sortie, les flux entreront ou sortiront stables en niveau, dans la mesure du possible. Pour les aéroports, le niveau de vol à l'extrémité du flux est nécessairement 0 (rappelons que pour l'instant, on considère que tous les aéroports sont à l'altitude 0),

PROX : définir plusieurs trajectoires 3D par flux origine-destination, en tenant compte des aéroports proches des frontières de l'espace. On tient compte ici du fait qu'un certain nombre d'avions sont susceptibles de sortir de l'espace aérien en configuration de descente vers un aéroport proche, ou bien d'entrer en montée en provenance d'un aéroport à proximité de la frontière. Pour chaque balise d'entrée, on sera donc susceptible de définir un niveau d'entrée stable en

croisière, un niveau d'entrée en montée, ou un niveau d'entrée en descente, et de même pour les balises de sortie.

MULTI : définir une trajectoire par sous-flux, chaque sous-flux étant caractérisé soit par un triplet (*niveau d'entrée, niveau de croisière, niveau de sortie*) pour les cas les plus courants, soit par un couple (*niveau d'entrée, niveau de sortie*) pour les cas dégradés, lorsque les avions n'ont pas la possibilité de faire une étape en croisière pendant leur transit dans l'espace aérien considéré. La différence par rapport au modèle précédent est que l'on peut définir plusieurs niveaux de croisière par flux origine-destination, et, le cas échéant, plusieurs niveaux d'entrée ou de sortie.

Les aéroports proches du modèle PROX sont simplement caractérisés par une distance entre aéroport et balise d'entrée ou de sortie, inférieure à une distance d_{prox} qui est un paramètre choisi. L'identification des sous-flux pour chaque couple origine-destination se fait donc de façon simple selon ce critère. Le niveau d'entrée ou de sortie pour les sous-flux évolutifs est choisi en prenant le niveau le plus bas au passage de la frontière, sur l'ensemble des vols concernés.

En revanche, pour le modèle MULTI, la détermination des sous-flux est en soi un problème de classification, pour lequel nous allons utiliser à nouveau la méthode de nuées dynamiques présentée au chapitre 3, section 3.3. Pour chaque flux origine-destination, les vols sont préalablement répartis en deux classes distinctes : celle des vols se stabilisant en croisière à l'intérieur de l'espace aérien, et celle des vols pour lesquels il n'est pas possible de faire une phase de vol stabilisé avant la sortie ou l'atterrissage. Pour cela, le profil vertical de chaque vol est calculé entre l'aéroport de départ et celui de destination, en fonction du modèle tabulé de performances utilisé dans les systèmes opérationnels européens (cf section 8.6, un peu plus loin). Les deux classes ainsi définies sont séparées en sous-classes en utilisant la méthode présentée au chapitre 3 (3.3), avec la définition classique du barycentre et la distance : $dist(A, B) = \max_i |x_i(A) - x_i(B)|$. Pour la première classe, la distance et le barycentre sont définis sur l'espace des triplets (EFL^1 , RFL^2 , XFL^3) associés à chaque vol. Pour la deuxième classe, on opère sur l'espace des couples (EFL , XFL). Dans chaque cas, c'est donc l'écart maximum calculé pour chacun des niveaux de vol représentatifs qui est retenu comme distance dans le critère d'agrégation.

Maintenant que sont définies les fonctions de distance et de barycentre, il reste à choisir les seuils d_1 , en-deçà duquel un point est ajouté à une classe, et d_2 , en-deçà duquel deux classes sont fusionnées. Dans notre application, un seuil sera donc un nombre de niveaux de vol. Ces seuils sont ajustés de façon empirique en fonction du nombre de sous-flux obtenus, et en tenant compte des résultats précédents sur la distribution des niveaux de vol demandés. Au vu des pics de demande constatés dans l'analyse des niveaux demandés, il apparaît judicieux de ne pas dépasser les 4 à 5 niveaux de vols pour le seuil d_1 si l'on souhaite distinguer ces pics dans les sous-flux calculés. Par ailleurs, des valeurs trop faibles peuvent aboutir à une multitude de sous-flux, auquel cas le processus d'agrégation perd son intérêt.

EFL	RFL	XFL	Evol. entrée	Evol. sortie	Nb.
BALSI -> MAMES					168
370	370	370	Stable	Stable	84
240	240	240	Stable	Stable	4
290	290	290	Stable	Stable	4
410	410	410	Stable	Stable	3

¹EFL : *Entry Flight Level*.

²RFL : *Requested Flight Level*.

³XFL : *eXit Flight Level*.

340	340	340	Stable	Stable	58
340	340	220	Stable	Descente	2
370	370	320	Stable	Descente	1
240	380	380	Montée	Stable	4
220	330	330	Montée	Stable	1
380	390	390	Montée	Stable	1
220	300	300	Montée	Stable	6
SITET -> PUMAL					139
360	360	360	Stable	Stable	57
270	270	270	Stable	Stable	2
230	230	230	Stable	Stable	1
330	330	330	Stable	Stable	12
340	340	250	Stable	Descente	8
370	370	200	Stable	Descente	1
370	370	330	Stable	Descente	1
370	370	260	Stable	Descente	1
330	350	350	Montée	Stable	13
320	410	410	Montée	Stable	1
260	270	270	Montée	Stable	1
240	390	390	Montée	Stable	2
290	310	310	Montée	Stable	2
260	350	350	Montée	Stable	15
280	370	370	Montée	Stable	3
380	450	270	Montée	Descente	1
290	330	260	Montée	Descente	4
300	370	210	Montée	Descente	1
270	370	200	Montée	Descente	2
290	370	260	Montée	Descente	1
260	270	260	Montée	Descente	1
250	340	240	Montée	Descente	3
330	370	200	Montée	Descente	6
XIDIL -> LFPG					112
240	240	0	Stable	Descente	76
290	290	0	Stable	Descente	12
380	380	0	Stable	Descente	3
330	330	0	Stable	Descente	10
180	180	0	Stable	Descente	11
AGN -> FAWBO					107
360	360	360	Stable	Stable	66
410	410	410	Stable	Stable	3
260	260	260	Stable	Stable	1
300	300	300	Stable	Stable	9
370	370	260	Stable	Descente	14
360	360	300	Stable	Descente	11

380	380	380	Stable	Descente	1
320	320	300	Stable	Descente	1
340	340	250	Stable	Descente	1
RAPOR -> LFPG					93
340	340	0	Stable	Descente	39
370	370	0	Stable	Descente	25
280	280	0	Stable	Descente	3
220	220	0	Stable	Descente	21
190	220	0	Montée	Descente	2
170	180	0	Montée	Descente	3
DERAK -> LFPG					101
370	370	0	Stable	Descente	30
400	400	0	Stable	Descente	5
260	260	0	Stable	Descente	7
300	300	0	Stable	Descente	4
340	340	0	Stable	Descente	55

TAB. 8.7: *Sous-flux des 5 flux principaux en France au dessus du FL145, définis selon les niveaux d'entrée, de croisière, et de sortie*

EFL	RFL	XFL	Evol. entrée	Evol. sortie	Nb.
LEMD -> LEBL					82
0	200	0	Montée	Descente	5
0	280	0	Montée	Descente	76
0	320	0	Montée	Descente	1
LEBL -> LEMD					82
0	170	0	Montée	Descente	2
0	220	0	Montée	Descente	1
0	290	0	Montée	Descente	79
EGLL -> DOGAL					50
0	350	350	Montée	Stable	38
0	380	380	Montée	Stable	6
0	290	290	Montée	Stable	6
LIML -> LIRF					47
0	250	0	Montée	Descente	47
LEBL -> LEPA					38
0	180	0	Montée	Descente	28
0	210	0	Montée	Descente	10
LIRF -> LIML					44
0	240	0	Montée	Descente	44

TAB. 8.8: *Sous-flux des 5 flux principaux en Europe au dessus du FL145, définis selon les niveaux d'entrée, de croisière, et de sortie*

Le tableau 8.7 montre, pour les cinq plus gros flux origine-destination en France le 21 juin 2002, les sous-flux du modèle MULTI obtenus en choisissant un seuil d_1 égal à trois niveaux de vol, et un seuil d_2 égal à un niveau de vol. Le tableau 8.8 montre les sous-flux calculés pour la même journée, en Europe.

On constate tout d'abord que les flux origine-destination en France se subdivisent en un grand nombre de sous-flux (23 pour SITET->PUMAL) avec différentes configurations d'entrée et de sortie, alors que pour les données Européennes on n'excède par trois sous-flux par origine-destination. Etant donné la taille de l'espace traité dans le cas européen, les niveaux d'entrée et de sortie n'entrent pas en ligne de compte pour les flux principaux : soit les avions décollent ou atterrissent en Europe, soit ils sont stables en niveau à l'entrée ou à la sortie. Le découpage en sous-flux correspond dans ce cas à des demandes différentes pour les niveaux de croisière, et on retrouve les différents pics de demandes constatés en section 8.3. En revanche, la taille de l'espace français est telle qu'il existe un certain nombre de vols évolutifs en entrée ou en sortie, à l'origine de la plus grande disparité des sous-flux. Paradoxalement, l'optimisation des trajectoires associées aux flux s'avèrera donc sans doute plus complexe dans l'espace aérien français que dans l'espace européen, en raison de la taille plus réduite de l'espace.

Par ailleurs, on constate que les vols sont en règle générale inégalement répartis entre les différents sous-flux d'un flux donné. Sur BALSIS->MAMES, on a deux principaux sous-flux correspondant à des survols stables au FL340 et au FL370, avec respectivement 58 vols et 84 vols, les neuf autres sous-flux ne dépassant pas chacun les 6 vols dans la journée. Il ne sera pas forcément pertinent de définir et de réserver de façon permanente une trajectoire 3D pour chacun de ces sous-flux.

Il est donc proposé, pour le modèle MULTI, de calculer d'abord les sous-flux pour l'ensemble des flux origine-destination, puis de sélectionner ceux contenant au moins un certain nombre de vols (une dizaine par exemple) pour lesquels on calculera des trajectoires 3D non-interférentes. Pour les modèles UNIC et PROX, on sélectionnera plus simplement les flux origine-destination d'une certaine taille (20 avions ou plus, ou 10 avions ou plus), puis on cherchera une trajectoire 3D par flux pour le modèle UNIC, ou plus pour le modèle PROX, en distinguant les sous-flux décollant ou se posant d'un aéroport proche, même si ceux-ci sont de taille modeste.

8.5 Modélisation des trajectoires

8.5.1 Adaptation du modèle aux cas des flux entrants et sortants

Dans le chapitre 7, avec la modélisation simplifiée, chaque avion décollait et se posait dans la zone géographique considérée. Comme nous l'avons vu dans l'analyse des flux et dans la section précédente, ce n'est pas le cas dans la réalité : un certain nombre de flux d'avions entrent ou sortent de l'espace aérien.

Il faut donc tenir compte des niveaux de vol en entrée ou en sortie, en plus des contraintes sur les niveaux de vol successifs (d_j, CFL_j) (on parlera aussi de *clairance* en niveau).

Pour les flux dont l'origine est une balise avec des survols stables en niveau, la trajectoire doit comprendre un segment à altitude constante sur une certaine distance après l'entrée. De même lorsque la sortie est une balise avec survols stabilisés, il faudra que la trajectoire comporte un segment stable d'une certaine longueur avant la balise de sortie. Ces paliers sont imposés afin de rester au plus proche des conditions usuelles de coordination des vols. En effet, en règle générale, la coordination d'un vol entre deux organismes de contrôle se fait à un niveau de vol fixe, l'avion devant être en palier au moment du passage de la frontière.

Dans le cas particulier des flux dont les avions proviennent ou sont à destination d'aéroports hors-zone proches de la frontière de l'espace aérien, des procédures spécifiques de coordination sont définies par les deux centres de contrôle concernés. Typiquement, une telle procédure comprendra une clause spécifiant que les avions en provenance de l'aéroport proche et entrant par une balise spécifiée devront se présenter en montée vers un niveau donné.

Il nous faut donc modifier la modélisation de nos trajectoires en fonction de ces éléments. En reprenant les notations introduites en 7.2.3, une trajectoire est donc déterminée par :

- le choix r de la route,
- le niveau de vol en entrée EFL ,
- la clairance initiale en niveau CFL_e . On aura $CFL_e \neq EFL$ dans le cas d'avions évolutifs en entrée de l'espace (cas des aéroports proches) et $CFL_e = EFL$ lorsqu'ils sont stables en niveau,
- la longueur minimale L_e imposée au palier d'entrée, lorsque les avions sont stables en niveau,
- la séquence de contraintes sur les niveaux de vol $\{(d_j, CFL_j) / j \in [0, p - 1]\}$, où p est le nombre de points de la trajectoire parmi lesquels on peut choisir la valeur de d_j . Pour toute clairance (d_j, CFL_j) , il existe une valeur $k \in [0, p - 1]$ telle que $d_j = k \times \frac{l}{p}$, où l est la longueur de la route choisie.
- le niveau de vol en sortie XFL
- la longueur du palier avant la sortie L_x

A partir de ces éléments, on pourra soit calculer une trajectoire filaire, comme dans la section 7.2, mais en prenant en compte les performances réelles de montée ou de descente d'un type d'avion choisi, soit calculer l'enveloppe des trajectoires filaires de tous les types d'avions d'un flux donné, toujours selon les performances réelles.

8.5.2 Options dans le choix des routes

Dans le chapitre sur le problème simplifié, le choix de la route était limité à trois possibilités illustrées sur la figure 7.2 (section 7.2). La valeur de r (0, 1, ou -1) indiquait un choix parmi les routes parallèles possibles.

Pour le problème réel, une autre possibilité a été ajoutée, qui consiste à choisir parmi les itinéraires standards définis sur le réseau de routes réellement exploité. Pour ce faire, on examine, dans les données archivées, les itinéraires demandés par chaque avion, afin de constituer pour chaque couple origine-destination l'ensemble des itinéraires possibles. Le choix de la route est alors toujours représenté par la variable r , dont la valeur indique alors l'itinéraire choisi dans l'ensemble des routes possibles, que l'on a préalablement indexé.

8.6 Prise en compte des performances réelles des avions

Dorénavant, les phases de montée et de descente des avions seront calculées en fonction des performances des avions décrites dans le fichier de données BADA, utilisé dans les centres de contrôle d'un certain nombre de pays européens. Ce fichier fournit entre autres des informations sur les *vitesse⁴ air* en montée, en descente ou en vol stable, les taux de montée minimal, nominal, et maximal, en pieds par minute, ainsi que le taux de descente, ceci pour la plupart des types d'avions et pour chaque niveau de vol.

⁴ La *vitesse air* est la vitesse de l'avion par rapport à la masse d'air dans laquelle il se déplace.

A partir de ces informations, il est facile de reconstituer le profil vertical d'une trajectoire de montée ou de descente, par vent nul. Comme on cherche à calculer des trajectoires statiques, valables pendant un certain temps, intégrer les données météo de gradients de vent n'a pas grand sens étant donnée la période de validité de telles données.

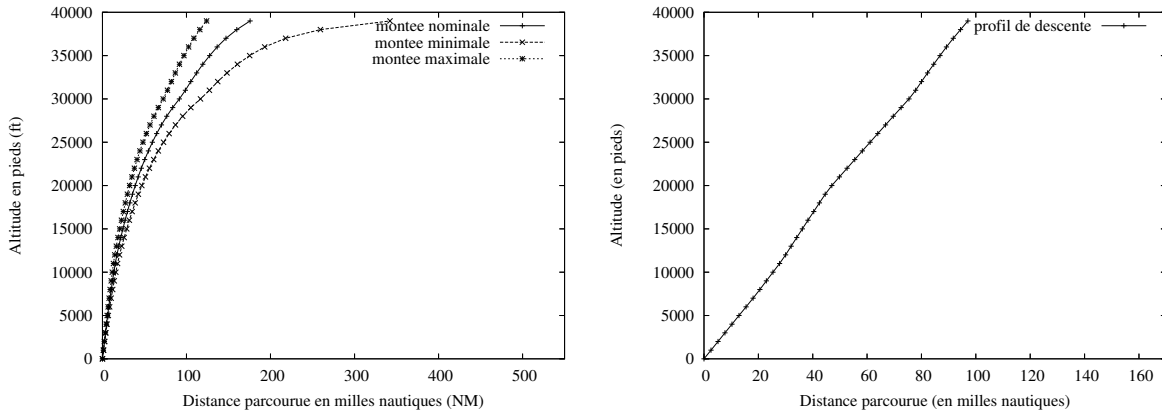


FIG. 8.14 – Profils de montée et de descente pour un Airbus A320.

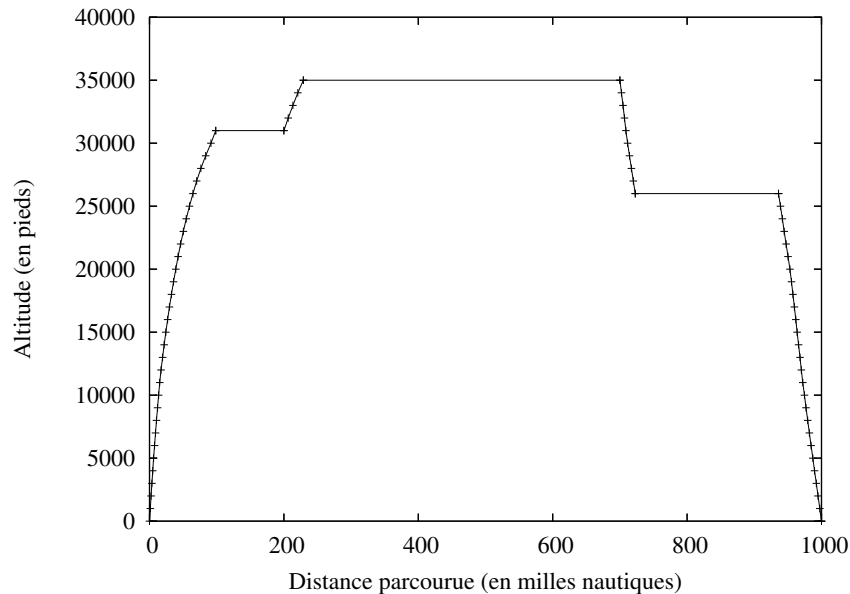


FIG. 8.15 – Exemple de profil de vol, pour un Airbus A320.

A titre d'exemple, la figure 8.14 montre, sur la partie gauche, les profils de montée minimale, nominale, et maximale pour un Airbus A320, avec en abscisse la distance parcourue depuis le décollage, en milles nautiques (NM), et en ordonnée l'altitude en pieds. La partie droite de la même figure montre le profil de descente pour ce même avion, avec en abscisse la distance restant à parcourir avant l'atterrissage. Une séquence de contraintes du type (d_j, CFL_j) (cf modélisation d'une trajectoire en 7.2.3) se traduit alors par un profil nominal du genre de celui de la figure 8.15.

Cependant, il y a, en règle générale, plusieurs types d'avions sur un même flux, avec des performances parfois disparates, comme le montre la figure 8.16, où sont représentées les montées nominales et les profils de descente de tous les types d'avions de la base de données. La figure 8.17 illustre la

disparité des profils calculés pour une séquence d'instructions sur les niveaux de vol, pour plusieurs types d'avions appartenant à un flux (ici : A340, B742, B743, B744, B762, B763, B772, DC10, L101, MD11). On voit à gauche l'ensemble des profils calculés, et à droite les enveloppes inférieure et supérieure de ces profils. Ce sont ces enveloppes qui déterminent la zone d'incertitude verticale associée à la trajectoire calculée pour un flux.

En pratique, afin d'éviter de calculer tous les profils possibles à chaque étape des algorithmes de résolution présentés par la suite, nous calculerons préalablement, pour chaque flux, les enveloppes inférieures et supérieures des profils de montée et de descente, en fonction des types d'avions du flux. Puis, lors des étapes de résolution des algorithmes, nous calculerons seulement deux profils (le profil inférieur et le profil supérieur), en fonction de ces enveloppes.

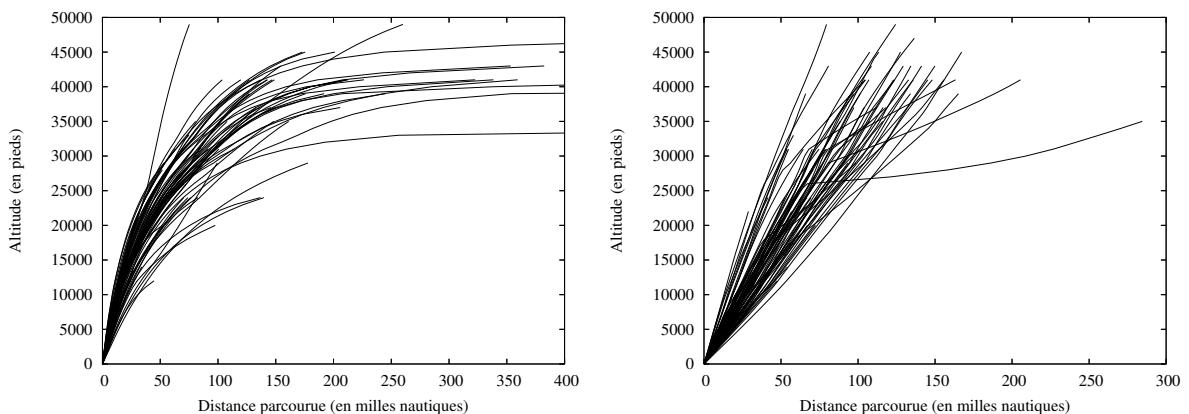


FIG. 8.16 – Profils de montée nominale (à gauche) et de descente (à droite) pour tous les types d'avions de la base de donnée.

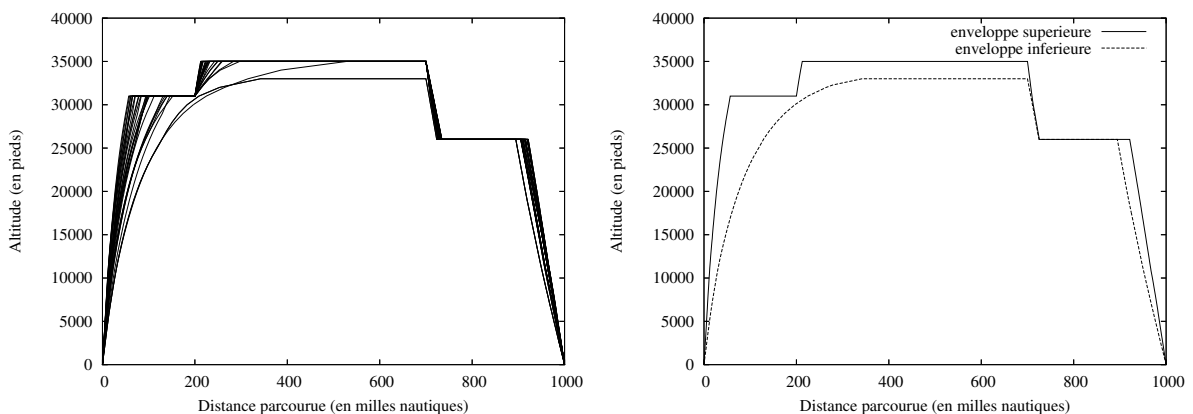


FIG. 8.17 – Exemple de disparité des profils de vol sur un même flux. A gauche : tous les profils ; à droite : l'enveloppe.

Dans la suite de ce chapitre, deux options seront donc possibles pour la modélisation des trajectoires :

- trajectoire filaire, comme dans le chapitre précédent, mais avec les performances de montée et de descente d'un type réel d'avion. On pourra alors soit choisir un type d'avion unique pour tous les flux, ou encore un type d'avion par flux ;

- trajectoire avec une zone d'incertitude dans le plan vertical, cette zone étant l'enveloppe des trajectoires calculées pour tous les types d'avions appartenant au flux considéré.

Dans le deuxième cas, il n'est plus possible de détecter les interférences entre trajectoires simplement en calculant une distance entre segments de droites, comme dans la section 7.3. Il faudra donc utiliser un autre algorithme de détection des interférences, prenant en compte les incertitudes sur les profils verticaux.

8.7 Détection des interférences entre trajectoires

8.7.1 Inconvénients de la détection avec la distance d

Les algorithmes présentés dans le chapitre 7 utilisent une détection d'interférences (cf section 7.3) qui fait appel à la distance d issue du produit scalaire suivant :

$$\begin{pmatrix} x1 \\ y1 \\ z1 \end{pmatrix} \cdot \begin{pmatrix} x2 \\ y2 \\ z2 \end{pmatrix} = \frac{x1 \cdot x2 + y1 \cdot y2}{N_h^2} + \frac{z1 \cdot z2}{N_v^2}$$

où N_h est la norme de séparation horizontale et N_v la norme de séparation verticale.

Si deux segments $s_1 \subset \mathcal{T}_1$ et $s_2 \subset \mathcal{T}_2$ sont distants d'au moins $\sqrt{2}$ avec cette distance, alors on peut être sûr qu'il n'existe aucun couple de points $(p_1, p_2) \in s_1 \times s_2$ tel que p_1 soit en conflit avec p_2 . Rappelons que le conflit est caractérisé par : $d_v(p_1, p_2) < N_v$ et $d_h(p_1, p_2) < N_h$, où d_h est la distance euclidienne dans le plan horizontal, et d_v la valeur absolue de la différence des altitudes.

Toutefois, la réciproque n'est pas vraie : il peut exister des segments pour lesquels

$$\forall (p_1, p_2) \in s_1 \times s_2 \quad d_v(p_1, p_2) > N_v \quad \text{et} \quad d_h(p_1, p_2) < N_h$$

mais qui sont pourtant distants de moins de $\sqrt{2}$ avec la distance d . Pour s'en convaincre, le lecteur pourra se reporter à la figure 7.4 et au commentaire associé, dans la section 7.3.

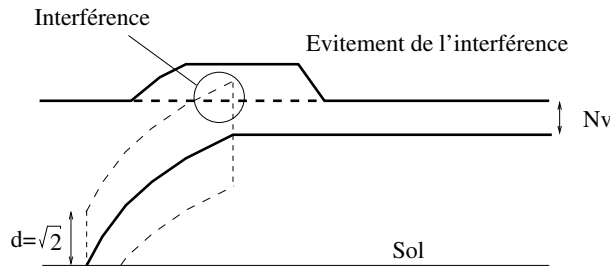
En utilisant la distance d , on peut donc détecter des interférences, alors qu'aucun couple de points n'est en conflit : par exemple, il n'y a aucun conflit entre deux segments stables séparés verticalement de N_v , et pourtant leur distance en utilisant d est inférieure à $\sqrt{2}$. Afin d'être plus précis dans la détection et afin d'éviter le problème évoqué ci-dessus, la distance d n'était utilisée dans la détection d'interférences que lorsqu'au moins l'un des deux segments était évolutif. La détection entre segments stables faisait appel aux distances d_h et d_v .

Malgré tout, il peut subsister des fausses détections, comme dans l'exemple illustré en figure 8.18, où une trajectoire en montée interfère avec une trajectoire stable. Une telle fausse détection peut conduire à dévier l'une des deux trajectoires lors de la résolution, inutilement puisqu'elles n'étaient pas vraiment conflictuelles. Des fausses détections analogues peuvent également se produire dans le plan horizontal.

8.7.2 Détection par intersection de tubes

Pour éviter ce type de problème, définissons autour de chaque segment un *tube* de section carrée, de largeur N_h et de hauteur N_v , le segment étant situé au milieu. Deux trajectoires \mathcal{T}_1 et \mathcal{T}_2 sont alors interférentes s'il existe $s_1 \subset \mathcal{T}_1$ et $s_2 \subset \mathcal{T}_2$ pour lesquels les tubes associés soient sécants.

Cette définition est facilement extensible à la détection d'interférences entre zones d'incertitude. Il suffit de définir les tubes en ajoutant à chaque zone d'incertitude une marge de $\frac{N_v}{2}$ en haut et en bas, et latéralement une bande de largeur $\frac{N_h}{2}$.

FIG. 8.18 – Exemple de problème dans la détection d'interférence avec la distance d .

L'algorithme déterminant si deux tubes 3D sont sécants est un peu moins simple que celui calculant la distance d . Pour déterminer si deux tubes Tu_1 et Tu_2 sont sécants, on regarde si :

- une arête d'un des tubes traverse une des faces de l'autre tube,
- ou si l'un des deux tubes englobe totalement l'autre.

Pour savoir si une arête traverse une face, on donne d'abord une orientation à cette face, vers l'extérieur du tube. On a alors trois cas possibles :

- soit les deux points sont du côté extérieur de la face,
- ou ils sont tous les deux du côté intérieur,
- ou encore il y en a un de chaque côté.

Dans les deux premiers cas, l'arête ne peut pas traverser la face. Dans le troisième, lorsqu'il y a un point de chaque côté, l'arête (p, q) traverse la face (a, b, c, d) si et seulement si elle est à l'intérieur d'un tube délimité par les quatre plans parallèles (p, q) et passant par les côtés (a, b) , (b, c) , (c, d) et (d, a) de la face, comme l'illustre la figure 8.19. Pour le déterminer, on regardera de quel côté de chacun de ces plans se trouve l'arête (p, q) . Considérons par exemple le plan passant par (a, b) et parallèle à (p, q) . Si le point c et l'arête (p, q) ne sont pas d'un même côté de ce plan, alors (p, q) ne peut pas traverser (a, b, c, d) . S'ils sont d'un même côté, on refait le même test pour le plan passant par (b, c) , avec le point d , et ainsi de suite jusqu'à ce que l'on ait testé tous les plans, auquel cas il y a effectivement intersection.

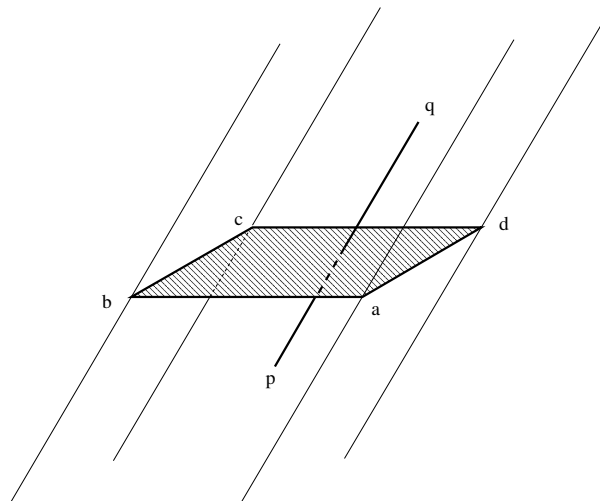


FIG. 8.19 – Intersection d'une arête et d'une face.

Si l'on a parcouru toutes les arêtes de Tu_1 et qu'aucune ne traverse une des faces du deuxième

tube Tu_2 , il reste trois possibilités :

- Tu_1 est totalement à l'intérieur de Tu_2 ,
- ou Tu_2 est à l'intérieur de Tu_1 ,
- ou alors Tu_1 et Tu_2 ne sont pas sécants.

La première possibilité a en fait été testée pendant le parcours des arêtes : chacune des arêtes de Tu_1 doit être du côté intérieur de chacune des faces orientées de Tu_2 . Pour lever le doute entre la deuxième et la troisième possibilité, on regarde si les arêtes de Tu_2 sont du côté intérieur des faces orientées de Tu_1 .

8.7.3 Interférences autour des aéroports ou dans les phases évolutives

La détection avec la distance d comme celle avec l'intersection de tubes n'ont jusqu'à présent pas pris en compte le cas de deux flux dont l'une des extrémités est un même aéroport. Dans ce cas précis, et quel que soit le type de détection choisi, deux segments ayant en commun un point (l'origine ou la destination) seront toujours interférents. Or, ce type d'interférence ne peut pas être réglé par des déviations de trajectoires, mais uniquement en séparant temporellement les arrivées et les départs de l'aéroport concerné. Ce séquençement est fait par l'organisme de contrôle gérant l'approche et les départs et arrivées de piste dans la TMA concernée.

En conséquence, les portions de trajectoires comprises dans un certain rayon autour des aéroports ne seront pas pris en compte dans la détection des interférences. La valeur de ce rayon est un paramètre choisi de façon à pouvoir au minimum séparer un flux aller d'un flux retour (entre deux aéroports quelconques) avec une déviation latérale. La valeur adoptée dans les exemples de résultats est de 15 milles nautiques.

Notons que lorsque deux flux ont une extrémité sur une même balise d'entrée ou de sortie, il est toujours possible et même souhaitable de séparer ces flux en les décalant en niveau de vol. C'est pourquoi le rayon dont nous venons de parler n'est défini qu'autour des aéroports et pas des balises frontalières.

La définition de ces zones circulaires sans détection ne suffit cependant pas à régler un certain nombre de problèmes. Par exemple, on peut souhaiter définir plusieurs trajectoires pour un couple (*origine, destination*) donné, en fonction par exemple des performances des avions sur ce trajet. Supposons que l'on veuille définir une trajectoire pour les avions évoluant uniquement dans l'espace inférieur (RFL < 19500 pieds), et une autre pour ceux demandant un RFL dans l'espace supérieur. Dans ce cas, il serait intéressant de considérer que les phases évolutives (montée ou descente) n'interfèrent pas : les deux flux utiliseraient des tubes communs pour la montée initiale et la descente finale, les phases en croisière et les autres évolutions éventuelles devant être par contre séparées. Ceci éviterait d'avoir à décaler latéralement un des deux flux, un tel décalage devenant de toute façon impossible si l'on veut définir plus de trois trajectoires pour un couple (*origine, destination*)

Un autre exemple est celui d'un faisceau de flux ayant une même origine et partant dans des directions très proches, comme sur la figure 8.20. On voit qu'il y a dans ce cas peu d'espoir d'arriver à séparer ces trajectoires aux alentours de la TMA avec les déviations latérales que nous nous autorisons. En effet, à partir d'un certain nombre de flux, les trajectoires décalées vont également se retrouver trop proches les unes des autres. Les solutions à ce problème sont multiples. Il est possible d'une part d'agrandir le rayon de la TMA, en espérant englober totalement les phases de montée ou de descente, de façon à séparer verticalement les multiples flux partant dans des directions proches. Cela pourrait être assez logique dans le fond : les TMAs gérant beaucoup de trafic devraient pouvoir se réserver plus d'espace, et le rayon de la TMA pourrait être fixé en fonction du nombre d'avions atterrissant ou décollant de l'aéroport. L'inconvénient est que l'on risque soit de devoir prendre des rayons de TMA

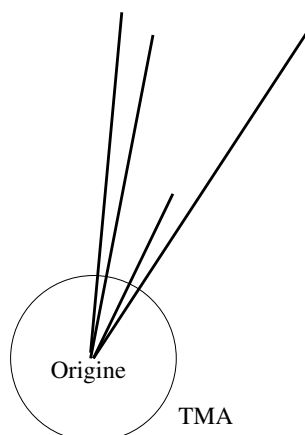


FIG. 8.20 – Exemple de flux partant dans des directions proches.

trop importants pour pouvoir englober totalement les montées initiales (cf figure 8.16), soit d'obtenir des paliers intermédiaires lors de la recherche de solution.

Une autre solution, plus simple et finalement équivalente pour les flux concernés, consiste à considérer que les montées initiales provenant d'un même aéroport n'interfèrent pas entre elles, et de même pour les descentes finales à destination d'un même aéroport. On considère en fait que la séparation des avions doit se faire par un séquençage temporel, et est à la charge des systèmes gérant l'approche de l'aéroport concerné. Par contre, on continuera de détecter les interférences entre montée initiale et descente finale, de façon à séparer les arrivées et les départs à l'extérieur des zones TMA.

8.7.4 Résumé des différents modes de détection

Pour résumer, nous aurons donc trois modes possibles pour la détection :

- interférences entre trajectoires filaires, avec la distance d ,
- interférences entre trajectoires filaires, par l'intersection de tubes de largeur N_h et de hauteur N_v définis autour de chaque segment de chaque trajectoire,
- interférences entre zones d'incertitude, par l'intersection de tubes définis autour de ces zones. Les tubes sont de largeur N_h et de hauteur égale en chaque point à celle de la zone d'incertitude, plus une marge de $\frac{N_v}{2}$ en haut et en bas.

Par ailleurs, une zone tampon (de l'ordre de 15 milles nautiques, paramétrable) est définie autour de chaque aéroport, où la détection d'interférence est inactive.

Enfin, nous nous réservons la possibilité d'inactiver la détection d'interférences entre les montées initiales provenant d'un même aéroport, considérant que le séquençage temporel des avions par les systèmes gérant les départs doit suffire à assurer la séparation sur un tel segment de vol. De même pour les interférences entre les descentes finales à destination d'un même aéroport, leur détection pourra être désactivée.

8.8 Adaptation des algorithmes au problème réel

Nous venons de voir dans les sections précédentes comment prendre en compte les performances réelles des avions, la disparité des types d'avions au sein d'un même flux, et comment améliorer la détection des interférences. Ces points ne modifient pas à proprement parler la mécanique des

algorithmes, mais seulement des fonctions auxquelles ils font appel. Par contre, le changement du modèle de trajectoire, avec l'ajout de niveaux de vol en entrée et en sortie de l'espace aérien, nous conduit à revoir les mécanismes adoptés en 7.6 et 7.7. Nous ne reviendrons pas sur le principe ni le détail des algorithmes, mais seulement sur les modifications apportées.

8.8.1 Adaptation de l' A^*

Pour ce qui est de l' A^* , les fonctions de coût et d'heuristique ne sont pas modifiées. Seul le calcul des noeuds fils change, selon que l'on a une entrée ou une sortie en palier, au lieu d'un décollage ou d'un atterrissage.

Représentation des noeuds

Rappelons que dans le modèle simplifié (section 7.6.2) un noeud était représenté par un quadruplet $(r, cfls, j, seg)$, où :

- r est le choix de la route,
- $cfls = \{(d_j, CFL_j) / j \in [0, p - 1]\}$ est une séquence de contraintes sur les niveaux de vol (on parlera aussi de *clairances* en niveau). Une *clairance* (d_j, CFL_j) signifie : à la distance d_j , débiter une évolution verticale vers le niveau de vol CFL_j . Une telle clairance se traduit donc par un segment de vol évolutif, puis un segment de vol stabilisé en croisière à CFL_j ,
- j est l'indice de l'étape en cours,
- seg vaut soit *Evolutif* soit *Croisière* selon que l'on soit sur le segment évolutif ou en croisière de l'étape en cours.

Le couple $(r, cfls)$ suffisait à déterminer totalement la trajectoire dans l'espace, connaissant les performances de montée et de descente, et avec comme contrainte implicite le fait que les avions devaient se poser sur leur aéroport de destination.

Comme nous enrichissons notre modèle en autorisant des entrées ou des sorties de l'espace aérien, il nous faut modifier la représentation des noeuds parcourus par l' A^* . Dorénavant une trajectoire sera déterminée par un n-uplet $(r, EFL, CFL_e, L_e, cfls, XFL, L_x)$ comme décrit en 8.5, et un noeud sera un n-uplet :

$$(r, EFL, CFL_e, L_e, cfls, XFL, L_x, etape, seg)$$

où *etape* identifie l'étape en cours : *Initiale*, *Clairance j* , ou *Finale*. Typiquement, une étape *Initiale* pourra être une montée partant de l'aéroport de départ ($EFL = 0$), vers un niveau cible initial (CFL_e), suivie d'une stabilisation à ce niveau. Dans le cas d'une entrée stable en niveau, l'étape *Initiale* aura une phase évolutive réduite à l'ensemble nul, suivie d'un segment de vol stabilisé de longueur au moins L_e . Une étape *Finale* sera soit une phase évolutive pour rejoindre un palier de longueur minimale L_x dans le cas d'une sortie en palier, soit une descente finale vers un aéroport, suivie d'une phase stabilisée réduite à l'ensemble nul. Le couple $(etape, seg)$ identifie donc précisément une phase de vol (évolutif, ou en palier) correspondant à une clairance en entrée ou en sortie, ou à une clairance à l'intérieur de l'espace aérien.

Revenons maintenant au calcul des noeuds fils dans le parcours de l'arbre par l' A^* . Pour les flux débutant et se terminant sur un aéroport, le calcul est analogue à celui décrit en 7.6.2. Par contre, il est modifié lorsqu'il y a une entrée ou une sortie de l'espace aérien, afin de prendre en compte le choix parmi plusieurs possibilités pour les niveaux de vol en entrée ou en sortie.

Cas d'une entrée en palier

Dans le cas d'une entrée en palier, la racine de l'arbre des possibilités pour le calcul d'une trajectoire aura toujours trois fils, comme en 7.6.2, chacun d'entre eux correspondant au choix d'une route. Toutefois, au lieu de déterminer pour chaque fils la plus longue montée initiale sans interférence, nous chercherons le plus long palier d'entrée possible sans interférence, au niveau de vol le plus proche du niveau d'entrée demandé tout en restant au dessus de FL_{min} . La longueur d'un palier d'entrée devra être au minimum de L_e . S'il n'existe aucun palier possible répondant à ces contraintes et n'interférant avec aucune trajectoire, le fils calculé est rejeté.

Cas d'une sortie en palier

Le cas d'une sortie en palier est pris en compte dans le calcul des fils des noeuds représentant une étape en croisière. Rappelons que pour une trajectoire terminant sur un aéroport, dès qu'une étape en palier se termine par un début de descente finale et si celle-ci est libre de toute interférence, alors la trajectoire est complète et l'algorithme se termine. C'est la courbe de descente finale vers l'altitude 0 qui détermine la fin de trajectoire. Par contre, dans le cas d'une sortie de l'espace aérien, on a la possibilité de choisir entre plusieurs niveaux de sortie si celui initialement prévu ne convient pas.

Donc pour toute étape en croisière représentée par

$$(r, EFL, L_e, \{(d_0, CFL_0); \dots; (d_j, CFL_j); (d_{j+1}, CFL_j)\}, XFL, L_x, j, Croisiere)$$

on calculera deux premiers fils (passage à l'étape suivante, ou palier alternatif plus court) comme en 7.6.2, et pour le troisième on cherchera le niveau de sortie XFL' le plus proche possible de XFL tel que les étapes $(j, Croisiere)$, $Finale$, $Evolutif$ et $(Finale, Croisiere)$ d'une nouvelle trajectoire

$$(r, EFL, L_e, \{(d_0, CFL_0); \dots; (d_j, CFL_j); (d'_{j+1}, CFL_j)\}, XFL', L_x)$$

soient libres de toute interférence. La nouvelle distance d'_{j+1} est égale à d_{j+1} lorsqu'il est possible de faire un palier de sortie de longueur au moins égale à L_x au niveau XFL' en partant de d_{j+1} . Sinon, d'_{j+1} marque le début de l'évolution vers le segment en palier de longueur minimale L_x . Le fils que l'on rajoutera dans ce cas dans la file d'attente à priorités de l' A^* correspond à l'étape évolutive finale de la nouvelle trajectoire :

$$(r, EFL, L_e, \{(d_0, CFL_0); \dots; (d_j, CFL_j); (d'_{j+1}, CFL_j)\}, XFL', L_x, Finale, Evolutif)$$

Cas des entrées en évolution

Pour les modèles d'allocation permettant de définir plusieurs trajectoires par flux, on s'autorise des entrées ou des sorties via des balises frontalières, en configuration de montée ou de descente.

Prenons le cas d'une entrée en montée. Pour chaque noeud fils de la racine de l'arbre (correspondant chacun à une route possible), on cherchera dans un premier temps le segment en montée, sans interférence, vers un niveau le plus proche possible du niveau de croisière préférentiel. En cas d'interférence, on essaie une évolution vers un niveau au dessous, selon le même mécanisme que pour les montées provenant d'un aéroport, sauf si ce niveau est inférieur ou égal au niveau d'entrée préférentiel (EFL initial). Si c'est le cas, on essaie alors successivement les paliers en entrée en dessous du niveau préférentiel d'entrée, selon le mécanisme décrit précédemment pour les entrées en palier, jusqu'à ce qu'on trouve un palier en entrée d'une longueur minimum L_e sans interférences ou jusqu'à ce que

l'on atteigne un niveau minimum autorisé pour ces paliers. Ce qui signifie dans ce dernier cas qu'on n'a pas trouvé de solution d'entrée.

Il est également possible de rencontrer des entrées en descente, lorsque les avions n'ont pas le temps de faire une phase en palier avant de se poser. Dans ce cas, on teste d'abord le segment de descente finale, et s'il interfère avec d'autres trajectoires, on essaie des entrées en palier à des niveaux inférieur au niveau préférentiel d'entrée.

Cas des sorties en évolution

Une sortie en descente correspond nécessairement à une descente finale vers un aéroport étranger, sinon, on impose un palier d'une certaine longueur avant la sortie. Le cas des sorties en descente est en fait géré, pendant le parcours des états par l' A^* , par la mise à jour des valeurs de XFL et L_x lors des étapes en palier. Lorsque le palier courant est à un niveau inférieur au niveau de sortie préférentiel, on force la valeur du XFL à celle du palier courant, et la valeur de L_x à la longueur minimale d'un palier en sortie. Dès que le palier courant repasse au dessus du niveau préférentiel de sortie, L_x repasse à zéro, et le XFL à la valeur du niveau préférentiel.

Une sortie en montée est nécessairement un cas dégradé de montée initiale où il n'est pas possible de se stabiliser au niveau préférentiel de croisière avant la sortie de l'espace. Dans ce cas, on teste d'abord la trajectoire initiale, puis, si elle interfère avec d'autres, on essaie successivement les paliers en sortie en dessous du niveau préférentiel de sortie.

Choix de l'ordre des flux

Les flux sont parcourus en considérant en priorité ceux demandant les niveaux les plus hauts, et ceux contenant le plus de vols. Plus précisément, ils sont triés selon l'ordre décroissant lexicographique des niveaux (EFL , XFL , RFL) et du nombre de vols.

Ce choix est motivé principalement par la configuration des flux au-dessus de la France. Quelques essais en triant les flux seulement par ordre décroissant de taille ont montré que l' A^* avait plus de difficultés à trouver des solutions, notamment pour les flux entrant à un niveau élevé et se terminant dans la TMA parisienne, qu'avec l'ordre proposé.

8.8.2 Adaptation de l'algorithme génétique

L'algorithme génétique est assez peu modifié par rapport à la section 7.7. Simplement, la population initiale est générée un peu différemment en choisissant aléatoirement, en plus des paliers intermédiaires, les valeurs des variables EFL et CFL_e pour les flux entrant dans l'espace aérien via une balise frontalière, et en choisissant aléatoirement le niveau XFL pour les flux sortant via une balise frontalière.

Modification des opérateurs

L'opérateur de croisement est également modifié pour tenir compte de ces éventuels niveaux d'entrée et de sortie, tout en gardant le principe exposé dans la section 7.7.

L'opérateur de mutation utilise l' A^* décrit dans la section précédente. Pour cet opérateur, on peut choisir parmi plusieurs options plus ou moins élitistes. On peut choisir par exemple de muter préférentiellement une trajectoire dont la *fitness locale* n'est pas bonne. Pour un élément de population qui n'est pas une solution (il reste des interférences entre trajectoires), cela veut dire que l'on s'attaque

en priorité aux trajectoires qui ont le plus de segments conflictuels. Lorsqu'il ne reste plus d'interférences, on cherchera alors à muter préférentiellement les trajectoires les plus coûteuses. Cette stratégie élitiste risque toutefois de focaliser la mutation sur certaines trajectoires, au détriment des autres.

Cette version de l'algorithme génétique a tout d'abord été testée sur les flux de 20 avions ou plus au-dessus de la France, le 21 juin 2002, avec le modèle UNIC (une seule trajectoire par flux) et le mode de détection basé sur la distance d , avec un seul type d'avion (A320). Deux tests correspondant à deux valeurs différentes de la racine du générateur de nombres aléatoire ont été réalisés, avec une population de 400 éléments évoluant sur 250 générations, une probabilité de croisement de 0.6 et de mutation de 0.1. Bien que satisfaisants pour ce qui est du critère d'adaptation (1.283677 et 1.279077, contre 1.238564 pour l' A^* sur le même problème), les temps de calcul étaient particulièrement importants : 32 heures 38 minutes pour le premier, et 22 heures 55 minutes pour le second, sur un CPU Xeon 2.8 GHz.

Après investigation, il apparut que ces temps de calcul rédhibitoires étaient dus à l'algorithme A^* utilisé dans l'opérateur de mutation. Dans cet opérateur, on choisit une trajectoire choisie au hasard (ou selon la *fitness* locale), que l'on remplace par une nouvelle trajectoire calculée par l' A^* . Celui-ci doit donc trouver un chemin dans l'espace encombré par le reste du trafic, c'est-à-dire l'ensemble des autres trajectoires. Or, celles-ci sont générées aléatoirement au départ de l'algorithme. En conséquence, l' A^* rencontre de plus grandes difficultés pour trouver une solution dans ce contexte (ou pour déterminer qu'il n'y en a pas) que lorsqu'il est appliqué séquentiellement à des flux dans un ordre choisi (les niveaux les plus hauts d'abord). Plus spécifiquement, les flux les plus pénalisants sont ceux au départ ou à destination de la TMA Paris ayant des niveaux de croisière préférentiels hauts, et qui croisent par conséquent un grand nombre des flux passant au-dessus de la France. Lorsqu'on utilise l' A^* seul, les trajectoires associées à ces flux sont calculées parmi les premières, dans un espace encore peu encombré.

Pour contourner cette difficulté, d'autres opérateurs de mutation ont été essayés, dans lequel l' A^* n'est pas utilisé, ou bien remplacé par un algorithme d'optimisation locale cherchant à améliorer le critère d'évaluation, en augmentant itérativement l'altitude et la longueur des paliers. Ces tentatives ont donné de piètres résultats, l'algorithme ayant des difficultés à trouver une solution.

Finalement, l'opérateur de mutation présentant le meilleur compromis en termes de résultats et de temps de calcul est le suivant. On choisit une trajectoire à muter :

- aléatoirement lorsqu'il ne reste aucune interférence (*fitness* supérieure à 1),
- et sinon, avec une chance sur deux, soit aléatoirement, soit d'une manière un peu plus élitiste en tirant une trajectoire au hasard un nombre de fois égal à la moitié du nombre de trajectoires et en prenant celle interférant le plus avec le reste du trafic.

Une fois choisie la trajectoire, on applique (selon un choix aléatoire) une des trois stratégies suivantes de mutation :

- bruite aléatoirement la trajectoire,
- s'il ne reste aucune interférence, utiliser l'algorithme de recherche locale évoqué précédemment pour améliorer la trajectoire (diminuer son coût),
- ou appliquer l'algorithme A^* , en ne considérant que les trajectoires interférant avec la trajectoire choisie, si l'on est en début de recherche (à moins du quart du nombre maximum de générations par exemple). Pour les autres générations, l' A^* est utilisé normalement, en considérant l'ensemble des autres trajectoires.

Nécessité d'un découpage en sous-problèmes

Comme nous allons le voir un peu plus loin dans la présentation des résultats, l'utilisation de l'algorithme génétique pour des configurations de test comportant environ 70 flux (flux de 20 avions ou plus par exemple) prend déjà plusieurs heures de calcul. Ce temps de calcul augmente encore considérablement pour peu que l'on cherche à séparer un nombre plus important de trajectoires (de l'ordre de 160 trajectoires pour les flux de 10 avions ou plus).

Afin de pouvoir traiter plus de trajectoires, il est naturel d'examiner d'abord si elles sont toutes susceptibles d'interférer, et s'il faut réellement les considérer toutes ensemble en un seul et même problème, ou bien s'il est possible de s'attaquer successivement ou en parallèle à plusieurs sous-problèmes indépendants. Une possibilité pour cela est d'examiner au préalable l'intersection des enveloppes de toutes les trajectoires possibles, de construire le graphe correspondant et d'en déterminer les composantes connexes (les sous-graphes indépendants les uns des autres).

Une autre solution évitant de calculer les enveloppes des trajectoires consiste à calculer dynamiquement ces composantes connexes, sur la base des interférences entre trajectoires. Le processus est alors le suivant :

1. calculer les sous-ensembles de trajectoires interférant entre elles,
2. appliquer l'algorithme génétique, successivement ou en parallèle, à chacun des sous-problèmes,
3. examiner si le résultat comporte de nouvelles interférences entre trajectoires appartenant à deux sous-problèmes différents,
4. si c'est le cas, fusionner les sous-problèmes concernés et revenir à l'étape 2, sinon on a trouvé une solution sans interférences.

Malheureusement, lorsqu'on applique ce procédé au problème des flux de plus de 20 avions au dessus de la France, on finit par obtenir un seul sous-problème, comprenant une soixantaine de flux. C'est-à-dire qu'on est ramené au problème global.

Quelques essais sur les données Europe, pour lesquelles la configuration des flux est différente, semblent plus prometteurs. Les modifications nécessaires pour une utilisation efficace de ce découpage en sous-problèmes sont en cours de développement.

8.9 Commentaires préalables à la présentation des résultats

8.9.1 La démarche adoptée

Chaque algorithme est testé avec différentes configurations de paramètres. Etant donné leur nombre, il n'est pas possible de passer en revue toutes les configurations possibles, et seuls sont présentés les résultats correspondant à un jeu de configurations types, pour lesquels un certain nombre de paramètres sont fixés (normes de séparation, niveau de coupure entre espaces inférieur et supérieur, pas de calcul des trajectoires, niveaux de vol minimum et maximum). Les deux algorithmes sont testés pour une journée de trafic donnée avec les différents modes de détection d'interférences possibles.

Les résultats bruts de chacun des algorithmes sont évalués en examinant le nombre d'interférences restantes, s'il en reste, et le coût des trajectoires (tel que défini en 7.2.4) Le critère idéal permettant de rendre compte simultanément de ces deux aspects est la *fitness* \mathcal{F} définie pour l'algorithme génétique en 7.7.3, que l'on peut calculer aussi pour les solutions fournies par l' A^* (en allouant éventuellement la trajectoire la plus directe lorsque celui-ci n'a pas trouvé de solution). Rappelons que cette *fitness* est strictement inférieure à 1 lorsqu'il reste des interférences, et supérieure à 1 sinon, et d'autant plus grande dans ce cas que le coût de la trajectoire est faible.

Par ailleurs, comme on cherche aussi à valider le concept même d'allocation de trajectoires 3D, on comparera au moyen d'un simulateur de trafic (CATS/OPAS) chaque solution trouvée, avec des trajectoires 3D imposées à un certain nombre d'avions, à une situation de référence où les trajectoires sont calculées sur la base des routes et niveaux demandés dans les plans de vol. Le nombre et la nature des conflits détectés par le simulateur seront les critères d'évaluation. Ces simulations nous permettront également de valider la détection d'interférences, en vérifiant qu'il ne reste plus dans les solutions trouvées de conflits entre avions appartenant à des flux différents (sauf éventuellement dans les phases évolutives en provenance ou à destination d'un même aéroport).

8.9.2 Choix des paramètres

L'algorithme génétique et l' A^* sont testés sur la journée archivée de trafic du 21 juin 2002. Le niveau FL145 (soit 14500 pieds) est choisi comme seul et unique niveau de coupure : seul le trafic demandant un niveau de croisière au-dessus de ce niveau sera pris en compte. Le niveau de coupure usuel entre l'espace aérien dit *supérieur* et l'espace aérien *inférieur* est le FL195. Le choix du FL145 augmente les possibilités de solutions, et par conséquent la taille de l'espace d'états à explorer. C'est donc un choix volontaire, permettant d'évaluer le temps de calcul des algorithmes dans un cas plus défavorable.

Les paramètres des programmes sont fixés comme suit :

- Le pas δ auquel on s'autorise des changements de niveau de vol est fixé à un dixième de la longueur de la route choisie.
- Le nombre maximum de paliers successifs de croisière que l'on s'autorise est fixé à 3.
- Le niveau de croisière minimum est le niveau de coupure FL145 (14500 pieds), sauf éventuellement pour le premier ou le dernier palier. En pratique, ce sera donc le FL150 sachant que les avions ne sont autorisés à voler que sur des niveaux multiples de dix, ce qui assure la séparation théorique des 1000 pieds.
- Le premier palier pourra éventuellement être plus bas (minimum FL60), sur une longueur maximale de $2 \times \delta$.
- Le dernier palier pourra également être entre le FL60 et le FL145, sur une longueur maximale de $2 \times \delta$.
- La norme de séparation horizontale est fixée à 6 milles nautiques pour la recherche de trajectoires par l' A^* ou l'algorithme génétique, et à 5 milles nautiques pour la détection des conflits par le simulateur de trafic. Cette marge supplémentaire est introduite pour tenir compte des biais dus à la projection stéréographique : la recherche des trajectoires 3D sans interférences se fait dans un espace euclidien avec des coordonnées x et y dans le plan de projection stéréographique, alors que le simulateur fait voler les avions en calculant incrémentalement des positions calculées en latitude et longitude.
- La norme de séparation verticale est fixée à 999 pieds pour la recherche de trajectoires, et à 800 pieds pour les simulations. La norme théorique est de 1000 pieds, avec des variations admises d'une centaine de pieds dans la tenue en palier du vol, ce qui explique la valeur choisie pour le simulateur.

8.9.3 Les configurations de référence

Notons que pour les simulations, il y aura pour chaque journée deux configurations de référence en fonction du mode de détection choisi. En effet, pour les modes **DIST-A320** et **ITUBES-A320**, tous les avions auxquels on a alloué une trajectoire 3D sont censés être du type A320. On fera donc pour

ces modes l'hypothèse d'une flotte aérienne totalement homogène, constituée exclusivement d'Airbus A320. Par contre, pour le mode **IZONES**, tous les types d'avions sont pris en compte.

Selon le cas, les configurations de référence seront donc constituées soit des plans de vol archivés de la journée choisie, tels quels, soit des plans de vol archivés dans lesquels on remplace les types d'avions par le type A320.

8.9.4 Description des configurations de test

Plusieurs configurations de flux sont extraites de la journée de trafic, en fonction du modèle d'allocation choisi. Pour chaque configuration, on teste trois modes de détections d'interférences entre trajectoires, décrits en 8.7.4 :

DIST-A320 : détection basée sur un critère de distance d (cf section 7.3) entre trajectoires filaires.

Dans ce cas, on considérera qu'à chaque flux est associé une trajectoire filaire, calculée selon les performances d'un type d'avion unique. Le type d'avion retenu est l'Airbus A320.

ITUBES-A320 : détection d'intersections de tubes définis autour des trajectoires filaires, calculées également selon les performances de l'Airbus A320.

IZONES : détection d'intersections de zones d'incertitudes définies pour chaque flux à partir de l'enveloppe de toutes les trajectoires possibles des avions du flux. Tous les types d'avions sont pris en compte.

En plus du type de détection, il existe deux possibilités pour le choix des routes (cf 8.5.2) :

PAR : on choisit parmi la route directe, une parallèle à gauche, ou une parallèle à droite,

STD : on choisit parmi l'ensemble des itinéraires standards archivés dans les plans de vols.

Les configurations de test sont nommées en fonction de la zone géographique, de la journée choisie, du modèle de flux (cf section 8.4) et des différents choix que nous venons d'exposer. Par exemple **FR-JUNE21-UNIC-A*-20-DIST-A320-PAR** sera le résultat de l' A^* pour les flux de plus de 20 avions en France, au-dessus du FL145 le 21 juin 2002, en modélisant une trajectoire par flux (modèle UNIC), en détectant les interférences avec la distance d , et en choisissant chaque route parmi trois routes parallèles. Autre exemple : on nommera **EU-JUNE21-MULTI-GA-10-IZONES-STD** le résultat de l'algorithme génétique pour la même journée, en Europe, mais pour les sous-flux calculés selon la méthode de classification évoquée en 8.4, de plus de 10 avions, avec une détection par intersection de zones d'incertitudes, et en choisissant chaque route parmi les itinéraires standard.

8.10 Résultats avec les données France, au-dessus du FL145

Au vu de la variété des configurations de test et afin de ne pas trop alourdir la présentation, les résultats pour les flux dans l'espace aérien français sont présentés en détail pour la première configuration de test (flux de 20 avions ou plus, résolution avec l' A^* , une seule trajectoire par flux), et plus succinctement en fonction des besoins pour les configurations suivantes.

8.10.1 FR-JUNE21-UNIC-A*-20-PAR

Le nombre de flux de 20 avions ou plus demandant un niveau de croisière au-dessus du FL145 (14500 pieds) est de 77, dont 6 sont rejetés avant lancement de l' A^* , soit parce que la distance parcourue est inférieure à 20 milles nautiques, soit parce qu'aucun avion du flux n'a matériellement le

Mode de détection	DIST-A320	ITUBES-A320	IZONES
\mathcal{F}	1.238564	1.272030	0.083336
Nb. échecs	0	0	1
Temps (en secondes)	134.17	73.51	106.95
Coût	296.64071	260.02877	(205.21511)
Nb. FL esp. sup.	19	19	20
Nb. FL esp. inf.	0	0	1
FL min prem.	180	180	140
FL min der.	180	180	180
FL min autres	180	250	270
FL max	390	390	400
Allgt. route	0.67 %	0.60 %	0.14
% trafic traité	39.60 %	39.60 %	39.30 %
Nb. conflits REF FL>60	2711	2711	3077
Total conflits <i>bf OPT FL>60</i>	2674	2565	2721
Même flux FL>60	360	339	396
Gain FL>60	14.64 %	17.89 %	24.44 %
Nb. conflits REF FL>145	1750	1750	2042
Total conflits OPT FL>145	1870	1745	1878
Même flux FL>145	329	308	358
Gain FL>145	11.94 %	17.89 %	25.56 %
Nb. conflits REF FL>195	1389	1389	1582
Total conflits OPT FL>195	1446	1371	1476
Même flux FL>195	321	300	342
Gain FL>195	19.00 %	22.89 %	28.32 %

TAB. 8.9 – Résultats de l'A* pour la configuration de test **FR-JUNE21-UNIC-A*-20-PAR**.

DIST-A320					
FL>60	Total	Même flux	Flux \neq	Mixtes	Autres
Nb conflits	2674	360	121	722	1471
% conflits	100.00 %	13.46 %	4.53 %	27.00 %	55.01 %
FL>145	Total	Même flux	Flux \neq	Mixtes	Autres
Nb conflits	1870	329	85	587	869
% conflits	100.00 %	17.59 %	4.55 %	31.39 %	46.47 %
FL>195	Total	Même flux	Flux \neq	Mixtes	Autres
Nb conflits	1446	321	18	543	564
% conflits	100.00 %	22.20 %	1.24 %	37.55 %	39.00 %

TAB. 8.10 – Répartition des conflits pour la configuration **FR-JUNE21-UNIC-A*-20-DIST-A320-PAR**.

ITUBES-A320					
FL>60	Total	Même flux	Flux \neq	Mixtes	Autres
Nb conflits	2565	339	102	653	1471
% conflits	100.00 %	13.22 %	3.98 %	25.46 %	57.35 %
FL>145	Total	Même flux	Flux \neq	Mixtes	Autres
Nb conflits	1745	308	36	532	869
% conflits	100.00 %	17.65 %	2.06 %	30.49 %	49.80 %
FL>195	Total	Même flux	Flux \neq	Mixtes	Autres
Nb conflits	1371	300	18	489	564
% conflits	100.00 %	21.88 %	1.31 %	35.67 %	41.14 %

TAB. 8.11 – Répartition des conflits pour la configuration **FR-JUNE21-UNIC-A*-20-ITUBES-A320-PAR**.

IZONES					
FL>60	Total	Même flux	Flux \neq	Mixtes	Autres
Nb conflits	2721	396	187	639	1499
% conflits	100.00 %	14.55 %	6.87 %	23.48 %	55.09 %
FL>145	Total	Même flux	Flux \neq	Mixtes	Autres
Nb conflits	1878	358	123	518	879
% conflits	100.00 %	19.06 %	6.55 %	27.58 %	46.81 %
FL>195	Total	Même flux	Flux \neq	Mixtes	Autres
Nb conflits	1476	342	88	476	570
% conflits	100.00 %	23.17 %	5.96 %	32.25 %	38.62 %

TAB. 8.12 – Répartition des conflits pour la configuration **FR-JUNE21-UNIC-A*-20-IZONES-PAR**.

temps d'atteindre le niveau FL145 avant de sortir de l'espace aérien français. C'est donc sur 71 flux qu'opère l' A^* .

Le tableau 8.9 présente les résultats de l' A^* avec routes parallèles selon les trois modes différents de détection des interférences, pour la journée du 21 juin 2002, en n'allouant qu'une seule trajectoire par flux (modèle UNIC). Les trois premières lignes donnent pour chaque mode la valeur de la *fitness* \mathcal{F} , le nombre de flux pour lesquels l' A^* n'a pas trouvé de solution, le temps de calcul en secondes sur un CPU Xeon 2.8 GHz. Il a paru intéressant de calculer et de présenter la valeur de \mathcal{F} pour les résultats de l' A^* afin de pouvoir comparer plus tard les résultats de l' A^* et de l'algorithme génétique (rappelons que \mathcal{F} n'est en fait utilisée que dans le processus de sélection de l'algorithme génétique, et pas dans l' A^*). Par ailleurs, cette valeur est une synthèse des informations sur les échecs dans la recherche de solutions et du coût des solutions trouvées : lorsqu'elle est inférieure à 1 c'est qu'il reste des interférences entre trajectoires, et dans le cas contraire, plus \mathcal{F} est élevée plus le coût cumulé des solutions est faible.

La partie intermédiaire du tableau détaille quelques éléments de comparaison des résultats, avec :

- le coût cumulé des déviations de trajectoires (tel que défini en 7.2.4),
- le nombre de niveaux de croisière occupés en espace inférieur (en-dessous du FL145) et en espace supérieur (au-dessus du FL145),
- la valeur des niveaux les plus bas pour le premier et le dernier palier (pour lesquels on s'autorise à descendre jusqu'au FL60) ainsi que pour les paliers intermédiaires,
- l'allongement relatif des trajectoires par rapport à la route la plus directe,
- et enfin le pourcentage de trafic que représentent les flux pour lesquels on a trouvé une solution.

Le bas du tableau présente une synthèse des résultats de la validation par le simulateur CATS/OPAS, avec trois niveaux planchers différents pour la détection des conflits (les conflits sont détectés au-dessus du FL60, du FL145, ou du FL195). Pour chaque niveau plancher, le tableau présente

- le nombre total n_{ref} de conflits détectés pour la situation de référence (rappelons que pour les deux premiers modes de détection tous les avions sont du type A320, et que seul le mode **IZONES** permet des comparaisons avec une situation de référence réaliste, avec tous les types d'avions),
- le nombre total n de conflits détectés pour le trafic modifié, avec des trajectoires 3D imposées aux flux principaux,
- le nombre m de conflits entre avions appartenant au même flux, qui sont censés être résolus par un séquençement au départ,
- et enfin le gain en terme de conflits détectés $\frac{n_{ref} - (n - m)}{n_{ref}}$.

Constatons tout d'abord que l'on ne trouve pas des solutions sans interférences pour tous les modes de détection, puisqu'il y a un échec avec la détection par intersection de zones d'incertitude. Ce qui veut dire qu'il y a un flux pour lequel l' A^* n'a pas trouvé de trajectoire sans interférences. Ce n'est ni très étonnant, ni très significatif : pour d'autres journées de trafic, on peut trouver des solutions pour les trois modes, ou bien avoir des échecs dans d'autres modes. Ce nombre d'échecs peut également être totalement différent, pour peu que l'on change l'ordre dans lequel les flux sont considérés.

Une deuxième constatation est que la détection basée sur l'intersection de tubes semble plus efficace en terme de coût que celle basée sur la distance d , ce que l'on confirmera sur d'autres jeux de tests dans la suite. Le coût de la solution **ITUBES-A320** est de 260.03, contre 296.64 pour **DIST-A320**. Le coût de la solution **IZONES** est mis entre parenthèses, pour signaler qu'il ne peut pas être comparé aux autres puisqu'il est calculé avec un flux de moins.

Les résultats sur la validation par simulations montrent des gains significatifs en terme de conflits

détectés, quel que soit le mode de détection. Pour le mode **IZONES**, le plus réaliste, on se situe entre 24% et 28% de conflits en moins selon le niveau plancher choisi pour la détection, pour un volume de trafic correspondant à moins de 40% du trafic total. Les conflits en rattrapage entre avions appartenant à un même flux sont soustraits au total des conflits : on a fait l'hypothèse qu'ils peuvent être réglés par un séquençement au départ, et éventuellement une régulation en vitesse.

Le tableau 8.10 détaille les résultats des simulations CATS/OPAS en fournissant le nombre et le pourcentage de conflits :

- total,
- entre avions appartenant à un même flux,
- entre avions appartenant à des flux différents,
- entre le trafic sur les flux ayant une trajectoire 3D allouée et le reste du trafic,
- entre avions appartenant au reste du trafic.

On constate, pour un plancher de détection au niveau FL145, que près de la moitié des conflits (46,5%) se produisent entre avions n'appartenant pas aux flux principaux (cf dernière colonne). Les 329 conflits entre avions appartenant à un même flux représentent 17,6% du total. Il reste par ailleurs un nombre assez conséquent de conflits mixtes, un peu plus de 31%, entre du trafic sur des trajectoires 3D allouées et du trafic n'appartenant pas aux flux principaux. Notons enfin qu'il reste un faible pourcentage (moins de 5%) de conflits entre avions appartenant à des flux différents. Les tableaux 8.11 et 8.12 détaillent le même type de résultats pour les deux autres modes de détection.

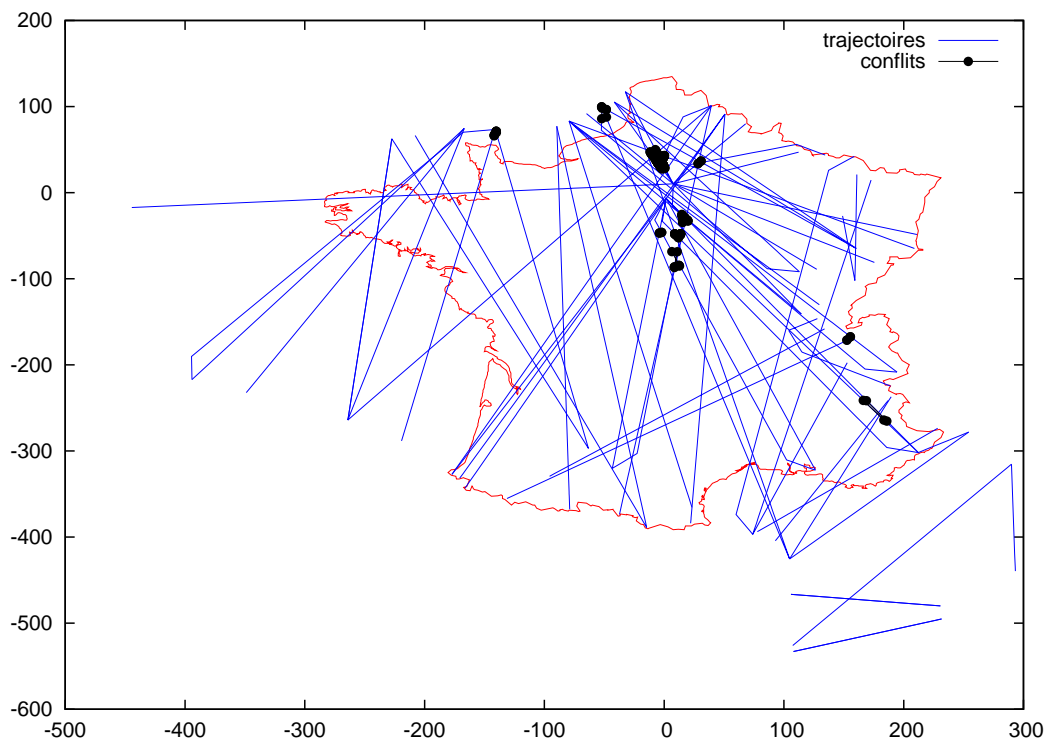


FIG. 8.21 – Conflits restants entre avions appartenant à des flux différents, pour la configuration FR-JUNE21-UNIC-A*-20-DIST-A320-PAR (projection stéréographique centrée sur Paris ; distances en miles nautiques).

Examinons de plus près la nature des conflits restants entre flux différents. Les figures 8.21 et 8.22 présentent une vue en plan et une vue tridimensionnelle des trajectoires trouvées pour chaque flux,

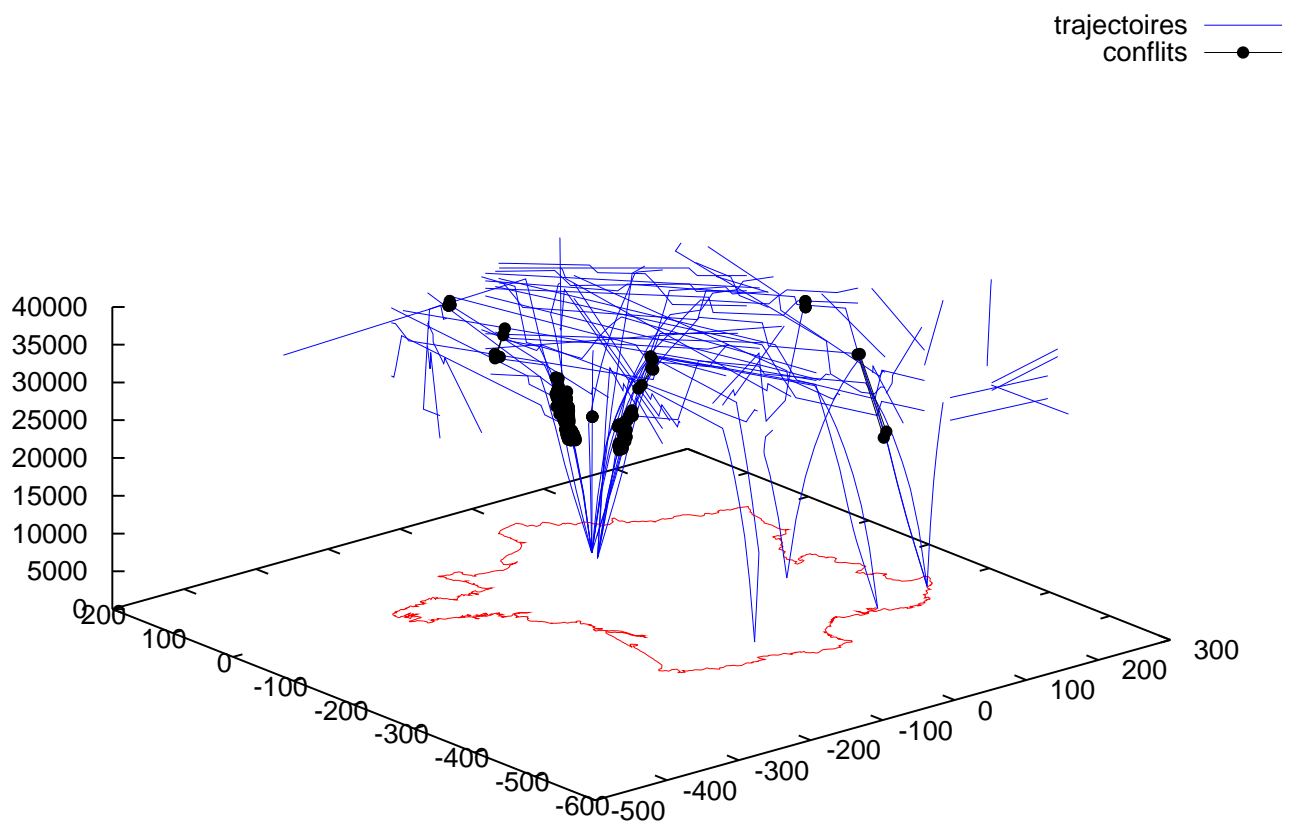


FIG. 8.22 – Vue 3D des conflits restants entre avions appartenant à des flux différents, pour la configuration FR-JUNE21-UNIC-A*-20-DIST-A320-PAR (plan horizontal : projection stéréographique centrée sur Paris ; distances horizontales en milles nautiques ; altitudes en pieds).

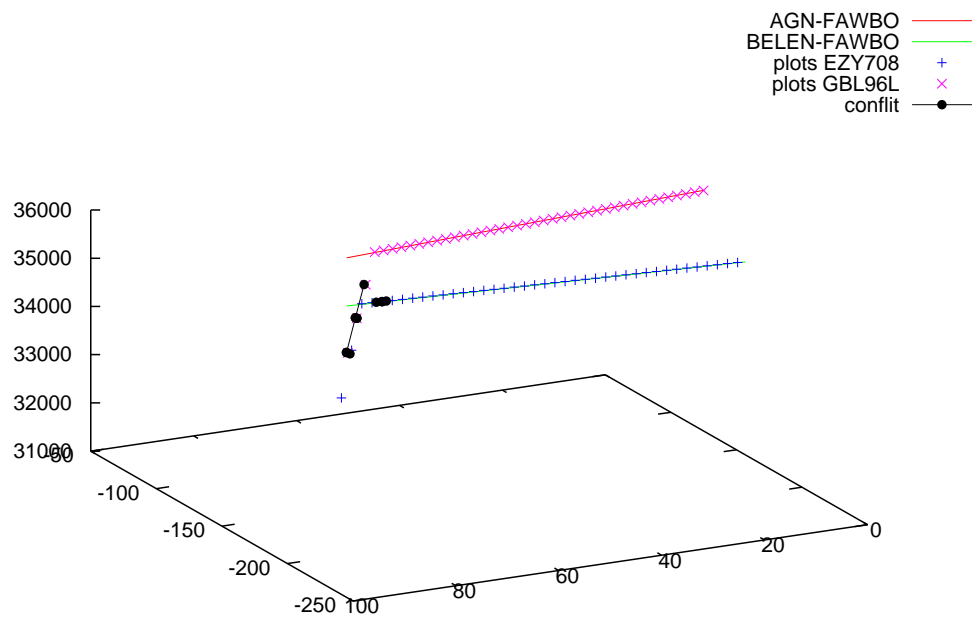


FIG. 8.23 – Conflit entre EZY708 sur BELEN-FAWBO et GBL96L sur AGN-FAWBO, avec la configuration FR-JUNE21-UNIC-A*-20-DIST-A320-PAR (projection stéréographique centrée sur Paris ; distances horizontales en milles nautiques ; altitudes en pieds).

ainsi que la trace des conflits détectés par le simulateur de trafic, entre avions appartenant à des flux différents, pour le mode de détection **DIST-A320**.

On constate de visu, ainsi qu'à l'examen des fichiers de traces sauvegardés lors de la simulation, que la quasi-totalité de ces conflits se produit entre des avions en montée provenant d'un même aéroport, ou entre des avions en descente vers un même aéroport. Rappelons que la détection d'interférences a justement été inhibée pour ce type de cas, pour lesquels on considère que la séparation doit être assurée par un séquençement au départ ou à l'arrivée par le contrôle d'approche.

Il reste malgré tout quelques conflits du type de celui présenté sur la figure 8.23, où quelques avions entrent en conflit en tout début de descente vers un aéroport étranger (Londres sur l'exemple), juste avant la sortie de l'espace aérien français. Il semble y avoir une légère différence entre les trajectoires calculées pour déterminer les niveaux de sortie et celles calculées par le simulateur. Une explication possible est que, dans le premier cas, on calcule les profils de vol dans un espace euclidien dont les points dans plan horizontal sont obtenus par une projection stéréographique, alors que, dans le deuxième cas, le simulateur calcule les positions successives des avions en latitude et longitude.

8.10.2 FR-JUNE21-UNIC-A*-10-PAR

Mode de détection	DIST-A320	ITUBES-A320	IZONES
\mathcal{F}	0.001199	0.002732	0.013158
Nb. échecs	23	16	3
Temps (s)	3845.89	3654.57	4753.57
Coût	(887.90478)	(844.52559)	(590.46861)
Nb. FL esp. sup.	25	25	26
Nb. FL esp. inf.	6	3	1
FL min prem.	60	60	150
FL min der.	80	90	140
FL min autres	150	150	150
FL max	390	390	400
Allgt.	1.54 %	1.17 %	0.29 %
% trafic traité	46.48 %	48.40 %	52.87 %
Nb. conflits REF FL>195	1389	1389	1582
Total conflits FL>195	1200	1143	1375
Même flux FL>195	312	290	390
Gain FL>195	36.1 %	38.6 %	37.7 %

TAB. 8.13 – Résultats de l' A^* pour la configuration de test **FR-JUNE21-UNIC-A*-10-PAR**.

Sur les 157 flux de 10 avions ou plus, l' A^* rencontre un taux d'échecs (voir tableau 8.13) relativement plus important que pour les 71 flux de plus de 20 avions. C'est tout-à-fait logique, sachant qu'à chaque nouvelle trajectoire que l'on calcule, c'est un peu plus d'espace que l'on occupe, réduisant d'autant les possibilités de trouver un chemin sans interférences pour les trajectoires suivantes. On remarque malgré tout le faible nombre d'échecs (seulement trois) avec le mode de détection le plus réaliste basé sur l'intersection des zones d'incertitudes.

La comparaison des coûts selon les différents modes de détection n'a pas grand sens, étant donné qu'ils représentent le cumul des coûts des trajectoires pour lesquelles l' A^* a trouvé une solution sans interférences, et qu'ils ne tiennent pas compte de celles pour lesquelles il y a eu échec.

Le gain en nombre de conflits est tout à fait significatif : le fait de séparer dans l'espace les 157 flux regroupant environ la moitié du trafic total, permet d'espérer une diminution de près de quarante pour cent (37,7% exactement pour le mode IZONES) du nombre des conflits au-dessus du FL195, en faisant l'hypothèse que l'on sache régler les conflits entre avions se suivant sur une même trajectoire 3D par un séquençement temporel et une régulation en vitesse.

Les résultats pour cette configuration de test sont donc très intéressants, mais doivent toutefois être relativisés. N'oublions pas que l'on s'est permis de choisir les niveaux d'entrée ou de sortie de l'espace aérien français sans tenir compte des éventuelles contraintes imposées par les organismes gérant les espaces adjacents. En pratique, les niveaux de vol auxquels sont transférés les avions doivent faire l'objet d'une coordination entre les organismes de contrôle concernés.

8.10.3 FR-JUNE21-UNIC-GA-20-PAR

Mode de détection	DIST-A320	ITUBES-A320	IZONES
\mathcal{F}	1.282858	1.315365	1.300095
Nb. échecs	0	0	
Temps (s)	23761.60	24831.51	66970.20
Coût	250.03534	224.15867	235.61843
Nb. FL esp. sup.	18	20	19
Nb. FL esp. inf.	0	0	1
FL min prem.	180	180	140
FL min der.	180	180	180
FL min autres	280	160	290
FL max	390		390
Allgt.	0.8 %	0.4 %	0.53 %
% trafic traité	39.60 %	39.60 %	39.60 %
Nb. conflits REF FL>195	1389	1389	1582
Total conflits FL>195	1315	1346	1453
Même flux FL>195	308	303	351
Gain FL>195	27.5 %	24.9 %	30.3 %

TAB. 8.14 – Résultats de l'algorithme génétique pour la configuration de test **FR-JUNE21-UNIC-GA-20-PAR**.

L'algorithme génétique fournit des résultats sur les flux de 20 avions ou plus, présentés sur le tableau 8.14, que l'on peut comparer avec ceux fournis par l' A^* de la section 8.10.1. Les temps de calcul, toujours sur un CPU Xeon 2.8 GHz, sont plus longs, l'algorithme génétique trouve des solutions sans interférences pour tous les modes de détection, de meilleure qualité que celles de l' A^* .

Rappelons encore une fois que les deux algorithmes ne cherchent pas à résoudre exactement le même problème. Le fait de les comparer ne donne en fait qu'une indication des qualités et des inconvénients de chacune des deux stratégies de recherche de trajectoires 3D (*I contre n* ou globale).

Le gain en nombre de conflits, toujours évalué en faisant l'hypothèse que les conflits en rattrapage sur une même trajectoire 3D sont résolus indépendamment, est globalement un peu meilleur que pour l' A^* . Une explication possible est que les trajectoires trouvées sont globalement moins déviées et occupent en quelque sorte moins d'espace, ce qui fait qu'elles interfèrent moins avec le reste du trafic. En dehors de cela, il n'y a guère de raisons pour que les avions volant sur les trajectoires séparées

gènèrent moins de conflits dans un cas que dans l'autre.

8.10.4 FR-JUNE21-UNIC-A*-20-STD

Mode de détection	DIST-A320	ITUBES-A320	IZONES
\mathcal{F}	0.029412	0.045455	1.105229
Nb. échecs	1	1	0
Temps (s)	342.98	200.56	230.42
Coût	(714.43164)	(670.59119)	683.25604
Nb. FL esp. sup.	23	20	19
Nb. FL esp. inf.	1	0	0
FL min prem.	100	180	180
FL min der.	170	180	180
FL min autres	180	210	230
FL max	390	390	400
Allgt.	0.00 %	0.00 %	0.02
% trafic traité	39.04 %	39.04 %	40.02 %
Nb. conflits REF FL>195	1389	1389	1582
Total conflits FL>195	1345	1372	1496
Même flux FL>195	303	298	357
Gain FL>195	25.0 %	22.7 %	28.0 %

TAB. 8.15 – Résultats de l'A* pour la configuration de test **FR-JUNE21-UNIC-A*-20-STD**.

Dans les résultats précédents, le choix de la route était restreint à trois routes parallèles, les plus directes possibles. Nous allons voir maintenant les résultats obtenus lorsque la route est choisie parmi les itinéraires standards, sur le réseau de routes utilisé en opérationnel. Rappelons que l'ensemble des routes possibles entre deux points est dans ce cas déterminé en examinant l'ensemble des itinéraires demandés par les avions dans les plans de vol archivés.

Le nombre de flux est de 72, soit un de plus que pour les routes directes. Les flux initiaux sont légèrement différents en raison du rallongement des trajectoires qui permet à quelques vols de plus d'atteindre le FL145 avant de sortir de l'espace.

La figure 8.24 montre une vue dans le plan des trajectoires trouvées avec le mode de détection IZONES, et la figure 8.25 montre ces mêmes trajectoires vues de côté, en se plaçant à l'Ouest.

Sur le tableau 8.15, on constate un échec pour chacun des deux premiers modes de détection, alors qu'on obtient une solution sans interférences pour le modes IZONES.

Remarquons que l'allongement des routes, calculé par rapport aux itinéraires standards les plus courts, est ici nul ou quasi-nul. L'explication tient au fait qu'il y a assez peu de variété, pour un même flux origine-destination, dans les itinéraires archivés dans les plans de vol. Ceux-ci diffèrent d'une ou deux balises au plus, et ils sont sensiblement équivalents en terme de distance. C'est une conséquence du principe économique qui veut que les opérateurs aériens demandent systématiquement la route la moins coûteuse. Les déviations permettant d'éviter les autres trajectoires se font donc quasi-exclusivement dans le plan vertical. Pour pouvoir proposer des routes qui soient réellement une alternative à l'itinéraire le plus direct, il faudrait donc pouvoir disposer de données archivées décrivant tous les itinéraires possibles (segments reliant les balises, sens uniques...) afin d'en calculer quelques uns décalés vers la droite ou vers la gauche.

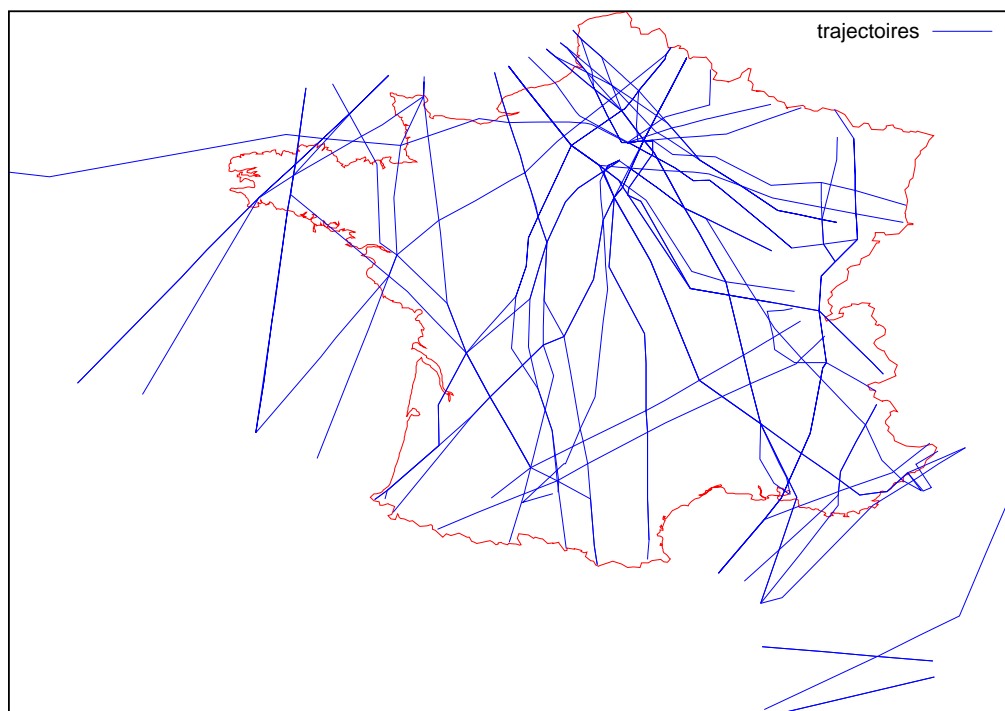


FIG. 8.24 – Trajectoires trouvées par l'A* pour la configuration de test FR-JUNE21-UNIC-A*-20-IZONES-STD.

La diminution du nombre de conflits est sensiblement équivalente à celle constatée pour la configuration de test avec les routes directes (cf tableau 8.9 dans la section 8.10.1 de ce chapitre), sauf pour le mode DIST-A320 où elle est plus forte avec les routes standard (25 % au lieu de 19 %).

Ces résultats démontrent qu'il est possible d'appliquer l'algorithme proposé à des situations très réalistes avec un trafic évoluant sur le réseau de routes opérationnel, bien qu'il reste encore des améliorations à apporter dans le calcul des itinéraires alternatifs, et ceci avec des gains potentiels intéressants. Faute de disposer des données permettant un calcul de routes alternatives réalistes, nous nous focaliserons essentiellement dans la suite des résultats sur l'utilisation de routes directes, en examinant les possibilités d'allouer plusieurs trajectoires 3D par flux origine destination.

8.10.5 FR-JUNE21-PROX-A*-15-PAR

Ici, on s'autorise pour chaque flux des trajectoires supplémentaires pour les avions évoluant vers des aéroports proches de la frontière. La détection d'interférence est désactivée entre trajectoires appartenant à un même trajet origine-destination, et les sous-flux correspondant à une évolution vers un aéroport proche sont considérés immédiatement après le flux principal dont ils sont issus. De cette façon, un avion devant par exemple débiter une descente vers un aéroport proche pourra suivre la trajectoire 3D du flux principal pendant un certain temps, puis en diverger au moment de commencer la descente.

Le nombre de trajectoires est ici de 95, en considérant les flux de 15 avions ou plus, au dessus-de la France. La façon de calculer les niveaux préférentiels d'entrée et de sortie est légèrement différente que pour le modèle UNIC : au lieu de considérer pour chaque vol les profils nominaux puis de prendre

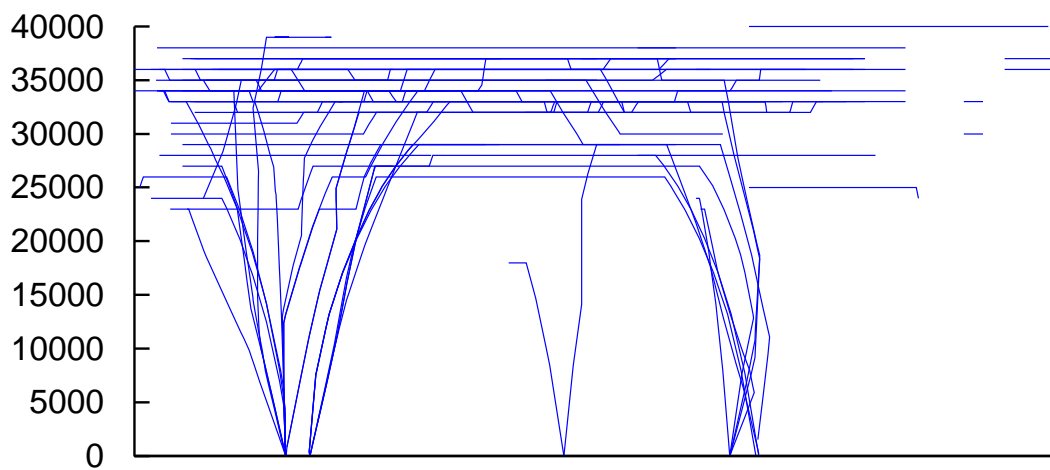


FIG. 8.25 – Vue de côté (Ouest) des trajectoires trouvées par l'A* pour la configuration de test FR-JUNE21-UNIC-A*-20-IZONES-STD (altitudes en pieds).

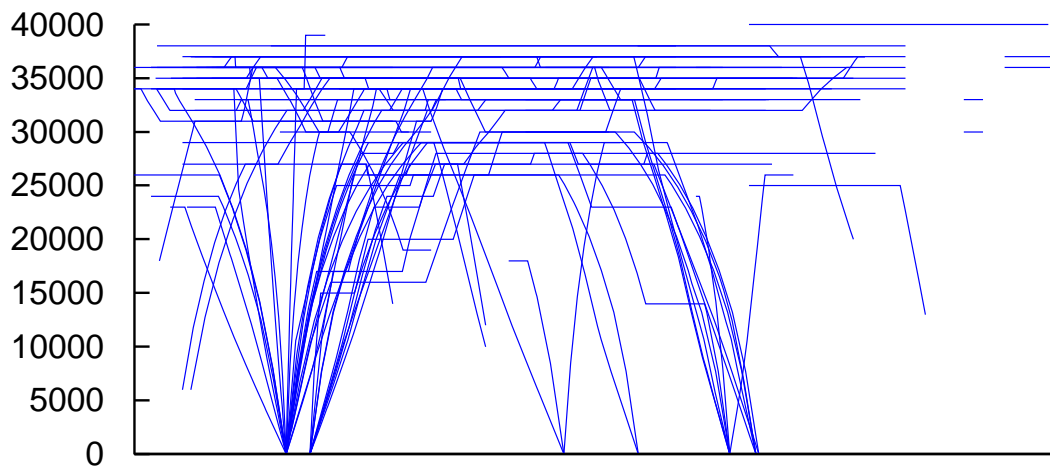


FIG. 8.26 – Vue de côté (Ouest) des trajectoires trouvées par l' A^* pour la configuration de test FR-JUNE21-PROX-A*-15-IZONES-PAR (altitudes en pieds).

Mode de détection	DIST-A320	ITUBES-A320	IZONES
\mathcal{F}	0.004425	0.125003	0.083335
Nb. échecs	7	1	1
Temps (s)	457.87	314.93	951.94
Coût	(498.82217)	(465.59162)	(370.67098)
Nb. FL esp. sup.	22	20	24
Nb. FL esp. inf.	2	3	1
FL min prem.	160	140	140
FL min der.	90	90	160
FL min autres	160	180	260
FL max	390	390	400
Allgt.	1.22 %	1.04 %	0.25 %
% trafic traité	39.79 %	41.70 %	41.58 %
Nb. conflits REF FL>195	1389	1389	1582
Total conflits FL>195	1357	1322	1450
Même flux FL>195	284	300	305
Gain FL>195	22.8 %	26.4 %	27.6 %

TAB. 8.16 – Résultats de l'A* pour la configuration de test **FR-JUNE21-PROX-A*-15-PAR**.

la borne inférieure comme niveau d'entrée, on utilise ici les performances minimales et maximales pour effectuer ce même calcul. En conséquence, un nombre plus important de vols au décollage ou à destination de la TMA Paris franchissent la frontière à un niveau de sortie en-dessous du niveau de coupure FL145, et le calcul des flux en est légèrement modifié.

Le tableau 8.16 présente les résultats pour cette configuration de test. La figure 8.26 illustre les trajectoires trouvées, vue de côté en se plaçant à l'Ouest. On distingue les trajectoires ayant des segments évolutifs en entrée ou en sortie de l'espace, correspondant aux évolutions vers des aéroports proches. Cette figure est à comparer à la figure 8.25 où les flux entraient et sortaient stables en niveau.

Le gain en nombre de conflits est sensiblement équivalent aux résultats obtenus avec une seule trajectoire par flux en considérant un volume équivalent de trafic (configuration FR-JUNE21-UNIC-A*-20-PAR), mais le nombre de trajectoires est plus important.

8.10.6 FR-JUNE21-MULTI-A*-10-PAR

Ici, on calcule plusieurs trajectoires par flux origine-destination en fonction des niveaux d'entrée, de croisière, et de sortie. Les sous-flux sont d'abord calculés pour tous les flux, puis on sélectionne ceux contenant 10 avions ou plus, évoluant au-dessus du FL145, soit 139 sous-flux dans notre cas. On constate sur le tableau 8.17 que ces sous-flux regroupent environ 30% du trafic total.

L'A* a rencontré des échecs pour quelques trajectoires, quel que soit le mode de détection. Les temps de calcul sont à rapprocher de ceux de la configuration de 10 avions ou plus avec le modèle UNIC, pour lesquels on opérait sur 157 flux. Dans le cas du modèle MULTI, une difficulté supplémentaire vient du fait qu'on cherche à éviter les interférences entre sous-flux appartenant à un même flux origine destination, contrairement au modèle PROX.

Les simulations de trafic montrent une diminution du nombre de conflits plus modeste, de l'ordre de 11 à 14%, que dans les autres cas où une proportion plus importante du trafic était considérée.

Mode de détection	DIST-A320	ITUBES-A320	IZONES
\mathcal{F}	0.001678	0.007937	0.031250
Nb. échecs	16	7	2
Temps (s)	4188.04	1960.46	483.01
Coût	(747.88246)	(755.38859)	(511.58789)
Nb. FL esp. sup.	25	24	25
Nb. FL esp. inf.	3	3	1
FL min prem.	60	140	140
FL min der.	130	70	150
FL min autres	150	150	200
FL max	390	390	390
Allgt.	1.14 %	1.14 %	0.21 %
% trafic traité	29.48 %	30.98 %	31.96 %
Nb. conflits REF FL>195	1389	1389	1582
Total conflits FL>195	1337	1387	1527
Même flux FL>195	144	148	139
Gain FL>195	14.1 %	10.8 %	12.3 %

TAB. 8.17 – Résultats de l' A^* pour la configuration de test **FR-JUNE21-MULTI-A*-10-PAR**, en détectant les interférences entre sous-flux.

8.11 Résultats sur l'Europe, au-dessus du FL145

Les résultats présentés pour les données Europe sont uniquement ceux issus des algorithmes de séparation des flux. La validation par une simulation évaluant les conflits restants n'a pas encore été faite. En tout état de cause, étant donné la faible proportion de trafic que représentent les flux entre aéroports auxquels on alloue ici une trajectoire, il ne faudra pas s'attendre à de fortes réductions du nombre de conflits. Nous avons déjà vu dans la section 8.3 tout l'intérêt qu'il y aurait à regrouper les aéroports dans de larges TMAs, pour définir ensuite des trajectoires 3D entre ces TMAs, ce qui permettrait potentiellement de traiter plus de trafic, et donc d'espérer un gain plus important en nombre de conflits.

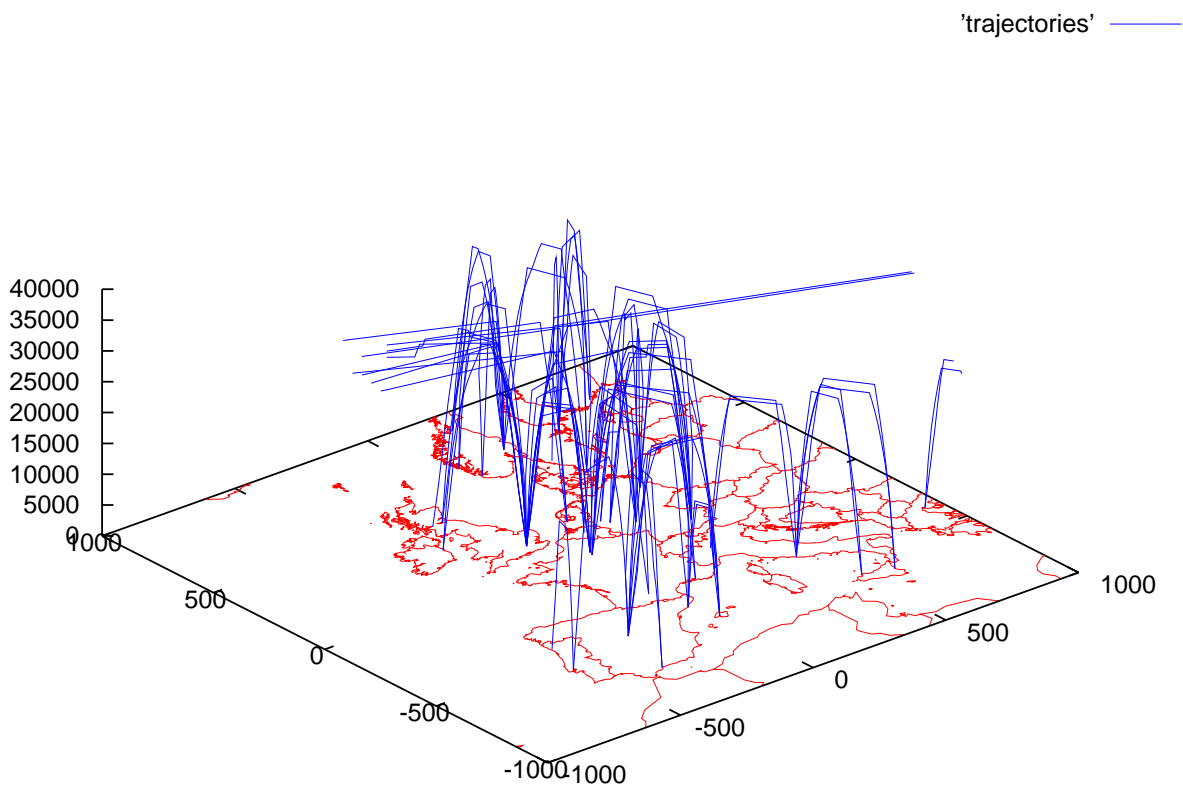
8.11.1 EU-JUNE21-UNIC-A*-20-PAR

Pour les 65 flux dans l'espace européen, de 20 avions ou plus au-dessus du FL145, l' A^* allouant une seule trajectoire par flux (modèle UNIC) trouve des solutions avec les deux modes de détection par intersection de tubes, mais rencontre deux échecs pour le mode DIST-A320. Le détail des résultats est présenté sur le tableau 8.18, et les figures 8.27 et 8.27 montrent les trajectoires allouées, avec le mode de détection IZONES.

On constate *de visu* que la configuration des flux est complètement différente de celle des problèmes dans l'espace aérien français. La solution pour l'Europe se traduit par un réseau en étoile, avec des routes décalées latéralement pour les flux de directions opposées entre deux même aéroports.

L'algorithme A^* a rencontré deux échecs pour le mode de détection basé sur la distance d avec un temps de calcul plus de deux fois plus long qu'avec le mode de détection entre tubes (ITUBES-A320), avec lequel on trouve une solution sans interférences. Pour le mode de détection entre zones d'incertitude, l'algorithme trouve une solution meilleure que pour les deux autres modes, en un temps

Mode de détection	DIST-A320	ITUBES-A320	IZONES
\mathcal{F}	0.010204	1.112137	1.120674
Nb. échecs	2	0	0
Temps (s.)	139.35	53.63	506.06
Coût	(566.08163)	587.59932	545.95383
Nb. FL UIR.	19	17	23
Nb. FL FIR	0	1	0
FL min 1st	170	170	150
FL min der.	160	140	150
FL min autres	-	260	200
FL max	370	370	370
Allgt. route	1.89 %	1.90 %	0.68 %
% trafic traité	6.05 %	6.20 %	6.20 %

TAB. 8.18 – Résultats de l' A^* pour la configuration de test **EU-JUNE21-UNIC-A*-20-PAR**.FIG. 8.27 – Vue 3D de la solution trouvée par l' A^* , pour la configuration de test EU-JUNE21-UNIC-A*-20-IZONES-PAR (plan horizontal : projection stéréographique centrée sur Paris ; distances horizontales en milles nautiques ; altitudes en pieds).

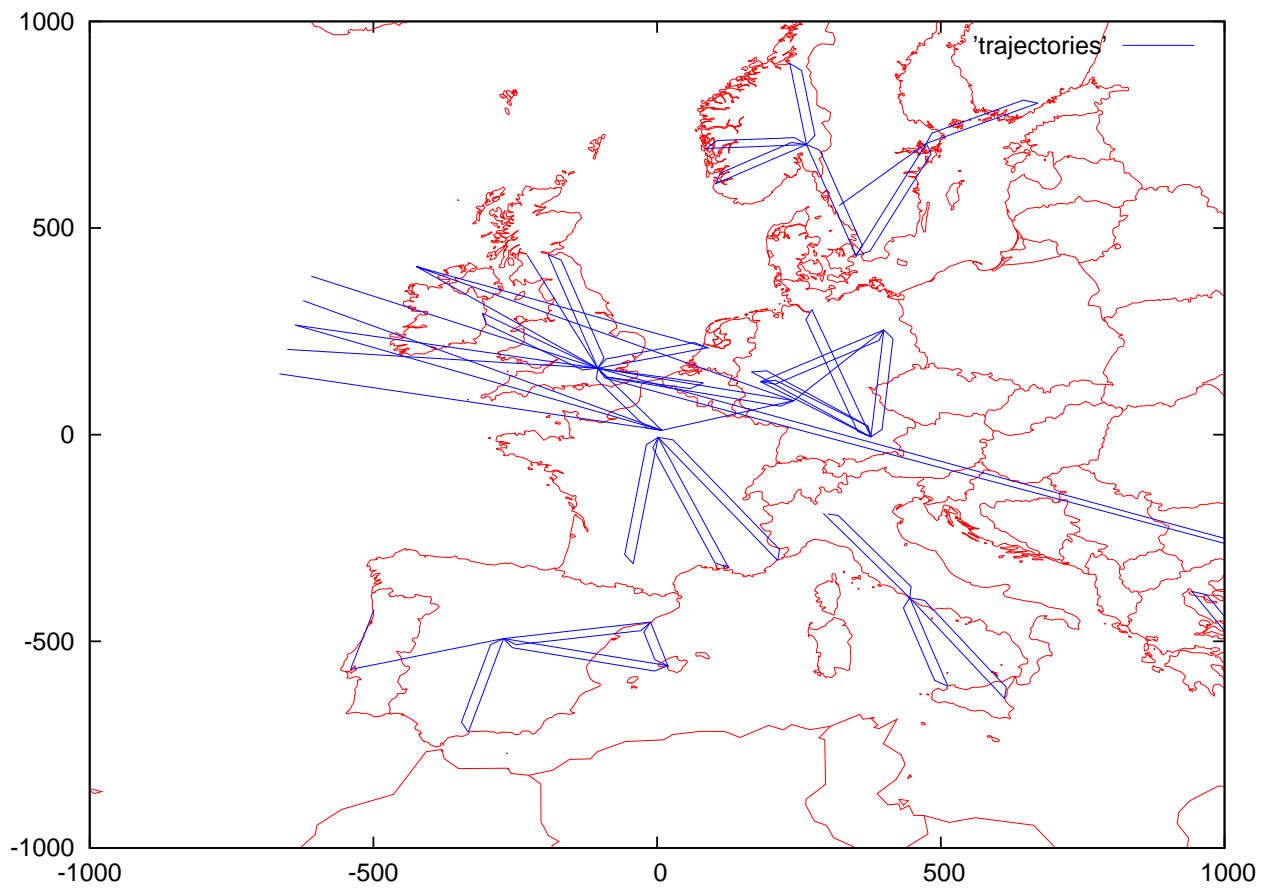


FIG. 8.28 – Solution trouvée par l'A*, pour la configuration de test JUNE21-EU-UNIC-A*-20-IZONES-PAR (projection stéréographique centrée sur Paris ; distances en milles nautiques).

de calcul sensiblement plus long (un peu moins de neuf minutes).

Bien sûr, il ne faut pas se fier à ces seuls résultats pour tirer des conclusions sur les performances des algorithmes selon le mode de détection choisi. Malgré tout, si l'on considère l'ensemble des résultats présentés, on a le sentiment que le fait d'utiliser le mode de détection d'interférences DIST basé sur la distance d mène à des solutions de moins bonne qualité, en des temps plus longs, que lorsqu'on utilise le mode ITUBES basé sur les intersections de tubes. On voit donc que le fait de se donner des marges supplémentaires d'espacement entre les trajectoires (avec la distance d), même relativement minimales, a un impact non négligeable sur la recherche de solutions.

8.11.2 EU-JUNE21-UNIC-GA-20-PAR

Mode de détection	DIST-A320	ITUBES-A320	IZONES
\mathcal{F}	1.112151	1.112542	1.119510
Nb. échecs	0	0	0
Temps (s.)	50309.76	43440.69	204629.79
Coût	587.52588	585.48004	551.29116
Nb. FL UIR	22	20	21
Nb. FL FIR	0	0	0
FL min 1er	160	170	150
FL min der.	170	170	170
FL min autres	180	230	200
FL max	370	370	370
Allgt. route	2.04 %	1.97 %	1.22 %
% trafic traité	6.20 %	6.20 %	6.20 %

TAB. 8.19 – Résultats de l'algorithme génétique pour la configuration de test **EU-JUNE21-UNIC-GA-20-PAR**.

Les résultats de l'algorithme génétique sur la même configuration de flux que précédemment sont présentés sur le tableau 8.19. L'algorithme trouve des solutions sans interférences pour les 65 flux de 20 avions ou plus au dessus de l'Europe, pour les trois modes de détection. La valeur de la *fitness* \mathcal{F} est meilleure que celle calculée pour l' A^* pour les modes de détection DIST-A320 et ITUBES-A320, et légèrement inférieure pour le mode IZONES. Au vu des temps de calcul, il n'a pas été fait de multiples tests avec des valeurs différentes pour la racine du générateur aléatoire, comme dans le chapitre 7. Il peut donc arriver que la solution trouvée soit moins bonne que celle de l' A^* , comme dans le problème jouet des flux sur un quadrillage, en particulier lorsque la solution trouvée par ce dernier est déjà de bonne qualité. Au vu de la configuration des flux origine-destination en Europe, plutôt répartis en étoile, il est possible que l' A^* ait effectivement trouvé une solution assez proche d'un optimum global.

Les temps de calcul de l'algorithme génétique sont plutôt longs, entre 12 et 56 heures pour le cas extrême. La cause n'en a pas encore été explorée. Une piste intéressante pour améliorer les performances de l'algorithme consiste à diviser le problème global en sous-problèmes indépendants, comme nous l'avons décrit dans la section 8.8.2. La mise en place de cette adaptation de l'algorithme est en cours de développement.

Mode de détection	DIST-A320	ITUBES-A320	IZONES
\mathcal{F}	0.022727	0.125002	1.132878
Nb. échecs	2	1	0
Temps (s)	845.15	54.66	1845.97
Coût	(738.67304)	(711.95800)	638.72551
Nb. FL esp. sup.	20	19	23
Nb. FL esp. inf.	0	0	0
FL min prem.	170	160	150
FL min der.	160	230	160
FL min autres	-	250	200
FL max	370	370	370
Allgt. route	1.83 %	1.56 %	0.30 %
% trafic traité	6.24 %	6.29 %	6.34 %

TAB. 8.20 – Résultats de l' A^* pour la configuration de test **EU-JUNE21-MULTI-A*-15-PAR**.

8.11.3 EU-JUNE21-MULTI-A*-15-PAR

Comme nous l'avons vu dans l'analyse des flux, la grande majorité du trafic aérien en Europe est intra-européen. Il n'a donc pas paru utile de tester le modèle PROX tenant compte des flux d'avions évolutifs traversant la frontière de l'espace aérien. Par contre, le modèle MULTI permettant d'allouer plusieurs trajectoires par flux à différents niveaux de vol garde tout son intérêt.

Le processus de classification selon les niveaux de vol fournit 85 sous-flux de 15 avions ou plus pour le trafic aérien, en traitant un peu plus de 6% du trafic total, comme pour les 65 flux du modèle UNIC. Le tableau 8.20 regroupe les résultats pour cette configuration de test, sensiblement analogue à ceux du modèle UNIC, avec des temps de calcul globalement un peu plus longs.

8.12 Conclusion sur l'allocation de trajectoires 3D

Dans ce chapitre, la modélisation simplifiée du chapitre 7 a été considérablement enrichie, et les algorithmes adaptés aux données réelles. Les trajectoires 3D tiennent dorénavant compte, dans le calcul du profil vertical, des performances réelles des avions et de la disparité des types d'avions au sein d'un même flux. Dans le plan horizontal, l'itinéraire peut être choisi soit parmi des routes parallèles les plus directes possibles, soit parmi des itinéraires définis sur le réseau de routes opérationnel. De nouveaux modes de détection d'interférences, par intersection de tubes 3D, ont été proposés, permettant d'une part de tenir compte de zones d'incertitudes englobant les trajectoires, et d'autre part de ne plus déclencher de fausses détections.

L'analyse des flux sur une journée de trafic en France et en Europe a fait ressortir des problématiques relativement différentes en fonction du contexte, avec d'un côté un trafic essentiellement international, avec un bon nombre de flux transitant par des balises frontalières, et de l'autre un trafic majoritairement intra-européen et des flux regroupant chacun moins de trafic que dans le cas du contexte français.

Cette étude a par ailleurs permis de visualiser les différents pics de demandes pour les niveaux de croisière, en fonction de la distance parcourue. Pour un même flux origine-destination, les différents pics de demandes correspondent à différentes classes d'avions, chaque avion demandant le niveau le plus adapté à ces performances.

Pour se rapprocher progressivement d'un certain niveau de réalisme, plusieurs modèles d'allocation de trajectoires ont donc été introduits. Pour le plus simple, une seule trajectoire 3D est associée à chaque flux origine-destination, et les entrées et sorties de l'espace aérien se font stables en niveau. Un deuxième modèle permet d'associer des trajectoires supplémentaires aux flux d'avions traversant la frontière de l'espace en configuration de descente ou de montée pour se poser ou décoller d'aéroports proches de la frontière. Enfin, le modèle le plus complexe classe les avions de chaque flux origine-destination en fonction des niveaux de vol demandés en entrée, en croisière, et en sortie, en utilisant la méthode d'agrégation présentée au chapitre 3 (3.3). Ce modèle permet ensuite d'affecter une trajectoire 3D à chaque sous-flux ainsi obtenu.

Sur les données France, les différents tests ont démontré l'efficacité des algorithmes lorsqu'il s'agit de séparer un nombre de flux correspondant à environ 40% du trafic, ce qui représente environ 70 trajectoires pour le modèle allouant une trajectoire par flux, une petite centaine en tenant compte des flux évolutifs vers les aéroports proches. Pour le modèle allouant plusieurs trajectoires 3D par flux en fonction des niveaux demandés, il faut près de 140 sous-flux pour pouvoir traiter environ 30% du trafic. Pour l'algorithme A^* , le taux d'échec est alors très faible ou nul et les temps de calcul sont courts (quelques minutes). L'algorithme génétique trouve dans tous les cas testés des solutions sans interférences, de meilleure qualité que celles de l' A^* , mais avec des temps de calcul plus longs (plusieurs heures).

L'intérêt de séparer les flux en leur allouant des trajectoires 3D non-interférentes a été évalué au travers de simulations en temps accéléré, en utilisant le simulateur CATS/OPAS modifié en conséquence. En réorganisant les flux pour environ 40% du trafic, on constate une diminution de l'ordre de 30% dans le nombre de conflits détectés au dessus du niveau de vol FL 195, en simulant la totalité du trafic. Ce gain est calculé en faisant l'hypothèse que les conflits en rattrapage, entre avions suivant une même trajectoire 3D, puissent être résolus dès le départ par un séquençage temporel et une régulation en vitesse.

Le gain en nombre de conflits passe à près de 40% (38% pour être exact) lorsqu'on alloue des trajectoires à un plus grand nombre de flux, regroupant un peu plus de 50% du trafic total. Cela fait par exemple 157 trajectoires à séparer, en affectant une seule trajectoire par flux de 10 avions ou plus. L'encombrement croissant de l'espace, au fur et à mesure que l'on calcule les trajectoires, augmente rapidement la difficulté pour l' A^* à séparer chaque nouvelle trajectoire des précédentes. Les temps de calcul augmentent en conséquence, en fonction du nombre d'états que l'algorithme doit parcourir avant de trouver un cheminement sans interférences, ou de déterminer qu'il n'y en a pas.

Quoi qu'il en soit, il n'est pas nécessairement judicieux de vouloir séparer un trop grand nombre de flux : affecter une trajectoire 3D permanente pour un flux comprenant seulement quelques avions dans la journée ne relève pas d'une utilisation efficace de l'espace.

Sur les données Europe, les solutions trouvées montrent une configuration de flux assez différente de la France, avec un réseau en étoile entre les principaux aéroports. L'évaluation du nombre de conflits par une simulation CATS/OPAS n'a pas encore été réalisée, mais au vu du volume de trafic traité (à peine plus de 6% pour les flux de 20 avions ou plus), on ne doit pas s'attendre à des gains aussi spectaculaires que pour les données France. Il y aurait donc tout intérêt à regrouper les aéroports dans des TMAs d'assez grande taille, comme dans l'étude TOSCA ([MGZK98]), avant de chercher à définir un réseau de trajectoires 3D séparées entre ces TMAs.

L' A^* se comporte sensiblement comme pour les données France, en terme de performances en fonction du nombre de flux traités. L'algorithme génétique montre généralement de meilleurs résultats que l' A^* sur les configurations testées, au prix toutefois de temps de calcul assez longs.

Au vu de la configuration des flux, il paraît possible d'améliorer les performances de l'algorithme en résolvant indépendamment, et à terme en parallèle, les sous-ensembles de trajectoires interférant

entre elles. Il faut ensuite fusionner si besoin est certains de ces groupes lorsque les solutions trouvées engendrent de nouvelles interférences entre trajectoires appartenant à des sous-ensembles différents, puis relancer à nouveau la résolution. Ces étapes sont répétées jusqu'à ce qu'aucune interférence ne soit détectée entre les solutions des différents sous-problèmes. Dans le pire des cas, ce processus aboutit à une résolution portant sur un seul groupe comprenant l'ensemble des trajectoires. Cette amélioration est en cours de développement et n'a pas encore été testée.

En conclusion, les perspectives de développement pour les algorithmes de séparation des trajectoires 3D sont d'une part la parallélisation de l'algorithme génétique, et d'autre part l'application dans le contexte européen à des flux entre TMAs plutôt qu'entre aéroports, sachant qu'il faudra alors organiser les flux de trafic à l'intérieur des TMAs.

Un autre axe de développement, que nous allons aborder dans le chapitre suivant, consiste à introduire la dimension temporelle dans la détection d'interférences. L'objectif n'est plus alors de définir un réseau de trajectoires 3D statiques, mais d'allouer dynamiquement des trajectoires 4D à chaque avion ou à des trains d'avions évoluant ensemble.

Chapitre 9

Planification de trajectoires 4D non conflictuelles pour des trains d'avions.

9.1 Introduction

Dans les chapitres précédents, nous n'avons pas du tout tenu compte de la dimension temporelle : les trajectoires construites dans l'espace sont allouées de façon permanente à leur flux, ou du moins pour une période assez longue (la journée par exemple).

Or il est bien évident que pour des flux de taille modeste, comportant par exemple quelques vols le matin et quelques autres le soir, il n'est pas utile ni même souhaitable de définir une trajectoire 3D permanente. Il pourrait donc être intéressant de tenir compte de la distribution temporelle des avions sur les flux pour définir des *trains d'avions*, ce qui permettrait d'allouer une trajectoire à chaque train, avec une validité limitée dans le temps.

En poussant ce raisonnement un peu plus loin, on voit immédiatement que pour les flux comportant encore moins d'avions, il vaudrait mieux ne pas geler la totalité de la trajectoire (et donc monopoliser de l'espace) pendant toute la durée d'occupation du flux par les avions.

Pour tenir compte de ces arguments, il est possible de découper chaque trajectoire en segments d'une certaine longueur, et d'associer à chaque segment un intervalle de temps pendant lequel il est susceptible d'être occupé par un avion. Ces intervalles temporels sont alors utilisés dans la détection des interférences. Par exemple, deux segments qui se croisent au même niveau de vol, mais dont les intervalles temporels ne se recoupent pas, ne seront pas interférents.

9.2 Définition des trains d'avions

Le principe d'agrégation par nuées dynamiques, décrit au chapitre 3, section 3.3, est à nouveau utilisé pour définir les *trains d'avions* de chaque flux. L'algorithme des nuées dynamiques nécessite de définir les fonctions suivantes, opérant sur un ensemble de points :

- une distance,
- une fonction calculant le barycentre de deux points,

et également deux seuils de décision, permettant de régler la granularité des agrégats créés par l'algorithme :

- d_1 : un point sera intégré à l'agrégat le plus proche (avec la distance choisie) s'il est à une distance inférieure à d_1 du barycentre de l'agrégat,
- d_2 : deux agrégats seront fusionnés s'ils sont à une distance inférieure à d_2 .

Deux critères sont susceptibles d’être pris en compte pour regrouper les avions. Tout d’abord, il y a le critère de l’heure d’entrée de chaque avion sur le flux. Le but est alors de construire des groupes d’avions tels que les temps d’entrée des avions appartenant à un même groupe soient les plus proches possibles, et que les groupes soient temporellement suffisamment éloignés. L’ensemble de points sur lequel opère l’algorithme d’agrégation est alors l’ensemble des temps d’entrée des avions du flux. La fonction distance est simplement l’écart en temps, et le barycentre de deux temps est calculé selon la définition classique.

L’autre critère consiste à regrouper les avions en fonction de leurs performances de vol. L’objectif dans ce cas est de construire des groupes d’avions ayant des performances de montée et de descente, et si nécessaire des vitesses de croisière, suffisamment homogènes. Les niveaux de vol en entrée de l’espace aérien, en croisière, et en sortie sont des conséquences directes de ces performances de vol. Il a donc été choisi de regrouper les vols selon ces niveaux, comme dans le modèle MULTI du chapitre 8 (section 8.4), avec une méthode de partitionnement par nuées dynamiques (cf chapitre 3, section 3.3).

La définition préalable des trains d’avions se fait donc pour chaque flux origine-destination en classant tout d’abord les avions selon les trois niveaux de vol (EFL,RFL,XFL) (ou seulement les niveaux d’entrée et de sortie dans les cas particuliers où il n’y a pas de segment en croisière). Chaque sous-flux est alors partitionné en utilisant le critère de l’heure d’entrée. Le trafic actuel n’étant pas spécifiquement organisé pour gérer des trains d’avions, il est nécessaire d’utiliser des seuils d’agrégation assez permissifs si l’on ne veut pas se retrouver avec des trains ne comprenant qu’un seul avion. Les résultats présentés par la suite utilisent des valeurs de soixante minutes pour le seuil d_1 (intégration d’un vol dans un groupe), et vingt minutes pour le seuil d_2 (fusion de deux groupes).

9.3 Prise en compte de la dimension temporelle dans le modèle

9.3.1 Trajectoires 4D

Nous avons vu dans le chapitre 8 que le fichier des performances avions contenait des données relatives aux *vitesses air* en vol stable, en montée, ou en descente. C’est à partir de celles-ci que nous allons calculer les temps sur chaque point de la trajectoire, avec deux possibilités en fonction du modèle de trajectoire :

- pour le modèle de trajectoire filaire calculée à partir des performances nominales d’un type d’avion donné, on prendra celles-ci pour calculer un unique temps de passage par point de la trajectoire,
- pour le modèle de trajectoire avec une zone d’incertitude basée sur l’enveloppe des performances d’un ou de plusieurs types d’avions, on prendra ces performances pour calculer un temps de passage *au plus tôt* et un temps de passage *au plus tard* sur chaque point de la trajectoire.

A partir de ces temps de passage, on calcule pour chaque segment l’intervalle maximal d’occupation.

9.3.2 Détection d’interférences en 4D

Les différents modes de détection d’interférences sont strictement identiques à ceux du 8.7, si ce n’est que l’on ajoute un test préalable de recouvrement des intervalles temporels calculés pour chacun des segments de trajectoires.

9.4 Choix de l'algorithme

Au vu des temps de calcul rencontrés avec l'algorithme génétique sur la recherche de trajectoires 3D sans interférences (chapitre 8), il n'a pas été jugé pertinent de tester celui-ci tel quel sur le problème de planification 4D, avec un plus grand nombre de trajectoires. Il paraît plus que jamais nécessaire de paralléliser au préalable l'algorithme génétique, en l'appliquant à des sous-problèmes partiellement indépendants (des ensembles de trajectoires interférant entre elles), selon le mécanisme évoqué en 8.8.2.

Seul l'algorithme A^* développé au chapitre 8 a donc été utilisé, avec une modélisation des trajectoires et une détection d'interférence prenant en compte la dimension temporelle. En dehors de cette adaptation, le mécanisme de l'algorithme reste inchangé par rapport à ce qui est décrit en 8.8.1

9.5 Résultats pour les données France

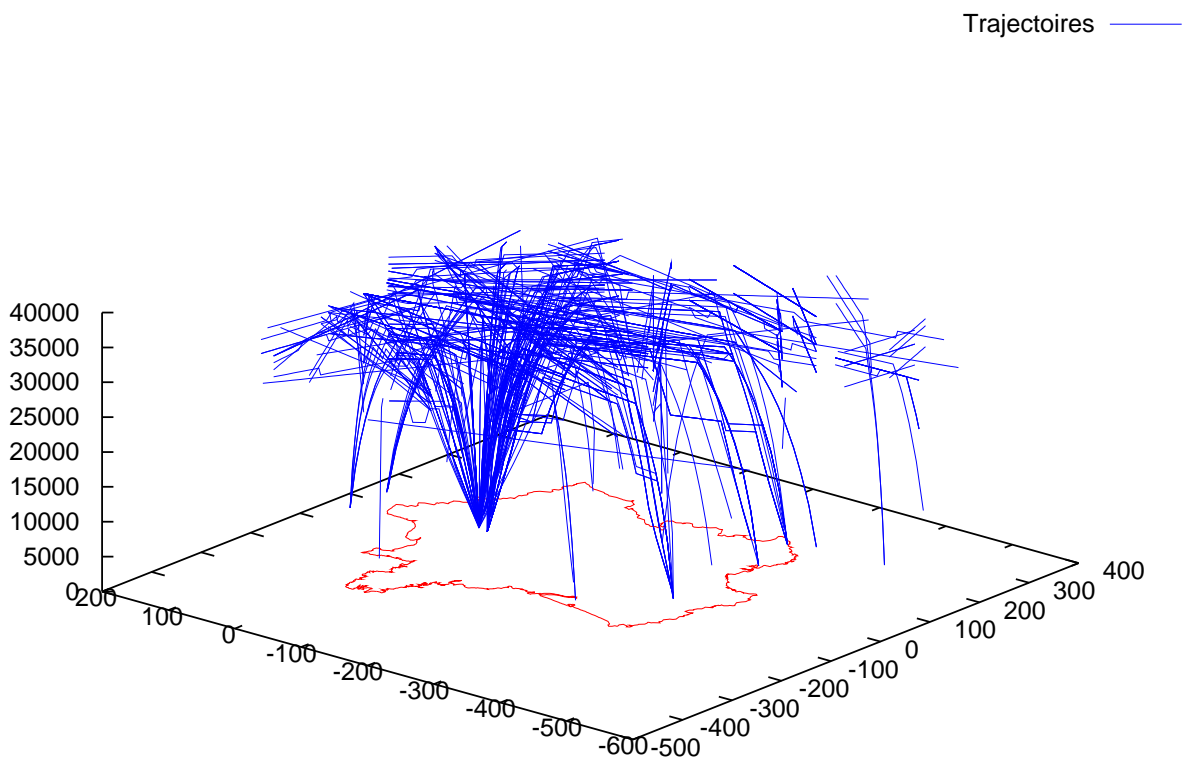


FIG. 9.1 – Traces des trajectoires 4D trouvées par l' A^* les trains de plus de 3 avions en France, le 21 juin 2002 (plan horizontal : projection stéréographique centrée sur Paris ; distances horizontales en milles nautiques ; altitudes en pieds).

Seuls quelques résultats préliminaires sont pour l'instant disponibles. Les algorithmes n'ont pas encore pu être testés de manière exhaustive.

trajectoires —

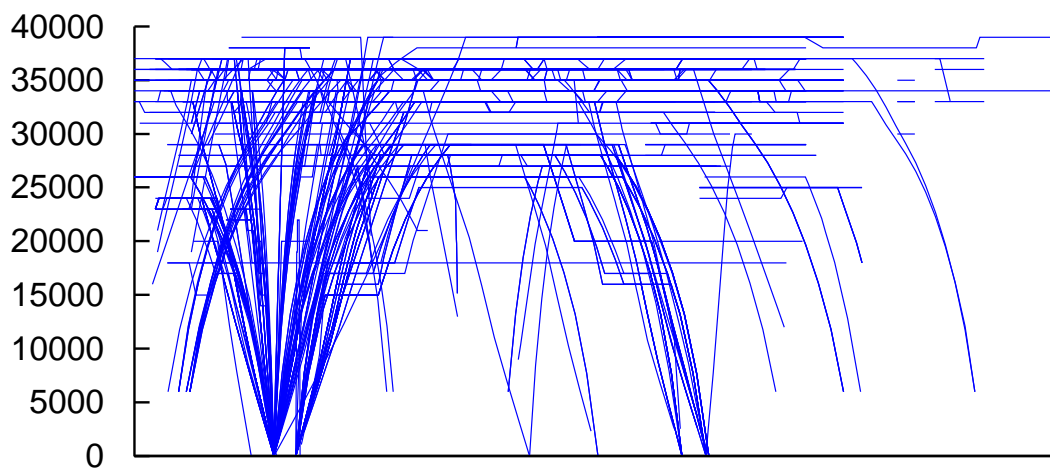


FIG. 9.2 – Vue de côté (Ouest) des trajectoires trouvées par l'A* les trains de plus de 3 avions en France, le 21 juin 2002 (altitudes en pieds).

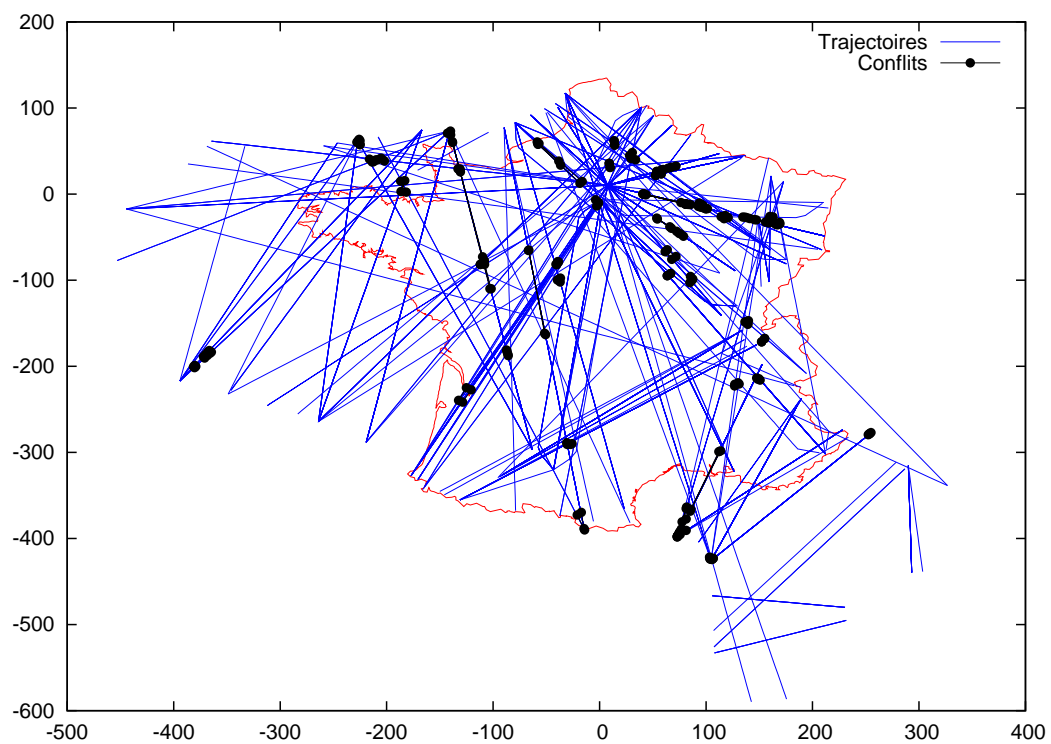


FIG. 9.3 – Trajectoires trouvées par l'A* les trains de plus de 3 avions en France, et conflits restants, le 21 juin 2002 (projection stéréographique centrée sur Paris ; distances en milles nautiques).

Sur les données France du 21 juin 2002, on trouve 446 trains de 3 avions ou plus, en agrégeant les vols selon les critères évoqués en 9.2. Ces trains d'avions regroupent 22,8 % du trafic total.

L'algorithme A^* , utilisé au dessus du FL145 avec les mêmes choix de paramètres qu'en 8.9.2 et avec le mode de détection entre zones d'incertitudes (cf. 8.7.4), fournit un résultat avec deux échecs sur les 446 trajectoires. Le coût cumulé des déviations de trajectoires (cf. 7.2.4, au chapitre 7) est de 1542.65917. Le temps de calcul est de 1 heure 47 minutes 34 secondes sur un CPU Xeon 2.8 GHz. La figure 9.1 montre la trace des trajectoires 4D trouvées par l' A^* . Une vue de côté (en se plaçant à l'Ouest) permet, sur la figure 9.2, de se représenter l'occupation de l'espace dans la dimension verticale.

La validation par une simulation de trafic montre que le nombre de conflits détectés au dessus du FL195 passe de 1582 à 1470, dont 135 en rattrapage à l'intérieur d'un même train d'avions. En faisant l'hypothèse que ces conflits soient réglés soit par les pilotes au moyen d'équipements de bord spécifiques soit par une régulation en vitesse, on obtient donc un nombre de conflits restants de 1335, soit une réduction de 16,6 % par rapport au trafic de référence. L'analyse des conflits restants montre qu'il reste 60 conflits entre avions appartenant à des trains d'avions différents, représentés sur la figure 9.3. Certains d'entre eux adviennent sur des segments de vol où la détection des interférences est inhibée (montées en provenance d'un même aéroport par exemple). Malgré tout, on constate sur la figure qu'il reste un certain nombre de conflits qui auraient dû être résolus (il y a notamment 22 conflits en rattrapage). Tous les problèmes ne sont donc pas réglés. Une explication possible peut venir des différences de modélisation entre le simulateur et l'algorithme que nous avons développé, et notamment pour tout ce qui a trait au calcul des temps.

9.6 Résultats pour les données Europe

Pour les données Europe du 21 juin 2002, il y a 599 trains de 3 avions ou plus, regroupant 7.3 % du trafic total. C'est donc une faible proportion du trafic qui est considérée, pour les raisons évoquées en section 8.3, et avec les mêmes palliatifs possibles (définition de flux entre TMAs de grandes taille) Avec les mêmes paramètres que précédemment, l'algorithme A^* trouve une solution sans interférences, avec un coût cumulé des déviations de 3482.05588. Les figures 9.4 et 9.5 montrent les traces 3D des trajectoires 4D trouvées par l'algorithme.

On constate un temps de calcul particulièrement élevé (plus de 24 heures), incompatible avec une éventuelle utilisation en temps réel de l'algorithme. Les temps de calcul individuels pour chaque trajectoire sont présentés sur la figure 9.6. Quelques trajectoires seulement sont à l'origine de ces temps de calcul importants. Si l'on considère les flux de 4 avions ou plus, le temps de calcul pour 190 trajectoires tombe à 1926.43 secondes, soit un peu plus de trente-deux minutes.

9.7 Conclusion sur la planification des trajectoires 4D

Nous sommes donc encore dans une phase exploratoire pour ce qui concerne la planification des trajectoires en quatre dimensions. Le modèle de trajectoire et la détection d'interférences développés au chapitre 8 ont été modifiés pour tenir compte des temps d'occupation de chaque segment de trajectoire par les avions. Ceux-ci sont regroupés en trains d'avions ayant des performances similaires, en fonction de l'heure d'entrée sur le flux origine-destination.

Les premiers résultats sur les données France et sur l'Europe semblent corrects. Il reste à examiner de près les causes des quelques conflits restants entre avions appartenant à des trains différents. Un autre point à régler est celui des temps de calcul, particulièrement longs sur les données Europe.

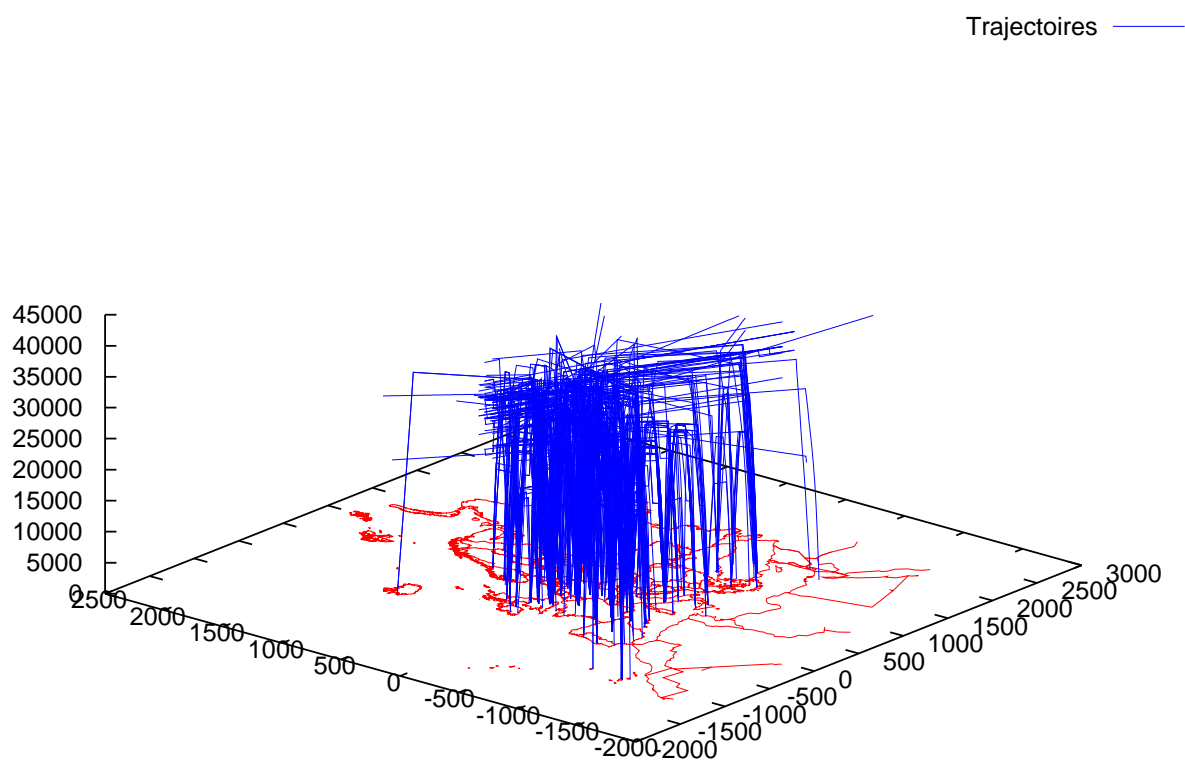


FIG. 9.4 – Traces des trajectoires 4D trouvées par l'A* les trains de plus de 3 avions en Europe, le 21 juin 2002 (plan horizontal : projection stéréographique centrée sur Paris ; distances horizontales en milles nautiques ; altitudes en pieds).

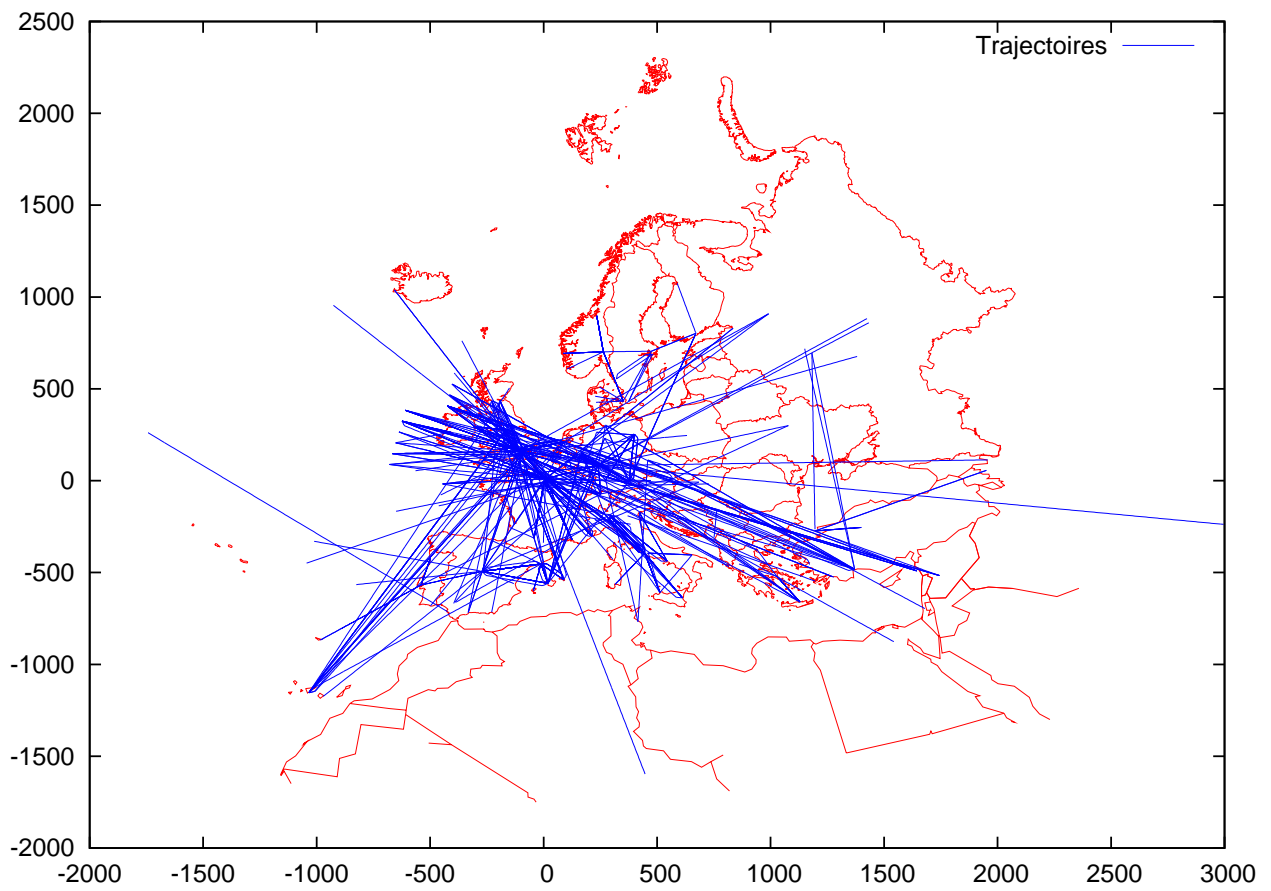


FIG. 9.5 – Vue dans le plan des traces des trajectoires 4D trouvées par l'A* les trains de plus de 3 avions en Europe, le 21 juin 2002 (projection stéréographique centrée sur Paris ; distances en milles nautiques).

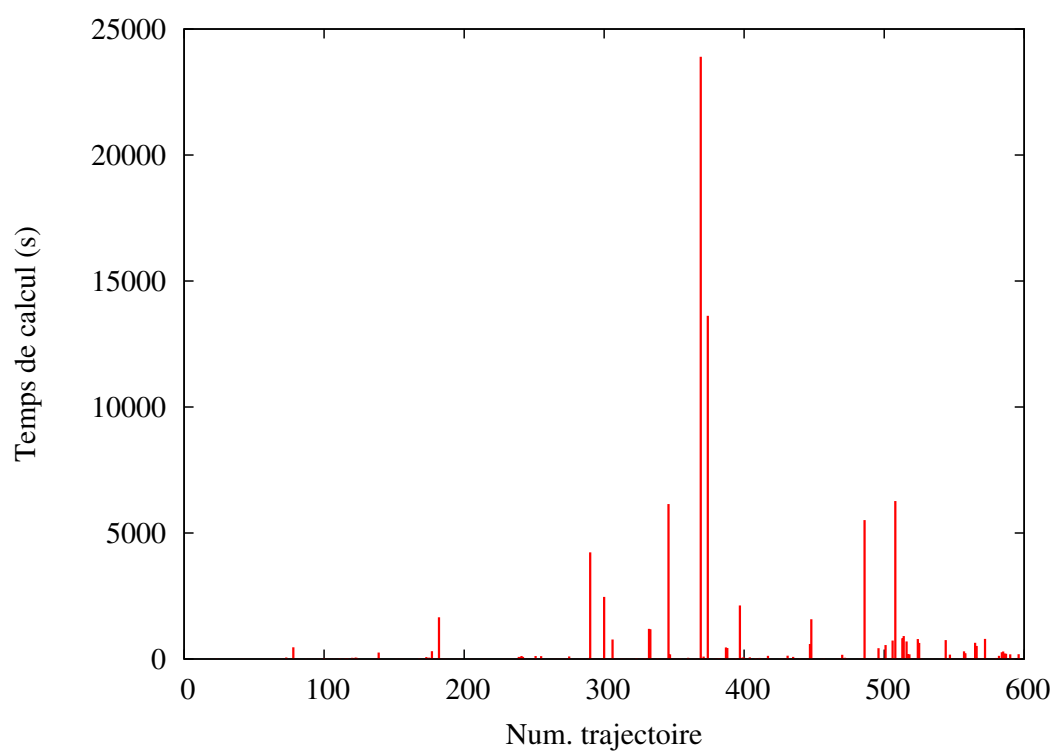


FIG. 9.6 – Temps de calcul des trajectoires 4D pour les trains de plus de 3 avions en Europe, le 21 juin 2002 (en abscisse : numéro de trajectoire ; en ordonnée : temps en seconde).

Parmi les solutions envisagées, on peut s'autoriser des trajectoires légèrement sous-optimales afin d'accélérer la recherche, à condition que les retours arrières dans l'arbre d'états soient principalement dus à la recherche d'un coût optimal, et non aux contraintes d'encombrement de l'espace. Si nous sommes dans le dernier cas, une autre solution possible serait de limiter l'éventail des choix possibles pour les niveaux de vol en palier, afin que l'algorithme détermine plus rapidement s'il y a ou non une solution pour la trajectoire considérée.

L'adaptation de l'algorithme génétique utilisé au chapitre 8 constitue à elle seule une autre voie à explorer. Le découpage en sous-problèmes, voire la parallélisation de l'algorithme, s'avèrent des préalables indispensables à une utilisation efficace sur le problème posé.

Chapitre 10

Conclusion et perspectives

10.1 Contributions

Cette thèse s'inscrit dans le domaine de l'optimisation globale appliquée aux flux de trafic aérien. Fondamentalement, l'amélioration de la gestion du trafic aérien peut être envisagée selon deux approches, que l'on peut éventuellement combiner. La première consiste à adapter la demande de trafic à la capacité du système de contrôle aérien. De nombreux travaux privilégiant cette voie ont été publiés, traitant notamment de la régulation du trafic (en allouant des créneaux de décollage par toute une variété de méthodes). La deuxième approche, choisie dans cette thèse, consiste à l'inverse à modifier le système ou la façon de le gérer, afin de traiter plus de trafic dans des conditions de sécurité identiques ou meilleures. D'un point de vue prospectif et en allant au-delà de l'amélioration empirique du système opérationnel, la deuxième approche ouvre un domaine de recherche encore peu exploré. Certes, comme nous l'avons vu aux chapitres 4, 5 et 6, il existe un certain nombre de travaux concernant la sectorisation, la construction de réseaux de routes 2D, ou l'allocation de niveaux de vols en croisière, sur la base du concept actuel (système centralisé, géré par des humains). Par ailleurs, on voit régulièrement émerger des concepts alternatifs de gestion du trafic (*free route*, *free flight*, *super-sector*, *sector-less*, évoqués dans le chapitre 2). Mais les travaux cités restent parcellaires, et les nouveaux concepts sont encore embryonnaires pour certains d'entre eux, et plus ou moins bien validés quant au gain attendu.

Dans cette thèse, le problème consistant à optimiser les flux de trafic aérien sans imposer de retards au décollage a été abordé en plusieurs étapes successives. Tout d'abord, nous avons considéré le système existant tel quel, et cherché à améliorer l'écoulement du trafic simplement en équilibrant les regroupements des secteurs élémentaires d'espace sur les positions de contrôle, de façon à répartir au mieux la charge de trafic entre celles-ci. Nous avons ensuite exploré différentes pistes correspondant à des degrés de liberté supplémentaires dans notre problématique : en premier lieu en jouant sur la géométrie du réseau de routes aériennes, ensuite en exploitant la possibilité de séparer verticalement les flux en croisière, puis en tenant compte de l'intégralité des profils de vol (montée, croisière, et descente) pour trouver des trajectoires 3D séparées pour les flux principaux, et enfin en introduisant la dimension temporelle, afin de planifier dynamiquement des trajectoires 4D non-conflictuelles pour des trains d'avions.

A chaque étape, nous avons cherché à employer les méthodes les plus adaptées à la nature et la taille du problème, ou éventuellement à la stratégie choisie (approche globale ou séquentielle par exemple). Pour les configurations de secteurs, le choix est guidé par l'aspect combinatoire du problème : les méthodes déterministes classiques de parcours d'arbre (*Branch & bound*, A^*) se révèlent

adéquates lorsqu'on opère dans un contexte réaliste, avec un nombre restreint de secteurs et de regroupements pré-définis. Dès lors que l'on augmente le nombre de secteurs ou que l'on s'autorise des regroupements quelconques, une méthode stochastique s'impose. Pour la construction d'un réseau planaire de routes, nous avons choisi d'explorer une voie légèrement différente de celle présentée dans [Meh00]. Nous avons montré qu'en utilisant une méthode d'agrégation par nuées dynamiques, puis une triangulation de Delaunay, il était possible de constituer un réseau selon un critère simple d'espacement des points de croisement. Ce critère permet de s'assurer qu'il reste bien autour de chaque point de croisement suffisamment d'espace pour permettre à des contrôleurs aériens de résoudre les conflits en déviant les trajectoires des avions. Les plus courts chemins à travers ce réseau, respectant une contrainte d'angle maximum entre deux segments consécutifs de routes, sont obtenus en utilisant un algorithme A^* . Dans [Meh00], cette contrainte d'angle était directement prise en compte dans l'algorithme d'agrégation. Le diagramme de Voronoï (dual de la triangulation de Delaunay) associé à ce réseau fournit un ensemble de mini-secteurs définis autour de chaque point de croisement. Le choix et l'implantation de l'algorithme d'allocation de niveaux de croisière n'ont pas été abordés dans cette thèse, où l'on a seulement cherché à valider l'allocation réalisée par l'algorithme de coloration de graphe développé par N. Barnier et P. Brisset ([BB02]). Pour la recherche de trajectoires 3D ou 4D séparées, l'algorithme A^* est tout à fait adapté lorsqu'on calcule les trajectoires séquentiellement, en évitant d'interférer avec les précédentes (stratégie 1 *vs.* n). Dans le cas d'une approche globale, où toutes les trajectoires sont considérées simultanément sans ordre préférentiel, nous avons vu que la taille du problème et le nombre de composantes connexes dans l'espace des solutions rendait impraticable l'utilisation de méthodes locales et/ou déterministes. En conséquence, nous avons choisi d'utiliser un algorithme génétique, pour lequel nous avons développé les opérateurs spécifiques de génération d'éléments, de croisement, de mutation, et d'évaluation de la *fitness*.

Nous avons pris soin lorsque c'était possible de valider les résultats obtenus et d'estimer le gain apporté par telle ou telle méthode. Ainsi, l'impact des regroupements de secteurs sur l'écoulement du trafic a été évalué en estimant les retards au décollage, au moyen d'un algorithme d'allocation de créneaux dont les données d'entrée sont le trafic et les capacités associées à chaque groupe de secteurs. La comparaison entre les schémas optimisés de regroupement et les schémas prévisionnels réellement déposés par le contrôle aérien donne pour le jour choisi une diminution de 69% des retards au décollage, en utilisant environ 20% de ressources en moins. Sur ce même sujet, nous avons par ailleurs démontré, par une rapide analyse statistique autour des instants de dégroupement, la non-pertinence des indicateurs actuellement utilisés dans l'élaboration des schémas d'ouverture de secteurs (flux entrants et capacités horaires). Nous avons donc à la fois démontré l'intérêt potentiel des algorithmes proposés et la nécessité de définir, préalablement à une utilisation opérationnelle, des indicateurs macroscopiques réalistes de la charge de travail du contrôleur. Les résultats concernant l'allocation de niveaux de croisière, la génération de trajectoires 3D séparées, et la planification de trajectoires 4D ont quant à eux été validés en considérant le nombre et la nature des conflits détectés. Le simulateur en temps accéléré CATS/OPAS a été modifié à cette fin.

La principale contribution de cette thèse, en dehors des résultats que nous venons d'évoquer, réside dans l'allocation de trajectoires 3D séparées pour les flux principaux. L'algorithme A^* et l'algorithme génétique testés sur des problèmes simples dans le chapitre 7 ont été appliqués dans le chapitre 8 à des données réelles, dans le contexte français, puis européen, en choisissant chaque route, soit parmi trois routes parallèles les plus directes possibles, soit parmi des itinéraires définis sur le réseau de routes opérationnel. La modélisation des trajectoires tient compte des performances réelles des avions. La disparité des performances des avions évoluant au sein d'un même flux a été prise en compte en définissant des zones d'incertitude autour de chaque trajectoire. Trois modes de détection des interférences ont été développés, basés pour la première sur une distance spécifique intégrant les normes

de séparation et, pour les deux autres, sur l'intersection de tubes dans l'espace, en prenant ou non en compte les incertitudes sur les profils. Différents modèles ont été proposés, permettant d'allouer une ou plusieurs trajectoires 3D par flux, en tenant compte des évolutions vers les aéroports proches de la frontière, ou de la disparité des niveaux d'entrée, de croisière, ou de sortie. Les simulations de trafic sur les données France, en faisant voler les avions sur les trajectoires 3D trouvées par les algorithmes, ont montré l'intérêt de la séparation des flux. La diminution du nombre de conflits détectés est d'environ trente pour cent en imposant des trajectoires non-interférentes à environ quarante pour cent du trafic, et peut monter jusqu'à près de quarante pour cent de diminution des conflits en traitant un peu plus de la moitié du trafic. Au-delà, les performances des algorithmes sont moins bonnes. Précisons que le gain en nombre de conflits est calculé en faisant l'hypothèse que les conflits entre avions suivant une même trajectoire 3D sont réglés au départ, par un séquençage temporel et une régulation en vitesse.

La planification de trajectoires 4D non-conflictuelles, décrite au chapitre 9, est également une contribution novatrice. Les flux de trafic sont dynamiquement agrégés en *trains d'avions* en fonction des niveaux d'entrée, de croisière, et de sortie demandés, ainsi que de l'heure d'entrée. L'algorithme A^* du chapitre 8 a été adapté afin de calculer séquentiellement pour chaque train d'avion une trajectoire 4D séparée des précédentes. A cet effet, la dimension temporelle a été introduite dans les différents modes de détection d'interférences présentés au chapitre 8.

10.2 Perspectives de recherche

Les voies de recherche restant à explorer sont nombreuses. Concernant les regroupements de secteurs, la détermination des bons indicateurs macroscopiques de la charge de travail du contrôleur (et des valeurs maximales associées) constitue en soi un sujet à part entière. On peut trouver dans la littérature une multitude d'indicateurs, qui restent encore à valider selon une méthode objective. Nous avons proposé pour cela d'étudier le comportement des indicateurs choisis autour des instants où les secteurs de contrôle sont dégroupés. Mais l'analyse statistique sommaire effectuée au chapitre 4 sur quelques indicateurs de base ne saurait suffire, et il faudrait faire une véritable analyse discriminante sur un ensemble plus vaste d'indicateurs.

Pour ce qui est de la construction d'un réseau de routes aériennes, il reste entre autres à vérifier si le réseau obtenu selon le critère d'espacement des points de croisement apporte un gain par rapport au réseau existant. Par ailleurs, on ne peut se contenter de la sectorisation fournie par le diagramme de Voronoï : les secteurs associés à chaque point de croisement sont trop petits. Il faudrait donc soit augmenter la distance minimale entre points de croisement, ce qui risque de rallonger les parcours dans le réseau, soit regrouper les mini-secteurs de façon équilibrée, tout en tenant compte d'un critère de convexité au sens des flux (un flux entre deux secteurs adjacents ne doit pas traverser un troisième secteur). Pour cela, on peut envisager d'adapter les algorithmes proposés au chapitre 4, ou d'utiliser des techniques de programmation par contraintes.

Pour la définition d'un réseau statique de trajectoires 3D, nous n'avons tenu compte jusqu'à présent que des flux entre aéroports (ou points d'entrée/sortie). Nous avons vu qu'il serait préférable, afin de traiter plus de trafic dans le contexte européen, de regrouper les aéroports en TMA de grande taille, et d'appliquer les algorithmes proposés aux flux définis entre les principales TMA. Par ailleurs, il reste à améliorer les performances de l'algorithme génétique en l'appliquant en parallèle sur des sous-problèmes séparés lorsque c'est possible (ce n'est utile que sur les données Europe).

Concernant la planification des trajectoires 4D, on peut envisager d'utiliser un algorithme génétique dérivé de celui du chapitre 8, sachant qu'il faut d'abord gérer le découpage en sous-problèmes de

plus petite taille, en fonction du graphe d'interférences entre trajectoires. Par ailleurs, il reste encore à tester exhaustivement les algorithmes, et à valider les résultats en termes de diminution du nombre de conflits.

10.3 Perspectives d'utilisation opérationnelle

Pour conclure, voici quelques éléments de réponse à l'habituelle question que se pose souvent, et parfois à juste titre, le milieu professionnel face aux résultats d'une démarche de recherche, même appliquée : à quoi cela va-t-il pouvoir servir ?

Dans une perspective à court terme, les méthodes et algorithmes proposés pour l'optimisation des regroupements de secteurs pourraient être rapidement mis en oeuvre pour élaborer les schémas prévisionnels d'ouverture de secteurs, à condition de définir des indicateurs macroscopiques valables pour la charge de travail du contrôleur, autres que ceux utilisés actuellement (flux horaires et capacités) et dont nous avons vu qu'ils n'étaient pas pertinents dans ce cadre. Il est également envisageable de les utiliser dans un outil d'aide à la décision pour les chefs de salle de contrôle, en prenant en compte des indicateurs de charge encore plus précis et mis à jour en temps réel.

Les méthodes de construction d'un réseau de routes et d'une mini-sectorisation associée, présentées au chapitre 5 méritent une plus ample exploration avant de pouvoir être exploitées pour la définition stratégique du réseau.

Les algorithmes permettant de définir un réseau statique de trajectoires 3D séparées ont atteint un assez bon niveau de réalisme, que l'on doit encore améliorer en tenant compte par exemple des différences d'altitudes entre aéroports. Pour réaliser un réseau pertinent, il faudrait définir ces trajectoires en prenant en compte le trafic sur plusieurs mois, et non plus sur une seule journée. Comme on peut utiliser les algorithmes en conservant le réseau de routes standard, il est tout à fait possible d'envisager une mise en place progressive de ces trajectoires 3D. Transformer un itinéraire standard en trajectoire 3D est assez simple : il suffit de publier une liste de points spécifiques le long de la route, et de niveaux de vols. Chaque pilote saura alors qu'en atteignant tel point, il doit commencer une évolution verticale vers tel niveau. Cette mise en place de contraintes sur les évolutions verticales des avions devrait toutefois s'accompagner d'une étude préalable d'impact sur la charge des secteurs de contrôle traversés par l'itinéraire concerné.

Dans une perspective à moyen terme, les algorithmes permettant de trouver des trajectoires 4D non conflictuelles pourraient trouver leur place dans des outils de planification tactique des vols. Chaque trajectoire 4D serait définie sur toute la traversée d'un volume d'espace assez grand (l'espace aérien français par exemple), en prévoyant des incertitudes temporelles suffisamment grandes. L'objectif ne serait pas alors de résoudre la totalité des conflits, mais de chercher à en éliminer un certain nombre en planifiant la trajectoire de chaque avion à travers l'espace considéré.

Enfin dans une perspective plus futuriste, et dans l'hypothèse d'un changement plus radical des méthodes de contrôle, la séparation des trajectoires 3D ou 4D pourrait trouver son application dans un contexte *super-sector* ou *sector-less*. Dans un tel contexte, le rôle d'un contrôleur aérien serait radicalement différent de la pratique actuelle : plutôt que de contrôler les avions à l'intérieur d'un volume d'espace sous sa responsabilité, le contrôleur aurait à guider un ou plusieurs avions sur la totalité du vol (ou une grande partie), tout en évitant le reste du trafic. En organisant le trafic au niveau stratégique de façon à éliminer les interférences entre les flux principaux, la tâche des contrôleurs serait grandement facilitée. Les algorithmes calculant les trajectoires 4D seraient quant à eux tout à fait adaptés à l'aide tactique au contrôleur, dans la réalisation de missions consistant à faire traverser un espace assez vaste à des trains d'avions.

Glossaire

ACAS : *Airborne Collision Avoidance System* cf TCAS.

ACC : *Air Control Center* Centre de contrôle du trafic aérien.

Airway : Route aérienne suivie par les avions en croisière.

AMAN : *Arrival Manager*. Système diffusant les informations sur les vols à l'arrivée d'un aéroport pour préparer leur séquençement.

ATC : *Air Traffic Control*. Système de contrôle du trafic aérien. Les organismes ATC sont chargés de la gestion tactique des vols traversant leur espace.

ATCC : *Air Traffic Control Center* Même chose que ACC, c'est-à-dire un centre de contrôle du trafic aérien.

ATFM : *Air Traffic Flow Management*. Terme désignant la gestion des flux de trafic aérien. La CFMU est un organisme ATFM gérant les flux au niveau de l'Europe.

ATM : *Air Traffic Management*. Gestion du trafic aérien.

CAUTRA : *Coordonnateur AUtomatique du TRafic Aérien*. Système assurant les traitements et la diffusion des plans de vol aux contrôleurs.

CENA : *Centre d'Etudes de la navigation Aérienne*.

CFL : *Cleared Flight Level* Niveau de vol (cf FL) autorisé par le contrôle aérien.

CFMU : *Central Flow Management Unit*. Cellule européenne de régulation des flux de trafic. Son rôle est de s'assurer que le trafic n'excède pas les capacités, en retardant le décollage de certains avions ou en leur faisant emprunter des routes alternatives.

Capacité : Une valeur seuil du flux de trafic entrant dans un secteur ou un groupe de secteurs, au-delà de laquelle celui-ci est considéré comme surchargé.

Clairance : *ou clearance* Autorisation donnée par le contrôle à un pilote, par exemple pour le roulage, le décollage, l'atterrissage, l'entrée dans un espace contrôlé, ou un changement de niveau de vol.

Configuration de secteurs : répartition de n secteurs sur k positions de contrôle.

Conflit : On parle de conflit entre deux avions lorsque ceux-ci ne sont plus séparés (cf l'entrée *séparation* de ce glossaire), ou risquent de ne plus l'être dans un futur proche (conflit potentiel).

COURAGE : *Calcul Optimisé des Uceso et des Régulations pour l'Amélioration de la Gestion de l'Espace*. Système opérationnel facilitant l'adaptation du dispositif de contrôle dans la phase pré-tactique et en temps réel.

Data-Link : *Liaison de données*. Systèmes de transmission automatique des données entre le sol et les avions ou entre avions.

Dégrouper : Plusieurs secteurs aériens peuvent être regroupés sur une même position, en un seul secteur de contrôle. Par abus de langage, on utilise le terme de “dégrouper” lorsqu’on coupe un tel secteur en plusieurs secteurs, en armant une ou plusieurs nouvelles positions de contrôle, auxquelles on affecte un certain nombre de secteurs élémentaires du secteur initial.

DGAC : *Direction Générale de l’Aviation Civile.*

DMAN : *Departure Manager.* Système diffusant les informations sur les vols au départ d’un aéroport vers les espaces qu’il va traverser.

EDIT : *Ecole Doctorale d’Informatique et Télécommunications.*

ENAC : *Ecole Nationale de l’Aviation Civile.*

EUROCONTROL : *European Organisation for the Safety of Air Navigation.* Organisme Européen chargé de la gestion du trafic aérien en Europe.

FAA : *Federal Aviation Administration.* Organisme de l’aviation civile américaine.

FL : *Flight level* Le niveau de vol ou altitude-pressure est l’altitude correspondant à la différence de pression entre la position occupée et la surface isobare 1013,25 hPa exprimée en centaines de pieds (ft). On le note FLhhh. Par exemple, un avion au FL330 vole à l’altitude-pressure de 33000 ft au dessus de l’isobare 1013,25 hPa.

FMP : *Flow management position* Dans chaque centre de contrôle, une cellule de gestion des flux travaille en collaboration avec la CFMU, et élabore entre autres les schémas d’ouvertures de secteurs.

FMS : *Flight Management System* Système de gestion des automatismes de l’avion, dont le pilotage automatique.

ft : *feet* 1 pied vaut 30,48 cm.

GPS : *Global Positionning System.* Système de positionnement par satellite provenant du département de la défense américaine.

HUD : *Head-up display* Les systèmes HUD (*Head Up Display*) ou “tête haute” sont des aides au pilotage où les informations nécessaires sont projetées directement sur la vitre du cockpit par un système optique. La principale innovation est cependant de proposer un pilotage “à l’énergie totale” d’une précision et d’une simplicité remarquables. Ces systèmes sont principalement développés sur les avions d’armes, et font leur timide apparition dans le civil, surtout outre-atlantique. L’inventeur du système est un ingénieur français, Mr Klopstein, qui dès les années soixante expérimentait ce principe de pilotage avec un collimateur de vol électronique.

LEEA : *Laboratoire d’Économie et d’Économétrie de l’Aérien.* Structure de recherche commune à l’ENAC et au CENA.

LOG : *Laboratoire d’Optimisation Globale.* Structure de recherche commune à l’ENAC et au CENA.

Mille nautique : cf NM

Niveau de vol : cf entrée FL de ce glossaire

NM : *Nautical mile* 1 mille nautique vaut 1852 mètres (1 minute d’angle sur un méridien).

Plan de vol : Ensemble d’informations décrivant les intentions de vol (aéroports de départ et de destination, route souhaitée et niveau de vol préférentiel, etc...). Le plan de vol est initialement fourni aux organismes de contrôle aérien par la compagnie aérienne, ou par le pilote. Il est ensuite enrichi (allocations de créneaux de décollage, mises à jour des heures, etc...) par les systèmes de traitement des plans de vols, avant d’être archivé.

- Position de contrôle** : Un meuble muni d'équipements de communication et de surveillance (radar, radio, téléphone...), armé par une équipe de deux contrôleurs qui assurent la séparation entre les avions dans le(s) secteur(s) qu'ils contrôlent.
- RFL** : *Requested Flight Level* Niveau de vol (cf FL) préférentiel, auquel la compagnie aérienne (ou le pilote) souhaiterait faire voler l'avion. Cette information figure dans le plan de vol.
- RFM** : Division *Routes, Flux, et Modélisation* du CENA.
- RVSM** : *Reduced Vertical Separation Minima* Dispositif autorisant des séparations verticales de 1000 pieds au dessus du FL295, au lieu des 2000 pieds imposés précédemment
- Schéma d'ouvertures** : Une estimation des configurations de secteurs qui seront utilisés le jour suivant (ou celui d'après). Cette estimation tient compte du trafic prévu et des ressources disponibles (contrôleurs et positions de contrôle).
- Secteur de contrôle** : L'espace aérien est divisé en petits volumes appelés *secteurs*, contrôlés chacun par une équipe de deux contrôleurs. Une équipe peut avoir sous sa responsabilité plusieurs secteurs regroupés.
- Séparation** : On définit une distance horizontale, exprimée en milles nautiques¹ (NM), la *séparation horizontale*, ainsi qu'une distance verticale exprimée en pieds² (ft) : la *séparation verticale*. On dit que deux avions sont *séparés* quand la distance qui les sépare projetée dans le plan horizontal est supérieure à la séparation horizontale, OU quand la différence de leurs altitudes est supérieure à la séparation verticale. Les valeurs courantes des séparations horizontales et verticales varient en fonction de l'environnement technique, et de la nature de l'espace dans lequel se trouvent les avions. A titre indicatif, les normes de *séparations radar* (lorsqu'on dispose de matériel de visualisation radar adéquat, de procédures de contrôle associées, dans des espaces où les avions doivent être équipés de transpondeurs, etc) usuellement appliquées dans l'espace aérien européen en-route sont de 5 NM horizontalement, et de 1000 ft verticalement au-dessus du FL 290 ou de 2000 ft au dessus. Toutefois, dans les espaces dits RVSM (Reduced Vertical Separation Minima) qui tendent à se généraliser, la séparation verticale au dessus du FL 290 est de 1000 ft.
- STPV** : *Système de Traitement des Plans de Vol*
- Strips** : Supports papier de format prédéfini permettant le suivi d'un avion par les contrôleurs.
- TCAS** : *Traffic alert and Collision Avoidance System*. Système embarqué détectant à très court terme les pertes de séparations en vol et donnant un conseil de descente ou de montée de dernier recours aux pilotes.
- TMA** : *Terminal Area* Il s'agit d'un volume d'espace défini autour d'un aéroport important, et éventuellement de ses aéroports satellites, et permettant de gérer le trafic en approche ou au départ de ce ou ces aéroports.
- Vitesse air** : Vitesse de l'avion par rapport à la masse d'air dans laquelle il se déplace.

¹ 1 mille nautique vaut 1852 m (1 minute d'angle sur un méridien)

² 1 pied vaut 30,48 cm

Bibliographie

- [ASBG02] J.M. Alliot, T. Schiex, P. Brisset, and F. Garcia. *Intelligence Artificielle et Informatique Théorique, 2ème édition*. Cepadues, 2002. ISBN : 2-85428.578.6. *Cité p. 37*
- [Bar02] N. Barnier. *Application de la programmation par contraintes à des problèmes de gestion du trafic aérien*. PhD thesis, Institut National Polytechnique de Toulouse, 2002. *Cité p. 8*
- [BB02] N. Barnier and P. Brisset. Graph coloring for air traffic flow management. In *CPAIOR'02 : Fourth International Workshop on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimisation Problems*, pages 133–147, Le Croisic, France, March 2002. *Cité p. 8, 9, 85, 86, 87, 184*
- [BBR01] N. Barnier, P. Brisset, and T. Rivière. Slot allocation with constraint programming : Models and results. In *Proceedings of the fourth USA/Europe Air Traffic Management R&D Seminar*, 2001. *Cité p. 58, 85*
- [BFM00a] T. Baeck, D.B. Fogel, and Z. Michalewicz. *Evolutionary Computation 1 : Basic Algorithms and Operators*. Institute of Physics Publishing, 2000. *Cité p. 36*
- [BFM00b] T. Baeck, D.B. Fogel, and Z. Michalewicz. *Evolutionary Computation 2 : Advanced Algorithms and Operators*. Institute of Physics Publishing, 2000. *Cité p. 36*
- [Bos97] J.F. Bosc. *Techniques d'évitement réactif et simulation du trafic aérien*. PhD thesis, Institut National Polytechnique de Toulouse, 1997. *Cité p. 8*
- [BP00] D. Bertsimas and S. Stock Patterson. The traffic flow management rerouting problem in air traffic control : A dynamic network flow approach. *Transportation Science*, 34(3) :239–255, August 2000. *Cité p. 85*
- [Bré79] D. Brélaz. New methods to color the vertices of a graph. *Communications of the ACM*, 22 :251–256, 1979. *Cité p. 86*
- [CDG⁺89] G. Celeux, E. Diday, G. Govaert, Y. Lechevallier, and H. Ralambondrainy. *Classification automatique des données : environnement statistique et informatique*. Dunod, 1989. *Cité p. 21, 23*
- [Cer91] R. Cerf. Asymptotic convergence of genetic algorithms. *Mathematics Subject Classification. 60F10, 60J10, 92D15.*, 1991. *Cité p. 20*
- [Com00] Performance Review Commission. Performance review report (covering calendar year 1999). Technical report, Eurocontrol, 2000. *Cité p. 7, 12*
- [Com01] Performance Review Commission. Performance review report, an assessment of air traffic management in europe during the calendar year 2000. Technical report, Eurocontrol, 2001. *Cité p. 7, 12*

- [Cou97] O. Coudert. Exact coloring of real-life graphs is easy. In *Design Automation Conference DAC97*, Anaheim, California, June 1997. *Cité p.* 87
- [DA98] N. Durand and J. M. Alliot. Genetic crossover operator for partially separable functions. In *Proceedings of the third annual Genetic Programming Conference*, 1998. *Cité p.* 55
- [DAB97] N. Durand, J.M. Alliot, and J.F. Bosc. Cats, a complete air traffic simulator. In *Proceedings of DASC97*, 1997. *Cité p.* 86, 89
- [DABM97] N. Durand, J.M. Alliot, J.F. Bosc, and L. Maugis. An experimental study of atm capacity. In *Proceedings of Europe-USA conference on Air Traffic Management*, 1997. *Cité p.* 86, 89
- [DAG01] N. Durand, J.M. Alliot, and G. Granger. Optimal resolution of en route conflicts. In *Proceedings of ATM2001*, 2001. *Cité p.* 86, 89
- [DASF95] D. Delahaye, J.M. Alliot, M. Schoenauer, and J.L. Farges. Genetic algorithms for automatic regroupement of air traffic control sectors. In *Proceedings of the Conference on Evolutionary Programming*, 1995. *Cité p.* 8, 49, 50
- [dBvKOS00] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry : Algorithms and Applications*. Springer-Verlag, 2000. ISBN? : 3-54065-620-0. *Cité p.* 26, 27
- [Del95] D. Delahaye. *Optimisation de la sectorisation de l'espace aérien par algorithmes génétiques*. PhD thesis, Ecole Nationale Supérieure de l'Aéronautique et de l'Espace, 1995. *Cité p.* 8
- [DGNS01] V. Duong, G. Gawinowski, J.P. Nicolaon, and D. Smith. Sector-less air traffic management. In *Proceedings of the fourth USA/Europe Air Traffic Management R&D Seminar*, 2001. *Cité p.* 8, 13, 17
- [DO97] D. Delahaye and A. Odoni. Airspace congestion smoothing by stochastic optimization. In *Evolutionary Programming VI*, 1997. *Cité p.* 85
- [DPST03] J. Dréo, A. Pétrowski, P. Siarry, and E. Taillard. *Métaheuristiques pour l'optimisation difficile*. Eyrolles, 2003. ISBN : 2-212-11368-4. *Cité p.* 20, 36
- [Dur96] N. Durand. *Optimisation de trajectoires pour la résolution de conflits aériens en route*. PhD thesis, Institut National Polytechnique de Toulouse, 1996. *Cité p.* 8, 36, 38, 42, 106
- [ES03] A.E. Eiben and J.E. Smith. *Introduction to Evolutionary Computing*. Springer, 2003. ISBN : 3-540-40184-9. *Cité p.* 36
- [Fab92] Y. Le Fablec. *Optimisation par algorithmes génétiques parallèles et multi-objectifs*. Master's thesis, Ecole Nationale de l'Aviation Civile (ENAC), 1992. *Cité p.* 36
- [Fab99] Y. Le Fablec. *Prévision de trajectoires d'avions par réseaux de neurones*. PhD thesis, Institut National Polytechnique de Toulouse, 1999. *Cité p.* 8
- [For65] E. Forgy. Cluster analysis of multivariate data : Efficiency vs. interpretability of classifications. In *Biometrics*, 1965. *Cité p.* 23
- [For92] S. Fortune. Voronoi diagrams and delaunay triangulations. In *Computing in Euclidean Geometry*, 1992. *Cité p.* 25, 26
- [For95] S. Fortune. Voronoi diagrams and delaunay triangulations. In *Computing in Euclidean Geometry*, 1995. *Cité p.* 25

- [fre95a] Free flight case study. Technical report, MIT/NASA/AATT Modeling Project, 1995. *Cité p. 8, 13*
- [Fre95b] Report of the rtca board of directors select committee on free flight. Technical report, RTCA, 1995. *Cité p. 8, 13*
- [Fre01] The eight-states free route airspace project : Free route airspace concept (frac). draft version 1.4. Technical report, EUROCONTROL, July 2001. *Cité p. 8, 13*
- [Fre02] Eight-states free route airspace project : Feasibility assessment report. Technical report, EUROCONTROL. Version 1.0, February 2002. *Cité p. 8, 13*
- [GA02] D. Gianazza and J. M. Alliot. Optimization of air traffic control sector configurations using tree search methods and genetic algorithms. In *Proceedings of the 21st Digital Avionics Systems Conference*, 2002. *Cité p. 10*
- [GAG02] D. Gianazza, J. M. Alliot, and G. Granger. Optimal combinations of air traffic control sectors using classical and stochastic methods. In *Proceedings of the 2002 International Conference on Artificial Intelligence*, 2002. *Cité p. 10*
- [GD03] J.B. Gotteland and N. Durand. Genetic algorithms applied to airport ground traffic optimization. In *Proceedings of the 2003 Congress on Evolutionary Computation*. CEC, 2003. *Cité p. 8*
- [GD04] D. Gianazza and N. Durand. Separating air traffic flows by allocating 3d-trajectories. In *Proceedings of the 23st Digital Avionics Systems Conference*, 2004. *Cité p. 10*
- [GDA04] D. Gianazza, N. Durand, and N. Archambault. Allocating 3d-trajectories to air traffic flows, using a^* and genetic algorithms. In *Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation*, 2004. *Cité p. 10*
- [Gia02] D. Gianazza. Optimisation des schémas d'ouverture des secteurs de contrôle aérien par méthodes classiques et algorithmes génétiques. In *Colloque des doctorants de l'Ecole Doctorale Informatique et Télécommunications de Toulouse*, 2002. *Cité p. 10*
- [Gia03] D. Gianazza. Optimisation des flux de trafic aérien : allocation de niveaux de vols et définition de trajectoires 3d. In *Colloque des doctorants de l'Ecole Doctorale Informatique et Télécommunications de Toulouse*, 2003. *Cité p. 10*
- [GNG⁺03] G. Gawinowski, J. Nobel, J.Y. Grau, D. Dothy, L. Guichard, J.P. Nicolaon, and V. Duong. Operational concepts for supersector. In *Proceedings of the fifth USA/Europe Air Traffic Management R&D Seminar*, 2003. *Cité p. 8, 13*
- [Gol89a] D. Goldberg. *Genetic Algorithms*. Addison Wesley, 1989. ISBN : 0-201-15767-5. *Cité p. 36, 37, 54*
- [Gol89b] D. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley, 1989. *Cité p. 36*
- [Got04] J.B. Gotteland. *Optimisation du trafic au sol sur les grands aéroports*. PhD thesis, Institut National Polytechnique de Toulouse, 2004. *Cité p. 8*
- [Gra02] G. Granger. *Détection et résolution de conflits aériens : modélisations et analyse*. PhD thesis, Ecole Polytechnique, 2002. *Cité p. 8, 16*
- [Hol75] J.H Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan press, 1975. *Cité p. 36*
- [Jar03] M.R. Jardin. Real-time conflict-free trajectory optimization. In *Proceedings of the fifth USA/Europe Air Traffic Management R&D Seminar*, June 2003. *Cité p. 85*

- [KF02] B. Kirwan and M. Flynn. Towards a controller-based conflict resolution tool - a literature review. Technical report, Eurocontrol, March 2002. ASA.01.CORA.2.DEL04-A.LIT. *Cité p.* 16
- [KM98] S. Koziel and Z. Michalewicz. A decoder-based evolutionary algorithm for constrained parameter optimization problems. In *Parallel Problem Solving from Nature*, 1998. *Cité p.* 44
- [Lei79] F. T. Leighton. A graph colouring algorithm for large scheduling problems. *Journal of Research of the National Bureau of Standards*, 84(6) :489–503, 1979. *Cité p.* 86
- [Let98] V. Letrouit. *Optimisation du réseau des routes aériennes en Europe*. PhD thesis, Institut National Polytechnique de Grenoble, 1998. *Cité p.* 8, 85
- [Mac67] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth International Symposium on Mathematical Statistics and Probability*, 1967. *Cité p.* 21, 23
- [Meh00] K. Mehadhebi. A methodology for the design of a route network. In *Proceedings of the Third Air Traffic Management R & D Seminar ATM-2000*, Napoli, Italy, June 2000. Eurocontrol & FAA. *Cité p.* 8, 69, 70, 83, 85, 184
- [MGZK98] L. Maugis, J.-B. Gotteland, R. Zanni, and P. Kerlirzin. TOSCA-II - WP3 : Assessment of the TMA to TMA hand-over concept. Technical Report TOSCA/SOF/WPR/3/03, SOFREA VIA, 1998. *Cité p.* 8, 85, 121, 124, 126, 171
- [Mic92] Z. Michalewicz. *Genetic algorithms+data structures=evolution programs*. Springer-Verlag, 1992. ISBN : 0-387-55387-. *Cité p.* 54
- [MS86] C. Morgenstern and H. Shapiro. Chromatic number approximation using simulated annealing. In *ACM Mountain Regional Meeting Proceedings*, 1986. *Cité p.* 86
- [Méd98] F. Médioni. *Méthodes d'optimisation pour l'évitement aérien : systèmes centralisés, systèmes embarqués*. PhD thesis, Ecole Polytechnique, 1998. *Cité p.* 8
- [OTA82] OTA. Airport and air traffic control system. Technical report, U.S. Congressional Office of Technology Assessment, 1982. *Cité p.* 16
- [Pea84] J. Pearl. *Heuristics*. Addison-Wesley, 1984. ISBN : 0-201-05594-5. *Cité p.* 33, 54
- [ReEHZ02] L. Rognin, I. Grimaud end E. Hoffman, and K. Zeghal. Impact of delegation of spacing tasks on safety issues. In *Proceedings of the 21st Digital Avionics Systems Conference on Air Traffic Management for Commercial and Military Systems*, 2002. *Cité p.* 17
- [Sap90] G. Saporta. *Probabilités, analyse des données et statistique*. Technip, 1990. *Cité p.* 23
- [SM96] M. Schoenauer and Z. Michalewicz. Evolutionary algorithms for constrained parameter optimization problems. In *Evolutionary Computation*, 1996. *Cité p.* 44
- [SM97] M. Schoenauer and Z. Michalewicz. Boundary operators for constrained parameter optimization problems. In *Proceedings of the Seventh International Conference on Genetic Algorithms*. ICGA, 1997. *Cité p.* 44
- [SM98] M. Schoenauer and Z. Michalewicz. Sphere operators and their applicability for constrained parameter optimization problems. In *Evolutionary Programming VII*, 1998. *Cité p.* 44
- [TBD02] H. TranDac, P. Baptiste, and V. Duong. A constraint programming formulation for dynamic airspace sectorization. In *Proceedings of the 21st Digital Avionics Systems Conference*, 2002. *Cité p.* 8

- [vKHHK95] C.H.M. van Kemenade, C.F.W. Hendriks, H.H. Hesselink, and J.N. Kok. Evolutionary computation in air traffic control planning. In *Proceedings of the Sixth International Conference on Genetic Algorithm*. ICGA, 1995. Cité p. 85
- [VM01] C. Verlhac and S. Manchon. Optimization of opening schemes. In *Proceedings of the fourth USA/Europe Air Traffic Management R&D Seminar*, 2001. Cité p. 8, 49
- [YG93] X. Yin and N. Germy. A fast genetic algorithm with sharing scheme using cluster analysis methods in multimodal function optimization. In C.R. Reeves R.F.Albrecht and N.C. Steele, editors, *In proceedings of the Artificial Neural Nets and Genetic Algorithm International Conference, Innsbruck Austria*. Springer-Verlag, 1993. Cité p. 41, 71
- [ZeEHR02] K. Zeghal, I. Grimaud end E. Hoffman, and L. Rognin. Delegation of spacing tasks from controllers to flight crew : impact on controller monitoring tasks. In *Proceedings of the 21st Digital Avionics Systems Conference on Air Traffic Management for Commercial and Military Systems*, 2002. Cité p. 17

OPTIMISATION DES FLUX DE TRAFIC AÉRIEN

Résumé : Cette thèse s'inscrit dans le domaine de l'optimisation globale appliquée aux flux de trafic aérien. Le problème abordé consiste à optimiser les flux de trafic aérien sans imposer de retards au décollage. On considère tout d'abord le système existant tel quel, en cherchant à améliorer l'écoulement du trafic simplement en équilibrant les regroupements des secteurs élémentaires d'espace sur les positions de contrôle. Des méthodes déterministes (A^* , Branch & bound) et un algorithme génétique sont utilisés pour répartir au mieux la charge de trafic entre les positions. Dans un deuxième temps on s'autorise à modifier la structure de l'espace aérien, en partant des flux directs origine-destination pour construire, par une méthode de partitionnement et une triangulation de Delaunay, un réseau de routes aériennes répondant à certains critères d'espacement des points de croisement. On évalue dans un troisième temps l'intérêt de séparer verticalement les flux aériens, dans leur phase de croisière. Cette évaluation porte sur le nombre et la nature des conflits détectés lors de simulations en temps accéléré, en allouant ou non des niveaux de croisières séparés. Dans un quatrième temps, on génère pour les principaux flux de trafic des trajectoires 3D complètes, séparées les unes des autres, en tenant compte de la disparité des performances des avions sur chaque flux. Deux types de stratégies sont explorées : une approche séquentielle où un algorithme A^* est appliqué successivement à chaque flux dans un ordre choisi, et une approche globale où toutes les trajectoires sont considérées simultanément, en utilisant un algorithme génétique. Les algorithmes sont d'abord testés sur des cas simples avant d'être appliqués aux données réelles, en France et en Europe. Enfin, en dernier lieu, la dimension temporelle est prise en compte afin de planifier dynamiquement des trajectoires 4D non-conflituelles pour des trains d'avions.

Mots clés : Trafic aérien, Optimisation, Algorithmes génétiques.

AIR TRAFFIC FLOWS OPTIMIZATION

Abstract : This work belongs to the field of global optimization, applied to air traffic flows. The problem being addressed consists of optimizing air traffic flows without regulating the traffic demand. Firstly, the current system is enhanced only by considering the sector configurations of the controllers working positions. Deterministic methods (A^* , Branch & bound) and a genetic algorithm are used to balance the workload between control positions, by splitting and merging airspace sectors. Secondly, we allow ourselves to modify the airspace structure. A routes network is computed from the direct origin-destination flows, with crossing points satisfying constraints of minimum distance, using a partitioning method and a Delaunay triangulation. Thirdly, the profit brought by the vertical separation of air traffic flows is assessed through fast-time simulations, by considering the nature of conflicts detected with or without a cruise level allocation. Fourthly, full 3D-trajectories are computed for the main flows, taking into account the variety of aircraft performances within each flow. Two strategies are proposed : the 1 vs. n strategy uses an A^* algorithm to compute each trajectory in turn, separating the new trajectory from the previous ones, and the global strategy applies a genetic algorithm to the whole set of trajectories. Both algorithms are first tried on basic flow configurations, and then applied to real traffic data over France and Europe. Finally, the time dimension is taken into account in order to generate conflict-free 4D-trajectories for groups of aircraft following the same route.

Key words : Air traffic, Optimization, Genetic algorithms



LABORATOIRE D'OPTIMISATION GLOBALE
CENTRE D'ÉTUDES DE LA NAVIGATION AÉRIENNE
ÉCOLE NATIONALE DE L'AVIATION CIVILE
7 avenue Édouard Belin - BP 54005
31055 Toulouse cedex 4

