

N° d'ordre :

THÈSE

présentée pour obtenir le titre de

DOCTEUR DE L'INSTITUT NATIONAL POLYTECHNIQUE DE TOULOUSE

Par : Jean-Baptiste GOTTELAND

École doctorale : Informatique et Télécommunications

Discipline : Informatique

Spécialité : Programmation et Systèmes

Laboratoire d'Optimisation Globale de l'École Nationale de l'Aviation Civile

OPTIMISATION DU TRAFIC AU SOL SUR LES GRANDS AÉROPORTS

Date de soutenance : le 2/11/2004, devant le jury composé de :

Mme.	Michèle SEBAG	Université Paris-Sud	(Rapporteur)
M.	Jin-Kao HAO	Université d'Angers	(Rapporteur)
M.	Vojin TOSIC	Université de Belgrade	(Membre du jury)
M.	Dominique COLIN DE VERDIÈRE	CENA	(Membre du jury)
M.	Jean-Marc GAROT	EUROCONTROL	(Membre du jury)
M.	Jean-Marc ALLIOT	CENA/LOG	(Directeur de thèse)
M.	Nicolas DURAND	CENA/LOG	(Directeur de recherche)

OPTIMISATION DU TRAFIC AU SOL SUR LES GRANDS AÉROPORTS

Résumé : *Une partie de plus en plus significative des retards aériens est imputable à la circulation des avions au sol sur les grands aéroports. Afin d'étudier les possibilités d'amélioration du trafic au roulage, un outil de simulation est développé et appliqué à Roissy Charles De Gaulle et Orly. La gestion du trafic est modélisée sous forme d'un problème de minimisation sous contraintes, sur lequel plusieurs méthodes d'optimisation sont comparées : une méthode déterministe, une méthode stochastique par algorithmes génétiques et une méthode hybride. L'outil de simulation résultant permet de mesurer l'influence de différents facteurs, comme l'horizon de prédiction, les incertitudes sur les vitesses de roulage ou encore l'application de sens uniques. La méthode hybride se révèle la plus efficace dans tous les scénarii envisagés et possède l'avantage d'être facilement adaptable à de nouveaux objectifs, comme le respect des créneaux de décollage imposés par la régulation européenne du trafic.*

Mots clés : Aéroport, Contrôle aérien, Optimisation, Algorithmes génétiques

GROUND TRAFIC OPTIMISATION IN BUSY AIRPORTS

Abstract : *Air traffic growth causes more and more significant congestion and ground delays on major airports. In order to study how this phenomenon can be improved, a ground traffic simulation tool is proposed and applied to Roissy Charles De Gaulle and Orly airports. Various optimization methods are developed to solve the problem linked with each ground traffic situation : a deterministic method, a stochastic method using a genetic algorithm and a hybrid method. The resulting tool allows to quantify the effect of several factors, as the size of the time window for traffic prediction, the speed uncertainties or the definition of some specific oneway taxiways. The hybrid method appears to be the most efficient in each situation and has the great advantage to be easily refined to handle correctly some new goals, like the application of the takeoff slots fixed by the European traffic flow management unit.*

Key words : Airport, Air traffic control, Optimization, Genetic algorithms



LABORATOIRE D'OPTIMISATION GLOBALE
ÉCOLE NATIONALE DE L'AVIATION CIVILE
7 avenue Édouard Belin - BP 54005
31055 Toulouse cedex 4



Remerciements

*Choisissez un travail que vous aimez
et vous n'aurez pas à travailler un seul jour de votre vie.*
(Confucius)

Un grand merci à tous ceux qui m'ont permis d'effectuer cette thèse, qui s'est révélée une expérience personnelle à la hauteur de ses promesses, à la fois passionnante et instructive. Ma gratitude s'adresse tout particulièrement aux membres du jury, qui ont accepté de juger mon travail : Michèle Sebag, directrice de recherche au CNRS, Jin-Kao Hao, professeur à l'Université d'Angers, Vojin Tosic, professeur à l'Université de Belgrade, Jean-Marc Garot, directeur du centre expérimental d'EUROCONTROL, Dominique Colin de Verdière, chef-adjoint du CENA, Jean-Marc Alliot, mon directeur de thèse, ainsi que Nicolas Durand, mon directeur de recherche.

L'étape initiale de cette longue aventure fut mon affectation au LOG et je n'oublierai pas la confiance et le soutien efficace que m'ont apportés, au cours de cette transition déterminante, Alain Printemps, chef du CENA, Nicolas Dubois, alors chef de département et Philippe Kerlizin, alors chef de la division RFM dont je salue tous les membres, sans oublier Yann Lefablec, qui a eu la bonne idée de finir sa thèse au moment précis où je pouvais commencer la mienne et qui a accepté un échange des rôles.

J'ai pu ensuite développer mes activités de recherche à partir des travaux de Brankica Pesic et grâce aux informations déterminantes fournies par la division ATT du CENA : je garde un excellent souvenir des réunions avec Erwan Page, chef de la division, qui a toujours su conserver une vision opérationnelle des problèmes abordés, Nicolas Marcou, qui a su prendre la suite en gardant les mêmes rapports humains, Isabelle Daguzon, sans qui je n'aurais jamais ouvert le manuel d'exploitation de Roissy, Caroline Aiglon, Philippe Amaury et Gérard Mavoian, qui ont été capables de m'éclaircir sur les points les plus obscurs de la description et du fonctionnement des aéroports. Merci d'être venus si régulièrement me chercher à Orly et pour tous ces moments agréables et productifs que nous avons partagés.

C'est aussi avec beaucoup d'enthousiasme que j'ai retrouvé l'ambiance chaleureuse et communicative du LOG : j'avais déjà pu apprécier, lors de mon DEA, les idées pertinentes de Jean-Marc Alliot associées aux conseils judicieux de Nicolas Durand, mais je ne connaissais pas encore les réponses avisées de Pascal Brisset, ni les idées de sorties aussi variées qu'agréables de Nicolas Barnier (à qui je ne regrette pas d'avoir offert du champagne, un soir matinal), ni l'allégresse et la serviabilité de Géraud Granger, ni même la vivacité (voire la dextérité) de Thomas Rivière. Je n'oublierai pas non plus ceux qui ont partagé mon bureau et qui ont su lui communiquer une convivialité unique : Franck Chatton (dont le poster de Walt Disney est resté au mur), Chiheb Kossentini (dont on retrouve encore des graphes de stratégies d'ouvertures pour Othello), Benoit Rulleau (qui ne contre jamais un 80), Nicolas Archambault (même s'il contre les 80 et a récemment déserté le bureau) et enfin Charles-Edmont Bichot (dont la dextérité va également étrangement croissante). S'ajoutent naturellement à

cette liste les membres du LEEA qui nous côtoient tous les jours et participent à la bonne humeur ambiante : Christian Bontemps (mon copilote préféré, parce qu'il n'a jamais peur, sauf des nuages), Nathalie Lenoir (ma copilote préférée, parce qu'elle va même dans les nuages), Kevin Guittet (qui a le génie de ne pas systématiquement gagner à la belote, ce qui n'est pas donné à tout le monde) et Karim Zbidi (qui m'a tant fait courir au badminton).

Cette expérience m'a également permis d'enseigner à l'ENAC et je dois beaucoup à la compétence d'Éric Blond (qui est également mon copilote préféré, pour ses approches parfaites sur Bordeaux) et à l'encadrement irréprochable de Catherine Ronfle-Nadaud (ma copilote préférée, parce qu'elle a un GPS). Je ne peux naturellement pas oublier Nicole Vaissière, sans qui un bon nombre de mes ordres de mission, déposés parfois en toute hâte, n'auraient jamais pu aboutir. Merci également à tous les membres du département MI de l'ENAC que j'ai rencontrés et que je vais continuer à côtoyer avec beaucoup de plaisir.

Enfin, je ne serai jamais assez reconnaissant envers ma famille et mes proches, pour l'indispensable soutien qu'ils m'ont apporté tout au long de ces années : merci à mes parents (pour tout, y compris une relecture active en plein mois d'août), à Julien et Anne-Chloé (pour leurs sujets de conversation inépuisables) ; merci à Anne Karczewski, qui a toujours su me reconforter et à ses parents, Henri et Alice, qui nous accueillent toujours si bien. Merci à Derf & Jeckle, pour accepter si souvent de partager nos WE, à Patrick & Magali, pour leur initiation aux mardis festifs, à Jérôme & Emmanuelle, pour leur art de la déco, à Arnaud & Sophie et Seb & Emma, pour leurs compétitions culinaires et à tant d'autres dont je n'ose citer les noms, mais qui se reconnaîtront !

J'oubliais : merci à Confucius, pour ses absurdités lumineuses.

Table des matières

Introduction	1
I Les problèmes aéroportuaires	3
1 Gestion du trafic aéroportuaire	5
1.1 Introduction	5
1.2 Contrôle du trafic aérien	5
1.2.1 Organisation générale	5
1.2.2 Capacité du système	6
1.2.3 Régulation du trafic	7
1.3 Le trafic aéroportuaire	7
1.3.1 Contrôle d'aéroport	7
1.3.2 Capacité d'un aéroport	8
1.3.3 Facteurs influençant la capacité aéroportuaire	9
1.4 Difficultés aéroportuaires identifiées	9
1.4.1 L'environnement	9
1.4.2 La sécurité	10
1.4.3 L'optimisation du trafic au sol	10
2 Analyse préliminaire	11
2.1 Introduction	11
2.2 Les principaux concepts pour l'optimisation du trafic au sol	11
2.2.1 Optimisation de la séquence de piste	11
2.2.2 Amélioration de la prévisibilité du trafic aéroportuaire	12
2.2.3 Systèmes évolués de surveillance et de guidage au sol (ASMGCS)	13
2.3 Les outils et les projets	14
2.3.1 Les outils opérationnels du contrôle d'aéroport	14
2.3.2 Quelques outils d'analyse	15
2.3.3 Les projets	15
2.4 Objectifs	17
2.4.1 Cadre de travail	17
2.4.2 Niveau de réalisme recherché	18
2.4.3 Résolution des situations	18

II	Modélisation	19
3	Modélisation de l'aéroport	21
3.1	Introduction	21
3.2	Vitesses nominales de déplacement	21
3.2.1	Méthodes de mesures	22
3.2.2	Résultats	24
3.2.3	Modèle retenu	24
3.3	Graphe des voies de circulation	26
3.3.1	Définitions et formalisation	26
3.3.2	Le graphe <i>direct</i>	27
3.3.3	Le graphe <i>dual</i>	28
3.3.4	Contraintes de validité des chemins	32
3.4	Conclusion	33
4	Modélisation du trafic	35
4.1	Introduction	35
4.2	Prédiction de trafic	35
4.2.1	Définition et paramètres de la prédiction	35
4.2.2	Effet de l'horizon	36
4.2.3	Modèle de trajectoire	37
4.2.4	Incertitudes sur la vitesse	39
4.3	Détection de conflits	41
4.3.1	Règles de séparation	41
4.3.2	Conflits entre avions	43
4.3.3	Gestion des incertitudes	43
4.3.4	Optimisation et complexité de la détection de conflits	45
4.4	Environnement de simulation	47
4.4.1	Itérations du système	47
4.4.2	Passage à la situation suivante avec incertitudes	47
4.5	Conclusion	48
5	Formulation du problème	49
5.1	Introduction	49
5.2	Le problème à résoudre	49
5.2.1	Variables	49
5.2.2	Contraintes	50
5.2.3	Critère d'optimisation	52
5.3	Analyse du problème	54
5.3.1	Complexité	54
5.3.2	Composantes connexes	54
5.3.3	Domaine d'admissibilité et optimum	55
5.4	Décomposition en problèmes indépendants	55
5.4.1	Principes	55
5.4.2	Clusters d'avions en conflit	55
5.4.3	Résolution itérative	56
5.5	Conclusion	56

III	Méthodes de résolution	57
6	Affectation de chemins	59
6.1	Introduction	59
6.2	Analyse préliminaire	59
6.2.1	Références	59
6.2.2	Notations principales	60
6.2.3	Coûts minimaux depuis un nœud	60
6.3	Méthode de la déviation maximale	61
6.3.1	Description de la méthode	61
6.3.2	Complexité	61
6.3.3	Résultats	62
6.4	Recherche des k meilleurs chemins	62
6.4.1	Formulation du problème	62
6.4.2	Relation de récurrence	64
6.4.3	Énumération récursive des k meilleurs chemins	65
6.4.4	Recherche de chemins sans boucle	67
6.4.5	Résultats	69
6.5	Recherche avec variation des coûts	71
6.5.1	Objectifs	71
6.5.2	Variation des coûts	72
6.5.3	Résultats	73
6.6	Conclusion	73
7	Résolution par des méthodes déterministes	75
7.1	Introduction	75
7.2	La méthode 1 contre n	75
7.2.1	Définition	75
7.2.2	Analyse du problème	76
7.2.3	Complexité	76
7.3	Classement fixe sans incertitude	78
7.3.1	Principes	78
7.3.2	Classement des avions	78
7.3.3	Optimisation de la trajectoire d'un avion	79
7.3.4	Application	81
7.4	Classement fixe avec incertitudes	83
7.4.1	Problèmes posés	83
7.4.2	Classement des avions	83
7.4.3	Optimisation de la trajectoire d'un avion	84
7.4.4	Application	85
7.5	Méthode avec variations du classement	87
7.5.1	Objectifs	87
7.5.2	Résolution par <i>Branch & Bound</i>	88
7.5.3	Recherche d'un classement optimal	90
7.6	Conclusion	95

8	Résolution par algorithmes génétiques	97
8.1	Introduction	97
8.2	Principes généraux	97
8.2.1	Présentation	97
8.2.2	Définition de la fonction d'adaptation et gestion des contraintes	100
8.2.3	Définition de la population	101
8.2.4	Renouvellement de la population	102
8.2.5	Convergence	104
8.3	Améliorations et adaptations classiques	105
8.3.1	Scaling	105
8.3.2	Sharing	106
8.3.3	Problèmes partiellement séparables	108
8.3.4	Optimisation locale en fin de convergence	110
8.4	Application au trafic au sol	111
8.4.1	Méthodologie	111
8.4.2	Définition de la population	111
8.4.3	Sélection	112
8.4.4	Caractère partiellement séparable du problème	113
8.4.5	Sharing	115
8.4.6	Terminaison de l'algorithme	116
8.4.7	Étalonnage de l'algorithme	116
8.4.8	Application à la simulation	119
8.5	Conclusion	122
9	Résolution par des méthodes hybrides	123
9.1	Introduction	123
9.2	Définition de méthodes hybrides	123
9.2.1	Recherche exacte et recherche approchée	123
9.2.2	Hybridation de méthodes exactes	124
9.2.3	Hybridation de méthodes approchées	125
9.3	Greffe d'un opérateur déterministe sur l'algorithme génétique	126
9.3.1	Principe	126
9.3.2	Implémentation	126
9.3.3	Application	127
9.4	Algorithme génétique pour la résolution 1 contre n	129
9.4.1	Objectifs	129
9.4.2	Définition de la population	129
9.4.3	Évaluation des individus	130
9.4.4	Caractère partiellement séparable du problème	131
9.4.5	Sharing	132
9.4.6	Étalonnage	133
9.4.7	Simulations	134
9.5	Conclusion	136

IV Applications	137
10 Mesure de différents facteurs	139
10.1 Introduction	139
10.2 Horizon de prédiction	140
10.2.1 Simulations	140
10.2.2 Méthode de mesure	140
10.2.3 Résultats	141
10.3 Sens préférentiels de circulation	144
10.3.1 Simulations	144
10.3.2 Influence sur les chemins	144
10.3.3 Influence sur les temps de roulage	145
10.4 Incertitudes sur la vitesse	147
10.4.1 Problèmes posés	147
10.4.2 Simulations	148
10.4.3 Résultats	148
10.5 Conclusion	150
11 Respect des créneaux de décollage	151
11.1 Introduction	151
11.2 Anticipation des temps de roulage	152
11.2.1 Problèmes posés	152
11.2.2 Estimation par la séquence de piste	152
11.2.3 Estimation par simulation	154
11.3 Prise en compte des créneaux de décollage	156
11.3.1 Approches envisagées	156
11.3.2 Modification du critère d'optimisation	157
11.3.3 Adaptation de la méthode 1 contre n	158
11.3.4 Affectation de bretelles d'entrée de piste	159
11.3.5 Simulations	159
11.4 Conclusion	162
Conclusions et perspectives	163
Annexes	165
A L'environnement aéroportuaire	167
A.1 Introduction	167
A.2 Infrastructures aéroportuaires	167
A.2.1 Postes de stationnement	167
A.2.2 Pistes	168
A.2.3 Voies de circulation	169
A.2.4 Autres aménagements	169
A.3 Exploitation de l'aéroport	169
A.3.1 Séparation des avions	169

A.3.2	Configurations de l'aéroport	170
A.3.3	Affectation des parkings	171
A.3.4	Affectation des pistes	171
B	Données disponibles	173
B.1	Introduction	173
B.2	Description topologique de l'aéroport	173
B.2.1	Éléments descriptifs	173
B.2.2	Analyse des données	176
B.3	Description du trafic	179
B.3.1	Types et performances des avions	180
B.3.2	Plans de vols	181
B.3.3	Enregistrements radar	184
C	Traitement des plans de vol	187
C.1	Introduction	187
C.2	Affectation de parkings	188
C.3	Affectation des pistes	189
C.3.1	Choix d'un QFU	189
C.3.2	Choix des bretelles de piste	190
C.4	Début des mouvements	190
D	Algorithmes de parcours de graphe	193
D.1	Notations	193
D.2	Algorithme de Floyd	194
D.3	Algorithme de Dijkstra	194
D.3.1	Description	194
D.3.2	Complexité	195
D.4	L'algorithme A*	195
D.4.1	Description	195
D.4.2	Complexité	196
D.5	Algorithme de type <i>Branch & Bound</i>	196
	Glossaire	199
	Bibliographie	203

Table des figures

1.1	Positions de contrôle du trafic	6
1.2	Capacité(s) de l'aéroport	8
3.1	Vitesse et taux de virage	22
3.2	Mesures directes	25
3.3	Mesures sur les trajectoires échantillonnées	25
3.4	Virages non pris en compte	28
3.5	Roissy face à l'Ouest	30
3.6	Roissy face à l'Est	30
3.7	Orly face à l'Ouest	31
3.8	Orly face à l'Est	31
3.9	Exemple de boucle ne passant jamais deux fois par le même nœud	32
4.1	Paramètres de la prédiction	36
4.2	Trajectoire discrétisée sans attente	38
4.3	Positions d'incertitudes	40
4.4	Réduction des incertitudes sur les points d'attente	41
4.5	Conflits avec incertitudes	44
4.6	Conflits entre avions qui se suivent	44
4.7	Simplification des conflits avec incertitudes	45
4.8	Simulation tabulée	46
5.1	Face-à-face à la fin de l'horizon	51
5.2	Surface de pareto en recherche multi-objectifs	53
5.3	Clusters d'avions en conflit	55
6.1	Applications de Dev_Max	63
6.2	Applications de REA et MPS	70
6.3	Exemple de détour difficilement utile	71
6.4	Applications de Deviations par MPS, avec $\delta = 300\%$	74
7.1	Illustration du problème 1 contre n sur un chemin	77
7.2	Pénalité en fonction du nombre d'avions sans incertitude	82
7.3	Temps de roulage moyens sans incertitude	82
7.4	Pénalité en fonction du nombre d'avions avec incertitudes	86
7.5	Retard moyen avec incertitudes	86
7.6	Temps de résolution moyens du problème 1 contre n	90

7.7	Exploration lexicographique des classements	91
7.8	Temps de résolution par tailles de clusters	94
7.9	Pénalité en fonction du nombre d'avions avec variations du classement	94
7.10	Retard moyen avec variations du classement	95
8.1	Étapes de l'algorithme génétique	99
8.2	Opérateur de mutation	103
8.3	Opérateur de croisement	104
8.4	Fonction de scaling exponentiel	105
8.5	Objectif du sharing	106
8.6	Allure de la fonction de sharing	107
8.7	Croisement adapté	110
8.8	Matrice d'adaptation	114
8.9	Influence du taux de croisement avec le critère homogène	117
8.10	Influence du taux de croisement avec le critère pondéré	118
8.11	Influence de la taille de la population sur la convergence	119
8.12	Influence du taux de sharing sur la convergence	120
8.13	Pénalité moyenne en fonction du nombre d'avions pour l'AG	121
8.14	Retard moyen avec l'AG	121
9.1	Influence des taux de croisement et de mutation déterministe	127
9.2	Pénalité moyenne avec mutation déterministe	128
9.3	Vitesse de convergence avec l'opérateur déterministe	129
9.4	Influence du taux de croisement de l'AG 1 contre n	133
9.5	Influence de la taille de la population sur la convergence	134
9.6	Pénalité moyenne en fonction du nombre d'avions pour l'AG 1 contre n	135
9.7	Retard moyen avec l'AG 1 contre n	135
10.1	Influence de l'horizon sur la moyenne des retards	141
10.2	Influence de l'horizon pour la méthode 1 contre n	142
10.3	Influence de l'horizon pour l'AG	142
10.4	Influence de l'horizon pour l'AG 1 contre n	143
10.5	Influence de l'horizon sur la taille des clusters	143
10.6	Influence des sens préférentiels sur les chemins	145
10.7	Influence des sens préférentiels sur les temps de roulage	146
10.8	Exemple de face-à-face à trois	146
10.9	Influence des sens préférentiels pour l'AG 1 contre n	147
10.10	Influence des incertitudes sur le retard	148
10.11	Influence des incertitudes pour l'AG 1 contre n	149
10.12	Influence des incertitudes sur la taille des clusters	149
11.1	Respect actuel des créneaux à Roissy	151
11.2	Anticipation des temps de roulage par la séquence de piste	154
11.3	Évaluation des temps de roulage par simulation	155
11.4	Aires d'attente	156
11.5	Critère pour un départ régulé	158
11.6	Stratégie d'affectation de bretelles par catégories	159
11.7	Stratégie d'affectation de bretelles par créneaux	159

11.8 Écarts aux créneaux sans réservation de bretelle	160
11.9 Écarts aux créneaux avec réservation de bretelle	160
11.10 Délais des vols non régulés sans réservation de bretelle	161
11.11 Délais des vols non régulés avec réservation de bretelle	162
A.1 Infrastructures aéroportuaires	168
B.1 Matrices de transitions	174
B.2 Bretelles d'entrée et de sortie de piste	175
B.3 Accès aux parkings	176
B.4 Description des transitions	177
B.5 Sorties rapides de piste	178
B.6 Transitions de rattrapage	179
B.7 Trace radar interrompue à Roissy	186
B.8 Traces radar fusionnées à Orly	186

Liste des algorithmes

II.1	Détection des conflits au roulage	46
III.1	Dev_Max : recherche des déviations bornées	61
III.2	REA : énumération récursive des k meilleurs chemins entre s et t	66
III.3	MPS : énumération des k meilleurs chemins sans boucle.	68
III.4	Deviations : recherche de chemins suffisamment différents.	72
III.5	A* pour la résolution du problème 1 contre n	80
III.6	<i>Branch & Bound</i> pour la résolution du problème 1 contre n	89
III.7	<i>Branch & Bound</i> pour la recherche du meilleur classement des avions	92
IV.1	Floyd : Coûts minimaux entre tous les couples de nœuds	194
IV.2	Dijkstra : Arbre des plus courts chemins depuis un nœud	195
IV.3	A* : Recherche d'un chemin entre un nœud source et des nœuds terminaux	196
IV.4	<i>Branch & Bound</i> avec recherche <i>meilleur en premier</i>	198

Introduction

Après une période de fléchissement conjoncturel, le trafic aérien a repris sa croissance et le dispositif de contrôle européen s'avère de nouveau proche de la saturation : l'augmentation des retards, qui pénalise tous les usagers, en est la principale caractéristique. Face à ce phénomène, les aéroports se trouvent souvent en première ligne :

- La grande diversité des intervenants (organismes de contrôle du trafic, compagnies aériennes, gestionnaires d'aéroports, sociétés d'assistance technique, mais aussi chambres de commerce, services de police, douanes, etc.) rend les opérations aéroportuaires extrêmement complexes et très sensibles à de nombreuses formes de dysfonctionnements.
- Le développement relativement récent de nombreux *hubs* commerciaux donne à l'aéroport un rôle essentiel, dans lequel les moindres retards ont des conséquences considérables, alors que cette forme d'exploitation de l'aéroport induit justement des pics de trafic difficiles à écouler.
- La multiplication des compagnies *low cost*, dont la survie est conditionnée par la minimisation des temps et des coûts d'escale, nécessite une organisation opérationnelle particulière et une réadaptation fréquente des lignes commerciales, sur des aéroports dont la taille n'est pas toujours adéquate.
- Enfin, la dégradation des conditions météorologiques, parfois non prévue, peut très rapidement paralyser l'aéroport, qui reste étrangement le seul endroit où les avions ne peuvent pas circuler sans visibilité.

Ces différents points mettent en évidence les limites d'un système dont le fonctionnement reste encore entièrement basé sur l'expertise et l'expérience humaine : un tel système s'avère beaucoup trop sensible aux imprévus et pose de nombreux problèmes de coordination entre ses intervenants. Il nécessite la définition et l'application de procédures opérationnelles rigides, qui aboutissent souvent à une exploitation sous-optimale des ressources disponibles : même si l'automatisation complète du système n'est pas matériellement envisageable et ne reste pour l'instant qu'un sujet de polémique, l'assistance de systèmes automatiques apparaît non seulement prometteuse pour l'optimisation des procédures au sol, mais semble de plus indispensable aux développements futurs des plus grands aéroports :

- De nombreuses innovations technologiques utilisant le radar de surface peuvent améliorer le guidage au sol des avions, en rendant notamment le fonctionnement de l'aéroport moins sensible aux baisses de visibilité et en augmentant globalement la sécurité.
- Des outils d'optimisation peuvent participer en temps réel à une meilleure répartition du trafic sur la plateforme, en cherchant à minimiser les temps de roulage et à maximiser l'utilisation des pistes. Lorsque les voies de circulation de l'aéroport sont nombreuses, ce travail d'optimisation peut difficilement se réaliser par l'application de procédures prédéfinies : le nombre de chemins et de points d'attente envisageables pour chaque avion est en effet combinatoire et l'ordre dans lequel les départs se présentent à la piste a une grande influence sur les retards engendrés.

Ce dernier point correspond au sujet de ce rapport de thèse : après une synthèse et une première analyse des problèmes aéroportuaires (première partie), une modélisation détaillée du trafic au sol permet de formaliser le problème d'optimisation à résoudre dans chaque situation de trafic (deuxième partie). La complexité de ce problème est élevée et différentes méthodes de résolution sont envisagées (déterministe, stochastique, puis hybride) et appliquées par simulation au trafic des deux principaux aéroports français : Orly et Roissy Charles De Gaulle (troisième partie). Diverses applications de ces méthodes d'optimisation sont ensuite présentées : elles permettent de mesurer l'influence de différents facteurs et peuvent facilement être adaptées à de nouvelles contraintes, comme les créneaux de décollage imposés aux avions traversant des espaces aériens saturés (dernière partie).

Cette thèse a donné lieu à deux publications aux conférences *Air Traffic Management R&D Seminar* organisées par EUROCONTROL [Gotteland 01, Gotteland 03c], deux publications aux colloques des doctorants de l'EDIT (École Doctorale d'Informatique et Télécommunications) [Gotteland 02, Gotteland 03a], une publication au séminaire CEC2003 (*Congress on Evolutionary Computation*) [Gotteland 03b] et à une participation à un chapitre applicatif de [Drého 03].

Première partie

Les problèmes aéroportuaires

Chapitre 1

Gestion du trafic aéroportuaire

1.1 Introduction

L'aéroport est indéniablement au centre du transport aérien : il est l'unique point de rencontre d'une multitude d'intervenants et constitue la seule position d'arrêt et de ravitaillement des avions. Les arrivées s'y transforment en nouveaux départs, ce qui lui confère un rôle particulier de dispositif d'alimentation des flux de trafic.

La complexité de l'environnement aéroportuaire le rend en contrepartie très peu accessible : ce premier chapitre n'a certainement pas la vocation d'aborder exhaustivement tous les aspects techniques du fonctionnement aéroportuaire, mais d'en présenter le cadre général pour en dégager les principales caractéristiques. Des informations plus précises sur les spécificités aéroportuaires sont néanmoins fournies en annexe A.

1.2 Contrôle du trafic aérien

1.2.1 Organisation générale

Les organismes du contrôle aérien sont responsables de l'écoulement du trafic dans l'espace dont ils ont la charge. Le service rendu aux usagers doit offrir des garanties de sécurité mais aussi le meilleur débit possible. L'espace aérien est pour cela partitionné en différentes positions de contrôle, dont la fonction est liée à la nature du trafic à gérer.

Trois principales catégories fonctionnelles de positions de contrôle se distinguent (figure 1.1) :

Le contrôle en route est responsable des vols en croisière : les avions contrôlés suivent des routes prédéfinies, à une altitude (*niveau de vol*) généralement élevée et dans des couloirs aériens réservés (*airways*). La surveillance du trafic dans ces espaces se fait le plus souvent par l'intermédiaire d'un écran radar et d'un contact radio avec les pilotes. Des manœuvres (virages, changements d'altitude, modifications de route) peuvent être ordonnées à tout moment par les contrôleurs pour assurer les séparations réglementaires entre aéronefs, tout en les guidant progressivement vers leur destination.

Le contrôle d'approche gère les avions aux alentours d'un aéroport, pendant leur phase de descente ou de montée. Le système de surveillance est identique à celui des secteurs en route, mais la densité de trafic est généralement plus élevée : des limitations de vitesse sont souvent imposées et peuvent être associées à des réductions de normes de séparation. Des procédures

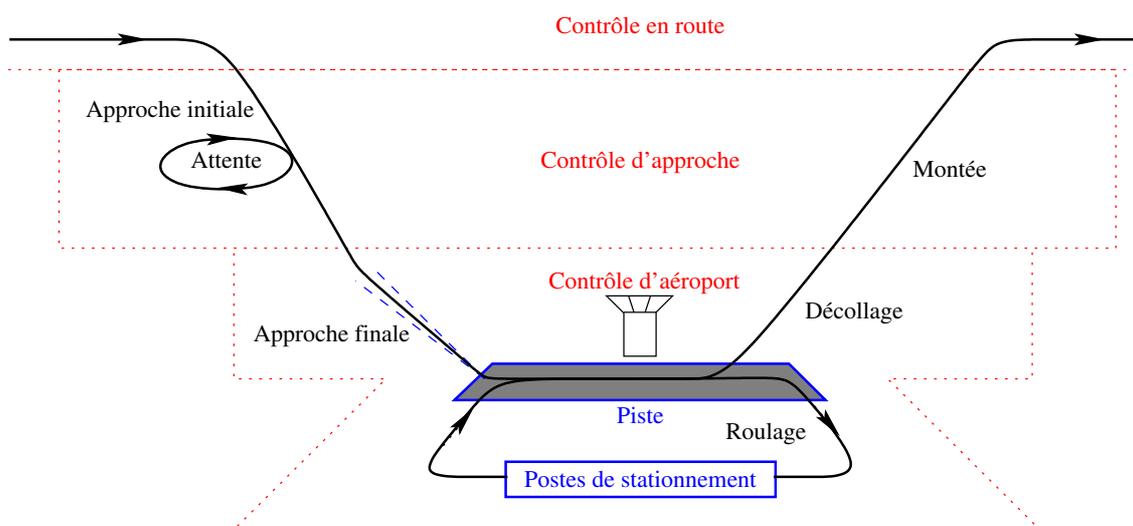


FIG. 1.1 – Positions de contrôle du trafic

d'attente spécifiques (*stack*) sont utilisées pour réguler le flux d'arrivée et préparer correctement le séquençage des atterrissages, bien avant la phase finale de la descente.

Le contrôle d'aéroport est responsable de toute la phase aéroportuaire des vols, depuis la descente finale de l'avion jusqu'à son décollage suivant, comprenant notamment le séquençage des mouvements sur la piste, le guidage au sol le long des voies de circulation (*taxiways*) et l'assistance pendant l'escale. La surveillance du trafic est avant tout visuelle (ces positions sont situées dans la tour de contrôle) mais les plus grands aéroports sont aujourd'hui munis d'un système de visualisation radar du trafic au sol.

1.2.2 Capacité du système

La notion de capacité permet de quantifier les limites de tout système offrant à des usagers un service partagé : elle se définit par la demande (ou charge) ne devant pas être dépassée pour que le service effectif rendu par le système corresponde à l'attente des usagers.

Pour le système de contrôle aérien, la capacité s'exprime en débit maximal d'avions pouvant être traité (nombre maximal de nouveaux vols par unité de temps). Elle peut s'exprimer de façon globale (à l'échelle d'un pays par exemple), mais chaque position de contrôle possède également sa propre capacité, fonction de la structure de l'espace, des moyens dont elle dispose et de la nature du trafic s'y écoulant.

La croissance du trafic aérien nécessite une constante amélioration de la capacité : le découpage de l'espace en différentes positions de contrôle est le moyen le plus utilisé pour maximiser la capacité globale du système de contrôle. L'efficacité du découpage admet pourtant une limite, car la coordination indispensable entre positions voisines induit inévitablement une charge additionnelle pour le système : en dessous d'une certaine taille élémentaire de chaque sous-espace (ou *secteur*), tout découpage supplémentaire aboutit, au contraire, à une diminution de la capacité globale.

1.2.3 Régulation du trafic

L'adaptation relative du trafic et de la capacité du système est le fruit d'un ensemble de filtres prédictifs, fonctionnant à échéances variables :

L'organisation à long terme est le filtre le plus macroscopique : son but est d'organiser le trafic plus de 6 mois en avance. La distribution des créneaux aéroportuaires pour la saison semestrielle suivante en est un exemple : des règles spécifiques fixées par l'IATA (International Air Transport Association) déterminent l'attribution de ces créneaux aux compagnies, en fonction de la capacité estimée des aéroports.

L'organisation à court terme (*pré-tactique*) consiste à réguler le trafic à partir des informations plus précises disponibles entre une semaine et un jour avant : cette régulation se base sur les plans de vol déposés, décrivant les intentions des compagnies, la capacité disponible, calculée en fonction du nombre prévu de positions de contrôle ou encore les données des années ou des semaines précédentes, lorsque certains phénomènes répétitifs sont attendus (pointes de trafic autour des jours fériés par exemple). En Europe, cette régulation est effectuée par un organisme centralisé : le CFMU (*Central Flow Management Unit*). Elle aboutit à des *créneaux de décollage* imposés à certains vols : ces créneaux correspondent à une heure précise de décollage, généralement retardée par rapport à l'heure initialement demandée, que les avions concernés doivent respecter (le contrôle d'aéroport en est averti et participe à leur application).

L'organisation en temps réel (*tactique*) permet également d'ajuster les flux de trafic, en fonction d'événements imprévus ou mal connus la veille (comme les phénomènes météorologiques par exemple). La régulation peut suivre le même processus que dans l'organisation à court terme (des créneaux sont imposés aux vols qui n'ont pas encore décollé), mais aussi se matérialiser par des changements de route (ou de destination) des vols en cours, ou même des annulations de vols.

1.3 Le trafic aéroportuaire

1.3.1 Contrôle d'aéroport

La grande diversité des opérations au sol nécessite souvent la coexistence de différentes positions de contrôle au sein d'un même aéroport :

- **Les positions « LOC »** gèrent les pistes de l'aéroport : elles sont responsables du séquençement des mouvements sur la piste, en accord avec les normes de séparation réglementaires, définies en temps et en distance en fonction des catégories de turbulence de sillage et des vitesses des avions. Le respect des normes induit notamment que la piste ne peut absorber qu'un seul mouvement à la fois. La principale difficulté liée à cette position est relative à la gestion du flux des arrivées, dont l'ordonnancement et la trajectoire ne peuvent plus être modifiés pendant l'approche finale. Les ordres de contrôle donnés aux avions concernent :
 - Les autorisations (*clairances*) de décollage et d'atterrissage ;
 - Les éventuelles modifications de vitesse pour les avions en approche finale : ces actions sont limitées par les performances des avions mais sont indispensables à la gestion correcte des pistes partagées par les départs et les arrivées ;
 - L'affectation des bretelles d'entrée et de sortie de piste, souvent négociée avec les pilotes ;
 - En dernier recours, la *remise de gaz* lorsque la piste n'est pas disponible pour une arrivée.

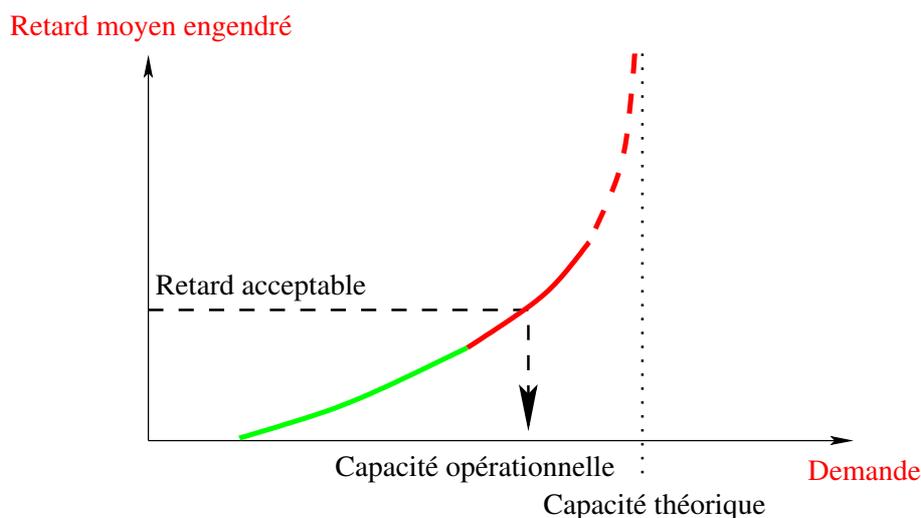


FIG. 1.2 – Capacité(s) de l'aéroport

- **Les positions « SOL »** sont chargées de la surveillance du trafic au roulage sur les *taxiways* : elles peuvent imposer un chemin à suivre à chaque avion et participent au respect des séparations aux abords des intersections, en ordonnant l'arrêt de certains d'entre eux. Aucune vitesse de roulage n'est cependant imposable : la consigne officielle préconise le roulage « au pas », à une vitesse adaptée aux conditions de circulation et le respect des séparations reste officiellement sous l'entière responsabilité des pilotes, ce qui rend parfois les prises de décision ambiguës.
- **Les positions « APRON »** sont responsables du séquençage des avions dans les zones de stationnement (ordre des entrées et des sorties de parkings).

1.3.2 Capacité d'un aéroport

Comme toute portion de l'espace aérien, un aéroport admet une capacité : l'ensemble du dispositif mis à la disposition des compagnies aériennes (installations et système de contrôle) aboutit à un débit maximal d'avions pouvant être traité. Différentes définitions de la capacité d'un aéroport sont cependant distinguées :

La capacité théorique de l'aéroport est le nombre de mouvements que le dispositif aéroportuaire pourrait idéalement écouler durant un temps spécifié sans tenir compte de la qualité de service (figure 1.2). Cette capacité correspond donc à un seuil ultime ou de saturation qu'il est en pratique impossible d'atteindre : elle ne peut se calculer que par extrapolation de mesures sur le trafic réel ou par l'intermédiaire de modèles de calcul représentatifs du fonctionnement de l'aéroport.

La capacité opérationnelle (ou pratique) correspond au débit maximal que l'aéroport peut réellement écouler, sans que la qualité de service en soit dégradée au delà d'un seuil fixé (figure 1.2). Ce seuil correspond en pratique à un niveau de sécurité requis (respect des règles de la circulation aérienne) et à un retard moyen ne devant pas être dépassé (3 à 4 minutes par avion par exemple). En général, cette capacité peut se déduire directement de l'analyse du trafic réel et de ses retards.

La capacité de programmation est la capacité affichée officiellement : elle est utilisée pour limiter le trafic aéroportuaire en amont, lors de l'attribution des créneaux aux compagnies. La différence entre la capacité de programmation et la capacité opérationnelle correspond à une marge de fonctionnement. Elle est relative aux incertitudes existant sur la demande : cette marge est d'autant plus importante que la demande finale de trafic dépasse la demande formulée lors la programmation.

1.3.3 Facteurs influençant la capacité aéroportuaire

La capacité de l'aéroport (qu'elle soit théorique, opérationnelle ou de programmation) est le résultat global du fonctionnement de chaque sous-système impliqué dans la gestion du trafic :

Les infrastructures de l'aéroport (annexe A) sont un premier facteur influençant la capacité. Le *dispositif de piste* (taille, nombre, disposition et équipement des pistes) est généralement considéré comme le point le plus sensible, mais l'impact des autres installations ne doit pas être négligé : la diversité et la disposition des parkings ou des voies de circulation (taille, accessibilité, équipements) influencent fortement les temps de roulage et peuvent être limitatifs pour certains types d'avion.

La nature et la répartition du trafic sont également des facteurs de première importance :

- La classification de l'aéroport détermine la nature des vols autorisés, en fonction de leur vitesse d'approche et de leurs objectifs (commercial, court ou long courrier), ce qui fixe indirectement l'ordre de grandeur de la capacité de l'aéroport.
- La configuration de l'aéroport (annexe A) détermine le taux de fréquentation de chaque catégorie d'avion et le ratio entre les flux d'arrivée et de départ sur chaque piste. Les normes de séparation entre chaque mouvement dépendent de leurs types et la capacité de la piste est donc différente dans chaque configuration.
- Enfin, les procédures d'exploitation des parkings et des voies de circulation (qui doivent être conformes aux règles de circulation), peuvent s'avérer plus ou moins efficaces en terme de retards générés. Sur ce point, la compétence technique des contrôleurs et des équipages (connaissance de la plateforme et de l'avion, séquençage des actions et aptitudes à collaborer) peuvent jouer un rôle considérable.

D'autres facteurs extérieurs au système aéroportuaire sont fortement limitatifs :

- Les problèmes environnementaux (écologie et gestion des nuisances sonores) ;
- Les politiques douanières et policières (anti-terrorisme notamment).
- Les phénomènes météorologiques dangereux (brouillard, grêle, orages, neige) ;

1.4 Difficultés aéroportuaires identifiées

1.4.1 L'environnement

Les problèmes environnementaux ont pris une ampleur conséquente pour tous les aéroports, quelle que soit leur taille : le développement des agglomérations et la prise en compte des nuisances sonores ou de la pollution induisent souvent des décisions limitatives pour la croissance du trafic aéroportuaire. Différents domaines d'étude sont considérés sur ce vaste sujet :

- La qualité de la concertation avec les riverains et l'analyse objective des nuisances ;
- La mise au point de procédures adaptées aux agglomérations limitrophes [Supino 03] ;
- La définition de limitations (horaires ou qualitatives) pour la circulation aérienne ;

- La promotion des innovations technologiques (moteurs moins polluants et moins bruyants).
- En France, l'aéroport d'Orly, rattrapé par l'agglomération parisienne, fait figure d'exemple sur le sujet : de nombreuses restrictions y limitent désormais le trafic.

1.4.2 La sécurité

L'amélioration de la sécurité est un souci permanent pour le service de contrôle et les usagers de l'aéroport. Sur ce point, l'aptitude à détecter les problèmes pouvant mettre en jeu la sécurité est primordiale : la concertation avec les pilotes et les contrôleurs d'aéroport, ainsi que l'analyse des incidents et des situations dangereuses observées s'avèrent indispensables. Les principaux sujets d'étude sont :

- L'amélioration de la couverture visuelle et radar à disposition des contrôleurs, pour augmenter la qualité de la surveillance du trafic ;
- La réduction des risques d'*incursion piste* (pénétration accidentelle de tout mobile non autorisé sur la piste).
- L'aide au guidage des avions sur la plateforme, pour diminuer les risques d'erreur de cheminement et donc de collisions, surtout par conditions de visibilité dégradées.

1.4.3 L'optimisation du trafic au sol

La gestion du trafic aéroportuaire a longtemps été mise au second plan, loin derrière les efforts menés pour le contrôle en vol. De récentes formes de dysfonctionnements pouvant aboutir à la congestion de l'aéroport, lourdes de conséquences pour l'ensemble des acteurs du transport aérien, inversent cette tendance :

- Les opérations au sol sont caractérisées par une grande diversité des imprévus : le nombre d'intervenants non coordonnés, la nécessité d'entretien et de contrôle régulier des infrastructures et des avions, la gestion des passagers et de leurs bagages sont autant d'éléments qui rendent difficile la planification et demandent une réactivité maximale en temps réel.
- Lorsque les conditions de visibilité sont dégradées, la circulation au sol peut rapidement être paralysée. Même lorsque le contrôle au sol dispose d'une visualisation radar de la situation, la réglementation actuelle ne prévoit pas le guidage précis d'un avion par le contrôle, car la responsabilité du pilote (pour l'anti-collision) n'est pas remise en cause. Ces situations (dangereuses) nécessitent donc souvent des pratiques opérationnelles à la limite de la réglementation, où la responsabilité de chacun n'est plus clairement spécifiée.
- Les nombreux *hubs* commerciaux impliquent que la plupart des mouvements sont programmés aux mêmes heures « stratégiques », pendant lesquelles l'aéroport sert essentiellement de relais : le moindre retard provoque inévitablement des effets collatéraux rarement maîtrisés.
- Les impératifs de la régulation européenne du trafic sont de plus en plus contraignants pour l'aéroport : les créneaux imposés sont des heures de décollage et non de départ du parking (justement parce que les temps de roulage sont trop imprécis). Leur respect est donc à la seule charge du contrôle d'aéroport, pour lequel ces contraintes peuvent parfois devenir ingérables.

Sur ces sujets, les réflexions menées ne doivent pas remettre en cause la sécurité : ils doivent être traités à niveau de sécurité constant (ou croissant).

Chapitre 2

Analyse préliminaire

2.1 Introduction

Dans le cadre général présenté au chapitre précédent, l'amélioration des conditions de circulation du trafic au sol se révèle un point crucial pour le développement futur des plus grands aéroports : les innovations attendues dans ce domaine peuvent en effet prétendre à la fois participer à réduire les problèmes environnementaux en limitant la congestion au sol, augmenter les conditions de sécurité en assistant le guidage des avions et réduire les retards par optimisation des trajectoires au sol.

La prise en compte relativement récente de ces intérêts a fortement accéléré l'évolution de ce secteur d'activité : la première partie de ce chapitre en présente les principales orientations. Même si très peu d'efforts sont encore portés sur la résolution précise des problèmes posés par chaque situation de trafic au sol, de nombreux projets visent à moderniser le dispositif de contrôle au sol et de nouveaux outils opérationnels sont déjà disponibles : la seconde partie en donne un aperçu. En fonction de cet état de l'art, la dernière partie aborde les objectifs et le cadre de travail dans lequel se situe toute la suite des travaux.

2.2 Les principaux concepts pour l'optimisation du trafic au sol

2.2.1 Optimisation de la séquence de piste

Sur les plus grandes plateformes aéroportuaires et dans les conditions « normales » d'exploitation, les pistes restent les éléments les moins capacitifs car les séparations imposées entre mouvements sont restrictives derrière les avions à forte turbulence de sillage et dépendent des conditions météorologiques. Sur ce point, différentes améliorations peuvent être envisagées.

Optimisation des procédures d'approche

La définition de procédures d'approche adaptées facilite le travail de séquençage initial des arrivées pour le contrôle d'approche, en rendant possible des changements de vitesses et de trajectoires des avions [Suarez 03]. La mise à disposition de différents hippodromes d'attente permet notamment de réguler le flux d'arrivée lorsqu'il dépasse la capacité de la piste. Dans certains cas, les procédures d'approche peuvent également être utilisées pour modifier légèrement l'ordonnement des arrivées, en fonction des contraintes de turbulence de sillage.

Plus généralement, l'amélioration des procédures d'approche peut être vue comme un problème d'optimisation complexe, contraint par les performances des avions (qui déterminent le type de tra-

jectoire qu'ils sont susceptibles de suivre) et dont le critère est la capacité de ces secteurs. Ce domaine d'étude nécessite une modélisation difficile des procédures aériennes utilisées pendant la phase d'approche. Il a toutefois l'avantage de pouvoir être traité hors de tout contexte temps réel et semble particulièrement porteur en Europe, où de nombreux aéroports voisins se partagent les mêmes secteurs d'approche.

Développement d'outils d'aide au contrôle

Le développement d'outils prédictifs (DST : *Decision Support Tool*) capables d'anticiper suffisamment le trafic aéroportuaire (atterrissages et départs) peut participer à la définition de meilleures séquences de piste. Là encore, la recherche d'un séquençement optimal des vols peut se formuler assez simplement comme un problème d'optimisation, contraint par les possibilités d'ordonnement des avions (en fonction des prévisions de trafic) et dont l'objectif est la minimisation des retards. Cependant, le problème doit ici être traité en temps réel, ce qui implique des contraintes supplémentaires, plus difficiles à exprimer.

Diminution des normes de séparation

Certaines conditions météorologiques particulières entraînent des baisses de capacités aéroportuaires considérables, notamment parce que le guidage de l'avion pendant toute l'approche finale (jusqu'à l'atterrissage) nécessite dans ce cas une augmentation importante de certaines normes de séparation. Deux aspects du problème peuvent être étudiés :

- La précision des équipements (embarqués ou au sol) permettant aux avions d'effectuer un atterrissage *de précision* (aux instruments et par mauvaise visibilité), qui relève de l'innovation technologique, peut évoluer favorablement et permettre, à terme, des réductions des normes de séparation en approche.
- L'étude aérodynamique de la turbulence de sillage des avions semble également porteuse. L'effet du vent fort sur la propagation de la turbulence peut par exemple aboutir à des normes de séparation en temps beaucoup moins contraignantes que les séparations en distance normalement pratiquées [Nicolaon 03, Cooper 03]. L'utilisation en temps réel d'instruments de mesure de l'importance du tourbillon (ou *vortex*) provoqué par un avion [Donohue 03] peut également apporter des gains de capacité non négligeables.

2.2.2 Amélioration de la prévisibilité du trafic aéroportuaire

L'aéroport apparaît comme l'endroit où le trafic aérien est le moins prévisible : l'enchaînement des actions pendant l'escale d'un avion relève d'une course contre la montre et dépend la plupart du temps de la gestion d'une multitude d'imprévus, comme cela a déjà été abordé dans la partie 1.4.3. Les temps de roulage sont de plus extrêmement sensibles au débit d'avions demandant la piste : dans ce contexte, l'amélioration de la prévisibilité du trafic nécessite une large diffusion des informations en temps réel, non seulement entre les positions de contrôle d'aéroport et d'approche, mais également entre chaque intervenant sur un même aéroport.

Coordination des aéroports

L'échange d'informations de trafic entre différents aéroports est depuis longtemps un sujet d'étude, bien que le nombre d'aéroports « coordonnés » soit encore relativement faible. La connaissance des retards prévus sur un aéroport permet pourtant d'optimiser la gestion des vols de tous les aéroports

qui lui sont connectés. Dans ce cadre, des systèmes (AMAN : *Arrival Manager* et DMAN : *Departure Manager*), mis à jour en temps réel, permettent d'informer les aéroports coordonnés des évolutions du trafic, mais utiliser l'ensemble de ces informations à bon escient reste encore un problème à part entière [Gilbo 03], surtout en ce qui concerne le DMAN [Idris 98].

Partage d'informations entre les intervenants (CDM)

Le cadre général du CDM (*Collaborative Decision Making*) lancé par le département AOP (*Airport Operations Unit*) d'EUROCONTROL vise à perfectionner la gestion en temps réel des opérations par le partage des informations entre les différents acteurs du trafic aérien (principalement le contrôle et les compagnies) : ces intervenants doivent prendre des décisions qui peuvent influencer l'ensemble du trafic alors qu'ils n'en ont qu'une vision partielle (les compagnies ne connaissent que leurs propres vols). Ce mode de fonctionnement (qui ne peut pas être optimal) doit être amélioré par le partage d'informations [Martin 01].

Ce concept apparaît à première vue délicat, dans un environnement fortement concurrentiel : l'idée est pourtant que chaque compagnie a plus à gagner avec les informations des autres qu'à perdre avec la divulgation des siennes. La régulation du trafic peut par exemple en être fortement améliorée : lorsqu'une compagnie sait qu'elle ne pourra pas utiliser certains de ses créneaux de décollage (à cause de retards en cours ou de problèmes techniques), elle pourrait le signaler à d'autres compagnies et les échanger. Sans cette coopération dynamique, les créneaux initiaux sont de toutes façons manqués et la régulation du trafic, au lieu d'optimiser sa répartition, pénalise finalement tous ses intervenants.

2.2.3 Systèmes évolués de surveillance et de guidage au sol (ASMGCS)

Les premiers concepts SMGCS (*Surface Movement Guidance and Control System*) introduits par l'OACI (Organisation de l'Aviation Civile Internationale) ont abouti au système actuel de suivi radar du trafic au sol. Depuis les années 90, ils font donc place aux nouveaux concepts ASMGCS (*Advanced SMGCS*), toujours sous l'égide de l'OACI en collaboration avec la Commission Européenne, EUROCONTROL, EUROCAE (*European Organisation for Civil Aviation Equipment*), la FAA (*Federal Aviation Administration*), la NASA (*National Aeronautics and Space Administration*) et les aéroports. Ces concepts fixent un ensemble d'objectifs pour les futurs systèmes de surveillance et de guidage des avions au sol [Vallée 01] :

La sécurité doit être augmentée, par une redéfinition des rôles et des responsabilités de chacun : particulièrement par mauvaise visibilité, l'emploi de la visualisation radar comme instrument de contrôle (et non comme simple source d'information auxiliaire) doit être officialisée. Pour cela, le développement de visualisations radar plus perfectionnées, offrant une vision complète et sûre de l'ensemble des mobiles sur la plateforme, (véhicules et aéronefs) doit pouvoir assurer le guidage des avions. Dans ce cadre, de nouvelles technologies sont proposées :

- Le D-GPS (*Differential Global Positioning System*), capable de fournir à tout mobile un positionnement dans l'espace, avec une précision inférieure au mètre, à partir de relèvements satellites et des corrections émises par un satellite géostationnaire ;
- L'ADS-B (*Automatic Dependant Surveillance*) permettant la diffusion de la position d'un mobile à tous les autres.

Des systèmes de détection de pertes de séparation et d'incursion piste doivent également être en mesure de donner l'alerte au contrôle (fonction de *filet de sauvegarde*).

La gestion du trafic au sol doit être facilitée et améliorée par des outils de supervision technique et opérationnelle :

- L'amélioration du système de guidage vise en particulier un fonctionnement de l'aéroport « tout-temps », au sein duquel les avions peuvent continuer à circuler dans des conditions de visibilité les plus réduites (brouillard dense).
- Des outils reliés au système de visualisation radar, capables d'effectuer le suivi des avions, peuvent participer à leur séquençage dans les zones de parking et sur les bretelles d'accès à la piste.
- Le *Data Link* (transmission automatique de certaines données entre le contrôle et l'avion) peut également jouer un rôle en simplifiant la tâche des contrôleurs et en rendant plus disponible la fréquence de communication radio.

Ce cadre très général montre tout l'intérêt porté sur la phase de roulage et l'importance des équipements mis en jeu : il semble évident, dans ce contexte, que cette phase du vol (longtemps délaissée) se destine à une véritable révolution technologique.

2.3 Les outils et les projets

2.3.1 Les outils opérationnels du contrôle d'aéroport

Les équipements des postes de contrôle d'aéroport varient naturellement d'un aéroport à l'autre et d'un poste à l'autre sur un même aéroport. Dans tous les cas, la communication avec les pilotes se fait par liaison radio et la coordination avec les positions de contrôle voisines (en route ou d'approche) par liaison téléphonique. Un système de *strips* (support papier de format prédéfini) permet la réception et le suivi des plans de vol (intentions des compagnies). Sur les grands aéroports, le poste de contrôle « SOL » comprend un système de visualisation ODS (*Operational Display System*) du trafic au sol représentant les données reçues d'un radar de surface. De nombreux systèmes opérationnels viennent s'ajouter à ce dispositif et l'objet de cette partie n'est certainement pas d'en faire une revue technique complète, mais de présenter ceux qui s'intègrent le mieux dans le cadre de l'optimisation du trafic. Roissy sert naturellement de référence sur ce sujet :

- **AVISO** (Aide à la Visualisation au SOL) [Vallée 01] effectue le traitement des données radar à Roissy et présente l'ensemble des véhicules et des avions en mouvement.
- **SALADIN** (Système d'Aide à LA Détection d'INcursions) [Vallée 01] est une extension du système AVISO, dans laquelle un système de détection d'incursions piste a notamment été ajouté : l'outil RIMCAS (Runway Incursion Monitoring and Collision Avoidance System), déjà installé sur de nombreux aéroports dans le cadre du ASMGCS, déclenche une alerte pour le contrôle.
- **MAESTRO** (Moyen d'Aide à l'Écoulement Séquencé du Trafic avec Recherche d'Optimisation) [Bourdon 98], installé à Roissy, à Orly et au centre de contrôle de la région parisienne (secteurs en route et d'approche), est un système d'aide à la régulation et à l'écoulement des vols à l'arrivée, qui diffuse les informations de trafic en temps réel (il entre dans le cadre des AMAN : Arrival Manager). Ses objectifs sont :
 - L'amélioration de la coordination entre contrôle en route, l'approche, et l'aéroport ;
 - La réduction des attentes en l'air (peu économiques) ;
 - La gestion globale des délais ;
 - Une meilleure utilisation de la capacité de résorption du système de contrôle.

2.3.2 Quelques outils d'analyse

De nombreux outils informatiques, utilisés dans le cadre d'études sur les futures évolutions du contrôle ou visant à spécifier des outils d'aide au contrôle sont développés. Ils permettent essentiellement de mesurer de manière plus ou moins macroscopique l'effet de différents facteurs sur l'écoulement du trafic :

- **TAAM** (*Total Airspace and Airport Modeler*) [Preston 99], qui est un simulateur de trafic complet développé par *Preston Aviation Solutions* : il simule toutes les étapes des vols, du parking de départ jusqu'au parking d'arrivée (*gate-to-gate*) avec un niveau de détail élevé, mais au prix d'un paramétrage parfois difficile. La gestion des séparations entre avions est gérée par des règles de priorité, mais aucune recherche d'optimisation globale des solutions n'est finalement proposée.
- **SIMMOD** (*SIMulation MODel*) [FAA 98], mis en œuvre par la FAA (*Federal Aviation Administration*), est également un simulateur *gate-to-gate*, mais plus simplifié que TAAM. Les séparations au roulage ne sont pas considérées et l'accès des avions à la piste est modélisé par une file d'attente (dans l'ordre théorique prévu d'arrivée à la piste), ce qui néglige donc les problèmes de circulation au sol des avions. Le simulateur peut en revanche évaluer les variations des retards en fonction des configurations des pistes.
- **OPAL** (*Optimisation Platform for Airports, including Land-side*) [van Eenige 02] développé par le NLR (*Nationaal Lucht & Ruimtevaartlaboratorium*) dans le cadre d'un programme dirigé par la Commission Européenne, est une plateforme d'analyse de l'ensemble des opérations aéroportuaires (y compris le séquencement de piste), reliant entre eux divers outils de simulation, capables de mesurer différents indicateurs de performance de l'aéroport (capacité, sécurité, nuisances pour l'environnement).
- **SALSA** (Systèmes d'Assistance à la circulation Locale et au Sol sur les Aéroports), développé par le CENA (Centre d'Étude de la Navigation Aérienne), est une plateforme de simulation intégrant un ensemble d'outils expérimentaux pour la gestion du trafic au sol. Une description détaillée des voies de circulation de l'aéroport est utilisée pour y faire évoluer les avions et générer des situations de trafic précises. Dans l'état actuel, il ne propose pas de méthode de résolution des situations au sol, mais les travaux qui seront présentés dans la suite de ce rapport donneront lieu, à terme, à l'intégration d'un module de résolution des situations au sol pour ce simulateur.
- **SAMBA** (Systèmes d'Analyse des Mouvements et des Besoins des Aéroports) développé par le CENA (Centre d'Étude de la Navigation Aérienne) propose une modélisation de l'aéroport permettant l'évaluation du service rendu par le système et la mesure de différents indicateurs sur l'aéroport et ses secteurs d'approche (charge, capacité, efficacité). Le projet est basé sur l'analyse d'échantillons de trafic issus des enregistrements produits par le système opérationnel.

2.3.3 Les projets

Beaucoup de projets européens (regroupant les organismes de l'Aviation Civile de différents pays, coordonnés par EUROCONTROL) ont visé à l'amélioration du trafic aéroportuaire :

- **ATHOS** (1996-1998) : *Airport Tower Harmonised Controller System* [Pham-Dumesnil 98], est un projet étudiant les différents postes de contrôle d'aéroport : l'objectif principal est de regrouper l'ensemble des outils opérationnels pouvant déjà exister en un seul, de manière à diffuser des informations pertinentes (« the right data at the right moment »). Expérimenté sur différents aéroports (dont Orly), le projet fait apparaître la nécessité de tels systèmes, mais n'aboutit pas à

un outil opérationnellement viable, d'autant plus qu'il se base sur une gestion de l'aéroport qui nécessiterait des évolutions préalables à préciser.

- **DEFAMM** (1996-1998) : *Demonstration Facilities for Airport Movement Management*, décrit un système qui centralise les informations de trafic et qui propose un chemin précis pour chaque avion [Monzel 98]. Lorsque le chemin est validé par le contrôleur, sa transmission au pilote est automatique (elle s'effectue par exemple par *Data Link*, sans communication radio). Les expériences réelles menées sur plusieurs aéroports (dont Orly) montrent les possibilités du système, mais ne font apparaître aucune baisse de charge de travail pour les contrôleurs, qui gardent l'entière responsabilité de la surveillance (le système ne résout pas les conflits aux intersections) et qui restent plutôt hostiles à l'absence de contact radio avec les pilotes : la confirmation orale du chemin que va suivre l'avion est supprimée, ce qui demande finalement une charge de surveillance supérieure.
- **MANTEA** (1996-1999) : *Management of Surface Traffic in European Airports* [Paul 99] définit de nouveaux outils d'assistance au contrôle d'aéroport. Le projet propose notamment un système de suivi du trafic relié au radar de surface, capable d'assurer la surveillance des mouvements en comparant les trajectoires prévues avec les trajectoires réelles. Il détecte les *conflits* (risques de pertes de séparation) et propose des solutions : l'aéroport est partitionné en différentes zones traitées séparément (l'optimisation n'est donc pas globale). Il propose par ailleurs un système d'anticipation des retards (MACAD : *Mantea Airfield Capacity and Delay*), calculés à partir de la charge de trafic prévue et de la capacité des pistes (les problèmes de circulation entre les parkings et les pistes ne sont pas considérés pour cette évaluation) : expérimenté sur plusieurs aéroports (dont Orly), le système semble prometteur pour les contrôleurs, bien qu'ils n'y voient aucune utilité immédiate autre que par mauvais temps et sous condition que la réglementation précise la responsabilité du système.
- **LEONARDO** (2001-2003) : *Linking Existing ON-Ground, Arrival and Departure Operations* est un projet de la Commission Européenne terminé en 2003. Il étudie les gains potentiels d'une centralisation des données des compagnies et du contrôle aérien, dans le cadre du CDM (*Collaborative Decision Making*).

Cet ensemble de projets montre globalement que des solutions techniques existent pour une meilleure gestion du trafic aéroportuaire, mais nécessitent encore des améliorations pour être vraiment bénéfiques et ne peuvent se réaliser sans évolution préalable de la réglementation et des procédures. Le programme **APR** (*Airport operations PRogramme*) d'EUROCONTROL [Miart 04] fixe aujourd'hui quatre grandes orientations pour l'amélioration du trafic aéroportuaire :

- Le CDM (*Collaborative Decision Making*) [Lane 03] ;
- Le ASMGCS [Adamson 02] ;
- La gestion de la capacité des pistes (ACE : *Airside Capacity Enhancement*) [De Clercq 03] ;
- La sécurité sur les pistes (détection des incursions) [Page 03].

Dans ce programme, le développement des concepts ASMGCS est planifié en quatre étapes, dont les deux dernières ne sont pas encore à l'ordre du jour :

1. La spécification d'un système permettant une meilleure surveillance de la circulation au sol et la définition de nouvelles procédures plus adaptées à ce système, concernant l'identification des avions et les instructions ou les autorisations (clairances) de contrôle [Adamson 03a] ;
2. L'augmentation de la sécurité au niveau des aires de pistes, par la détection systématique de toute incursion et le déclenchement d'alerte pour le contrôle [Adamson 03b] ;
3. La détection des conflits (risques de collision) sur toute la plateforme et entre tous les mobiles (avions et véhicules), accompagnée de systèmes de guidage des avions plus perfectionnés ;

4. La résolution automatique des conflits et le guidage automatique des avions au sol.

Toujours dans le cadre du ASMGCS, la FAA (*Federal Aviation Administration*) développe les projets **SMA** (*Surface Movement Advisor*) [Lawson 97] et **SMS** (*Surface Management System*), dont le but est la spécification d'outils d'aide au contrôle d'aéroport [Atkins 03]. Le projet fait partie du programme *Free Flight* et en a hérité le découpage en deux étapes successives :

- Dans la première (*Free Flight Phase I*), deux simulations ont été effectuées sur l'aéroport international de Dallas Fort Worth, avec des contrôleurs opérationnels. Il en résulte la spécification de trois composants essentiels : un système évaluant la charge de trafic à venir, un outil d'estimation de la séquence de piste et une interface pour les contrôleurs.
- Dans la deuxième (*Free Flight Phase II*), les outils spécifiés pour le SMS sont développés et mis en situation, lors de simulations sur l'aéroport international de Memphis. Leur efficacité est mesurée par la précision des prédictions de trafic, la pertinence de la séquence de piste prévue et le jugement des opérationnels. Les résultats sont prometteurs et le SMS semble exploitable dans un futur proche, bien qu'une amélioration de la prédiction de trafic soit encore à l'ordre du jour.

D'un point de vue général, la définition d'une nouvelle forme de contrôle au sol est donc largement entamée et les solutions techniques pour l'évolution de la surveillance du trafic sont déjà disponibles, bien que les notions de guidage automatique des avions au sol (avec optimisation des trajectoires) soient encore un point peu étudié.

2.4 Objectifs

2.4.1 Cadre de travail

Les projets qui viennent d'être décrits définissent le cadre général dans lequel s'inscrivent les travaux qui seront présentés dans les chapitres suivants : les évolutions attendues du système de contrôle (concepts ASMGCS) rendent possible le suivi du trafic en temps réel par un système informatisé, ce qui laisse envisager l'intégration d'outils automatiques participant à l'optimisation de la circulation des avions. Les caractéristiques d'un tel outil sont les suivantes :

- **Le système est destiné au contrôle d'aéroport** : l'ensemble du trafic aérien peut être connu (notamment le trafic des secteurs d'approche), mais l'optimisation ne concerne que la phase aéroportuaire des vols, c'est-à-dire le séquençement de piste, le cheminement des avions le long des voies de circulation et les accès aux parkings.
- **Le système doit être prédictif** : il doit anticiper les problèmes risquant de survenir, pour assister le contrôleur. Pour cela, l'ensemble des contraintes liées à l'écoulement sécurisé du trafic doit être connu du système (infrastructures de l'aéroport, règles de circulation, caractéristiques des avions et intentions des compagnies).
- **Les situations doivent être considérées en temps réel** : l'état du système doit évoluer dynamiquement avec les avions. Ceci implique que le temps pour étudier chaque situation est limité et que les informations disponibles sur les situations à venir peuvent être incomplètes : ces concepts s'intègrent ainsi dans le cadre de la rationalité limitée [Simon 92], ce qui rend l'évaluation des solutions proposées plus délicate.
- **Un objectif d'optimisation globale** doit néanmoins caractériser la recherche de solutions : le but n'est pas seulement d'assurer la séparation des avions, mais aussi de proposer une répartition efficace du trafic sur la plateforme (maximisation du débit ou minimisation des retards).

2.4.2 Niveau de réalisme recherché

La définition d'un système prédictif peut se faire de manière plus ou moins macroscopique : l'analyse des données disponibles (annexe B) permet, sur ce point, de dégager les premières spécifications du modèle :

- La description détaillée des voies de circulation de l'aéroport permet d'envisager plusieurs possibilités de cheminement pour chaque avion, en intégrant éventuellement certaines pratiques opérationnelles, comme l'application de sens préférentiels et le respect des configurations réelles de l'aéroport.
- La base des plans de vol journaliers (annexe B.3.2) permet de générer des situations de trafic comparables (en débit tout au moins) aux situations réelles : cette étape de simulation est indispensable à la mise au point du système. Le réalisme de l'affectation des parkings et des pistes sera cependant limité :
 - Le manque d'information sur les limites d'utilisation des parkings et sur les mouvements poste-à-poste ne permet pas d'envisager sérieusement l'optimisation des affectations de parkings pendant toute une journée de trafic : ces affectations seront donc considérées comme des données d'entrée du problème, supposées respecter les contraintes opérationnelles.
 - Une partie (certes minoritaire) des plans de vol est incomplète et nécessite des traitements particuliers pour être prise en compte (annexe C).
 - Les procédures d'affectation de bretelle d'entrée ou de libération de la piste ne sont pas précisées et doivent être modélisées par des règles ajoutées (annexe C.3.2).

Dans ce cadre, les enregistrements des traces radar des avions (annexe B.3.3) se révéleront particulièrement utiles pour évaluer le réalisme des situations simulées.

2.4.3 Résolution des situations

La résolution de chaque situation de trafic consiste à trouver les manœuvres qui doivent être imposées aux avions (ordres de contrôle) pour optimiser globalement leurs trajectoires en respectant les règles de circulation. Différentes approches peuvent être envisagées pour formuler rigoureusement le problème d'optimisation à résoudre :

- **L'approche par capacité** semble par exemple attrayante : les éléments structurels de l'aéroport (pistes, portions de voies de circulation, parkings) sont représentés par un graphe orienté. Chaque élément (ou nœud) du graphe se voit attribuer une capacité et un modèle de file d'attente permet de calculer les retards induits par un flux d'avions dans ce graphe. Le problème consiste alors à trouver un flux optimal, qui minimise les retards induits. Ce problème combinatoire (par rapport à la taille du graphe et au nombre d'avions) peut être résolu par programmation linéaire [Stoica 02]. Cette approche trouve donc une répartition optimale du trafic, mais admet ses limites : la notion de capacité ne peut pas traduire précisément les problèmes de circulation lorsque les voies sont dépendantes sans être connectées (aux alentours des intersections notamment).
- **L'approche par simulation détaillée** permet de mieux représenter le trafic au sol : les trajectoires envisageables des avions sont calculées précisément, et le problème consiste à trouver un ensemble de trajectoires qui respecte les normes de séparation à chaque instant et qui minimise une fonction de coût donnée.

La deuxième approche formalise un problème plus complexe mais également mieux paramétrable et plus représentatif de la réalité : toute la suite de ce rapport lui est consacrée.

Deuxième partie

Modélisation

Chapitre 3

Modélisation de l'aéroport

3.1 Introduction

Dans le cadre de la simulation du trafic aéroportuaire, la modélisation de l'aéroport est essentiellement destinée à l'attribution de différentes possibilités de cheminement à chaque mouvement. Pour être acceptables, les chemins proposés ne doivent pas seulement être conformes avec les tracés au sol, mais aussi respecter quelques règles opérationnelles :

- Les chemins proposés doivent être sans boucle : il est difficilement avantageux de faire passer un avion deux fois au même endroit.
- Un avion ne doit pas circuler via un parking, même si ce dernier est inoccupé : certains parkings disposent d'une sortie en marche avant et pourraient être interprétés (à tort) comme des raccourcis.
- Un avion ne doit circuler dans les aires de piste que lorsque cela est indispensable : juste avant le décollage, juste après l'atterrissage ou s'il lui est nécessaire de traverser la piste.

Cette liste demanderait à être complétée pour chaque aéroport : l'attribution de chemins aux avions doit rester évolutive et adaptable facilement à de nouvelles contraintes.

Telle qu'elle est fournie, la description des voies de circulation de l'aéroport peut être modélisée sous forme d'un graphe orienté, reliant entre elles chaque intersection de taxiway et chaque voie d'accès aux parkings ou aux pistes : la recherche des cheminements possibles pour un avion correspond à l'énumération des meilleurs chemins, avec contraintes, dans ce graphe.

La notion de *meilleurs* chemins est alors relative aux coûts affectés aux arcs du graphe : pour prendre en compte le temps de parcours des chemins (et non uniquement leur longueur), le coût affecté à chaque arc doit représenter un temps. Il est donc nécessaire de modéliser la vitesse de circulation des avions sur chaque portion ou voie de circulation. Le calcul de ces vitesses, appelées vitesses nominales de déplacement, est détaillé dans la première partie de ce chapitre. Les parties suivantes sont consacrées à l'élaboration du graphe, de telle façon qu'il décrive aussi rigoureusement que possible les voies de circulation de l'aéroport et leurs restrictions d'utilisation.

Le développement et l'application d'algorithmes de recherche de chemins dans ce graphe est l'étape initiale de la résolution des situations de trafic et le chapitre 6 lui est entièrement consacré.

3.2 Vitesses nominales de déplacement

La vitesse de roulage des avions n'est pas fixée : les seules consignes opérationnelles explicitement formulées préconisent une vitesse adaptée aux risques relatifs à la situation de l'avion. La vitesse doit

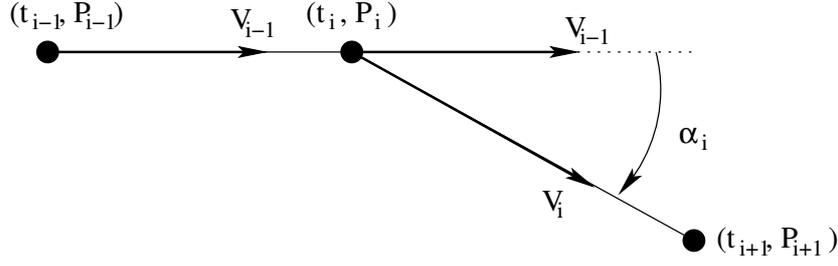


FIG. 3.1 – Vitesse et taux de virage

donc être modélisée différemment selon la nature du taxiway où il se trouve :

- Pour des portions de taxiways relatives à des procédures spécifiques, comme les bretelles de libération de la piste ou les voies d'entrée et de sortie de parking, les vitesses maximales peuvent être définies comme des paramètres de la simulation.
- Pour toutes les autres portions (les *transitions* de la description de l'aéroport, telles que définies en annexe B.2.1), la vitesse de roulage doit prendre en compte le virage que doit effectuer l'avion.

Les enregistrements des traces radar réelles des avions servent naturellement de référence à la modélisation de ces vitesses.

3.2.1 Méthodes de mesures

Deux méthodes peuvent être utilisées pour étudier la relation entre la vitesse de roulage des avions et le virage qu'ils effectuent. Elles seront toutes deux développées :

- La première consiste à estimer directement un taux de virage et une vitesse instantanée sur chaque plot radar, pour étudier leur corrélation. Les imprécisions des échos radar peuvent cependant biaiser ces mesures.
- La seconde méthode consiste à échantillonner la trajectoire de chaque avion selon un pas de temps fixe, en effectuant une extrapolation barycentrique des plots radar successifs : l'objectif est de lisser les trajectoires et de réduire les imprécisions inhérentes aux échos radar.

Mesures directes

La trajectoire au sol d'un avion est définie par une suite de n échos radar successifs $(t_i, P_i)_{1 \leq i \leq n}$:

- t_i est l'heure du relèvement radar du $i^{\text{ème}}$ plot, donnée à la seconde près. La suite $(t_i)_{1 \leq i \leq n}$ est donc supposée strictement croissante.
- P_i est la position qui correspond au $i^{\text{ème}}$ plot de l'avion, fournie au mètre près.

Sur chaque plot intermédiaire de la trajectoire ne correspondant pas à l'arrêt de l'avion (figure 3.1) la vitesse de roulage instantanée v_i et le taux de virage τ_i de l'avion peuvent être estimés par :

$$\forall i \in \{2; \dots; n-1\}, \text{ tel que } P_{i-1} \neq P_i \neq P_{i+1} \left\{ \begin{array}{l} v_i = \frac{\| \overrightarrow{P_{i-1}P_i} \|}{t_i - t_{i-1}} \\ \tau_i = \frac{|\alpha_i|}{\| \overrightarrow{P_{i-1}P_i} \|} \end{array} \right.$$

avec $\alpha_i = (\overrightarrow{P_{i-1}P_i}, \overrightarrow{P_iP_{i+1}})$

Cette définition de taux de virage correspond à un angle de virage rapporté à une distance parcourue. Ce taux est donc rendu indépendant de la vitesse de parcours adoptée par l'avion : il peut être considéré comme inversement proportionnel au rayon de courbure R du chemin en ce point :

$$R = \frac{2\pi}{\tau}$$

Mesures sur les trajectoires échantillonnées

La suite $(t_i, P_i)_{1 \leq i \leq n}$ désigne toujours les n plots radar successifs d'un avion. Un pas de temps élémentaire $\delta \in]0, t_n - t_1[$ permet d'extraire de l'intervalle $[t_1, t_n]$ $n_\delta - 1$ sous intervalles de même largeur $([t'_j, t'_{j+1}])_{1 \leq j \leq n_\delta - 1}$:

$$n_\delta = 1 + E \left(\frac{t_n - t_1}{\delta} \right) \quad (\text{où } E \text{ désigne la partie entière.})$$

$$\text{et } \forall j \in \{1, \dots, n_\delta\}, t'_j = t_1 + \delta(j - 1)$$

L'échantillonnage de la trajectoire radar toutes les δ secondes est la suite de plots $(t'_j, P'_j)_{1 \leq j \leq n_\delta}$ extrapolés de la façon suivante :

$$\forall j \in \{1, \dots, n_\delta\}, \quad t_1 \leq t'_j < t_n$$

et comme la suite $(t_i)_{1 \leq i \leq n}$ est strictement croissante :

$$\exists ! i \in \{1, \dots, n - 1\} \quad t_i \leq t'_j < t_{i+1}$$

$$\text{Ce qui permet de définir : } \overrightarrow{OP'_j} = \frac{1}{t_{i+1} - t_i} \left[(t_{i+1} - t'_j) \overrightarrow{OP_i} + (t'_j - t_i) \overrightarrow{OP_{i+1}} \right]$$

La vitesse de passage sur chaque point de la trajectoire échantillonnée est alors estimée par :

$$v_j = \frac{\| \overrightarrow{P'_{j-1}P'_j} \| + \| \overrightarrow{P'_jP'_{j+1}} \|}{2\delta} \quad (1 < j < n_\delta)$$

Et lorsque cette vitesse est non nulle, un taux de virage moyen est calculé :

$$\tau_j = \frac{2 | \overrightarrow{P'_{j-1}P'_j}, \overrightarrow{P'_jP'_{j+1}} |}{\| \overrightarrow{P'_{j-1}P'_j} \| + \| \overrightarrow{P'_jP'_{j+1}} \|} = \frac{\alpha'_j}{\delta v_j} \quad \text{avec } \alpha'_j = (\overrightarrow{P'_{j-1}P'_j}, \overrightarrow{P'_jP'_{j+1}}) \quad (1 < j < n_\delta)$$

Filtrage des mesures

Afin d'exclure les vitesses particulières liées aux atterrissages et aux décollages et autant que possible les détections erronées du radar, un plot ou un point extrapolé d'une trajectoire ne sera retenu que sous certaines conditions :

- Le point est en dehors des aires de piste (la distance à l'axe de piste le plus proche est supérieure à une borne D fixée) ;

- La vitesse de passage estimée v sur le point est bornée : $v \in [v_{inf}; v_{sup}]$. La borne inférieure permet d'exclure les points correspondant à l'arrêt de l'avion (points d'attente) et la borne supérieure permet d'éliminer les incohérences dues aux fusions de trajectoires radar ou aux pertes de détection d'un avion pendant un intervalle de temps trop grand.

Les trajectoires d'entrée et de sortie de parking n'étant pas enregistrées dans les données radar, les vitesses et les taux de virage ainsi sélectionnés correspondront au roulage normal sans attente des avions.

3.2.2 Résultats

Les deux méthodes de mesure des couples (τ, v) sont effectuées pour chaque vol d'une journée de trafic à Roissy et à Orly avec les paramètres suivants :

Largeur des aires de piste : $D = 90m$
Borne inférieure des vitesses : $v_{inf} = 1m/s$
Borne supérieure des vitesses : $v_{sup} = 35m/s$
Pas de discrétisation des trajectoires : $\delta = 5s$

et à partir des enregistrements radar suivants :

20/03/2000	Roissy	Orly
Nombre de trajectoires enregistrées	1403	442
Nombre d'échos radar sélectionnés	605799	102239
Nombre de points échantillons	113615	20809

Les courbes 3.2 et 3.3 montrent le maximum, la moyenne et l'écart type des vitesses de roulage pour chaque valeur du taux de virage (mesuré au dixième de degré par mètre près). Ces courbes fournissent également les vitesses correspondant aux quantiles à 95% et aux 3^{ème} quantiles de chaque distribution.

Les mesures faites après échantillonnage des trajectoires s'avèrent plus linéaires et les valeurs extrêmes de la vitesse ou du taux de virage sont réduites : L'échantillonnage des trajectoires permet de lisser l'impact des imprécisions liées aux échos radars et ces mesures serviront donc de référence à la modélisation des vitesses de roulage. L'écart type des distributions reste néanmoins relativement élevé pour les faibles valeurs du taux de virage (écart type autour de 3 pour une vitesse moyenne inférieure à 10 m/s) ce qui veut dire que les vitesses pratiquées au roulage sont très variables, surtout en ligne droite.

Globalement, les vitesses décroissent sensiblement lorsque le taux de virage augmente. Pour un taux de virage maximal autour de $2^\circ/m$ (correspondant à un rayon de virage d'une trentaine de mètres), l'écart type est inférieur à $\frac{1}{2}$ et les vitesses sont de l'ordre de 4 m/s. À Roissy, une particularité doit également être soulignée : les vitesses maximales en ligne droite ($\tau = 0$) sont beaucoup plus élevées qu'à Orly : elles dépassent 20 m/s (contre 14 m/s à Orly).

3.2.3 Modèle retenu

Vitesse associée à un taux de virage

Chaque voie de circulation de l'aéroport est décrite par une ligne brisée $(P_i)_{1 \leq i \leq n}$. On peut donc lui associer une longueur l et un taux de virage moyen τ : aux vues des mesures effectuées sur les

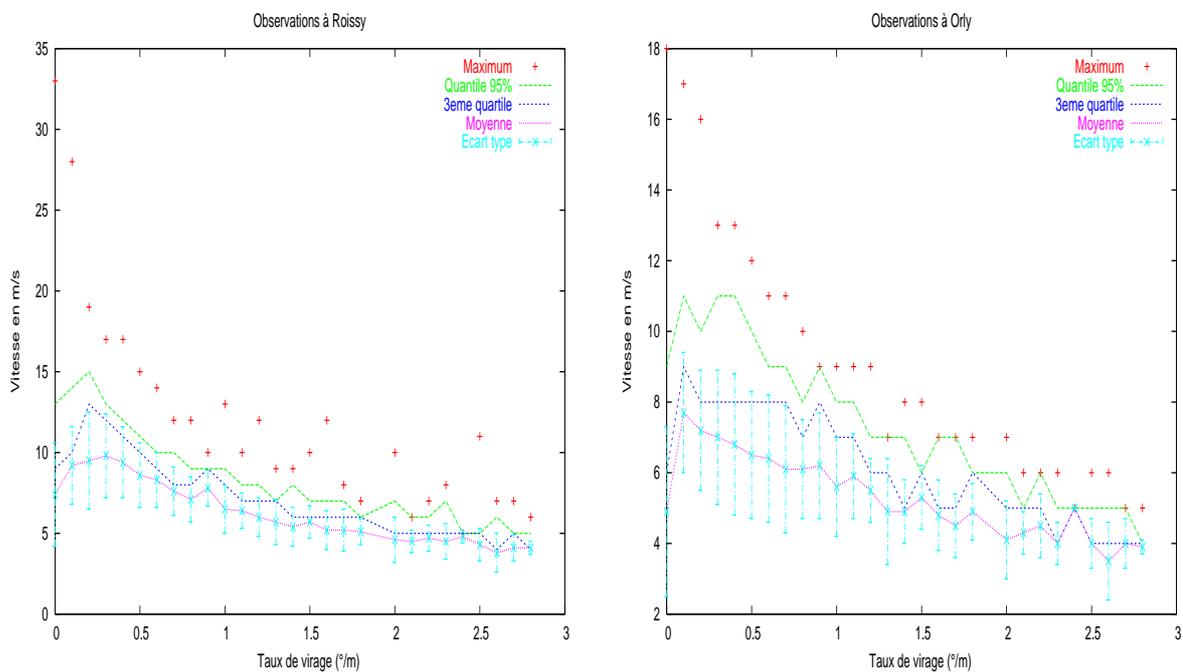


FIG. 3.2 – Mesures directes

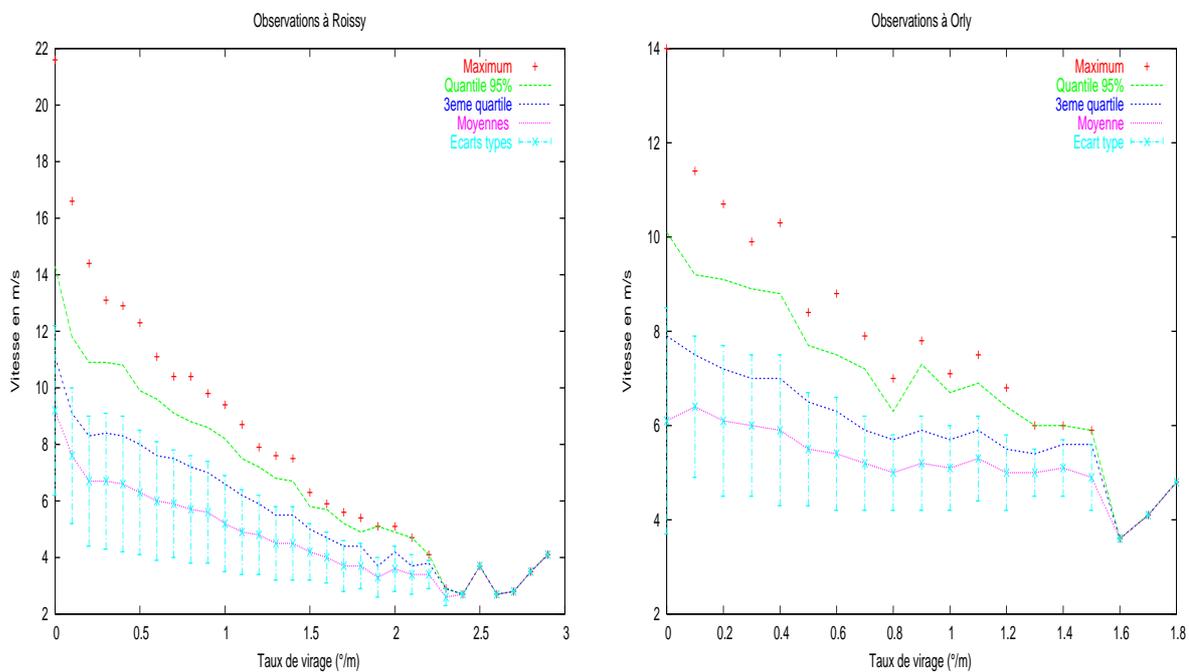


FIG. 3.3 – Mesures sur les trajectoires échantillonnées

trajectoires radar des avions, on modélisera la vitesse moyenne de roulage sur chaque portion comme une fonction v linéaire décroissante du taux de virage τ , bornée à une vitesse minimale V_{\min} :

$$l = \sum_{i=2}^n \|\overrightarrow{P_{i-1}P_i}\| \quad \text{et} \quad \tau = \frac{1}{l} \sum_{i=2}^{n-1} |(\overrightarrow{P_{i-1}P_i}, \overrightarrow{P_iP_{i+1}})|$$

$$\begin{cases} \text{Si } \tau < \tau_0 & v(\tau) = V_{\min} + (V_{\max} - V_{\min})(1 - \frac{\tau}{\tau_0}) \\ \text{Sinon,} & v(\tau) = V_{\min} \end{cases}$$

V_{\max} et V_{\min} , les vitesses maximales et minimales de roulage, ainsi que le taux de virage limite τ_0 (au dessus duquel la vitesse de roulage sera V_{\min}) seront des paramètres du simulateur et pourront être adaptés à l'aéroport.

Vitesses spécifiques

Certains endroits de l'aéroport nécessitent la définition de vitesses qui ne soient pas définies directement en fonction du taux de virage :

- **Pour une bretelle de sortie de piste**, une indication de vitesse maximale est fournie par la catégorie de la bretelle de sortie (au pas, lente, normale ou rapide). Chaque catégorie de bretelle de sortie sera donc associée à une vitesse spécifique constante. Ces vitesses seront également des paramètres du simulateur, éventuellement adaptables à chaque aéroport.
- **Pour les voies d'entrée de parking**, on supposera que les pilotes adoptent des vitesses réduites mais fonction malgré tout du(des) virage(s) à effectuer : la vitesse nominale sera alors modélisée comme la moitié de la vitesse $v(\tau)$ correspondant normalement au taux de virage τ .
- **Pour les sorties de parkings par push-back**, on supposera que cette manœuvre délicate implique une vitesse constante et réduite jusqu'au point de libération de l'avion. Cette vitesse sera donc également un paramètre du simulateur.

Ces précisions montrent que la modélisation des vitesses nominales demande la mise en place de nombreux paramètres, dont l'impact sur le trafic simulé peut être conséquent et dont les valeurs devront être adaptées judicieusement à l'aéroport. En général, une étude du trafic réel sur l'aéroport considéré sera nécessaire pour paramétrer correctement le modèle.

3.3 Graphe des voies de circulation

3.3.1 Définitions et formalisation

Dans cette partie, les voies de circulation de l'aéroport sont modélisées sous forme d'un graphe orienté $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, où \mathcal{N} est un ensemble fini de nœuds (ou sommets) et \mathcal{A} un sous-ensemble d'arcs de $\mathcal{N} \times \mathcal{N}$. Les éléments de \mathcal{A} sont associés à des coûts par une fonction c . Dans notre cas, ces coûts représentent des temps de parcours et sont donc positifs ou nuls, ce qui sera une hypothèse importante pour le choix et la mise en œuvre des algorithmes de parcours de graphe :

$$\mathcal{G} = (\mathcal{N}, \mathcal{A}) \quad \begin{array}{l} \mathcal{A} \subset \mathcal{N}^2 \xrightarrow{c} \mathbb{R}^+ \\ (m, n) \mapsto c_{mn} \end{array}$$

Les définitions classiques suivantes sont utilisées :

- **L'ordre** du graphe est le nombre de nœuds de \mathcal{N} .

- **Le degré** d'un nœud est le nombre d'arcs ayant ce nœud pour extrémité (en amont ou en aval).
- **Les nœuds fils** d'un nœud sont ceux qui lui sont reliés en aval par un arc de \mathcal{A} :

$$\mathcal{Fils}(n) = \{m \in \mathcal{N} \mid (n, m) \in \mathcal{A}\}$$

- Réciproquement, l'ensemble des **nœuds pères** d'un nœud, noté $\mathcal{Pères}(n)$, correspond à ceux qui lui sont reliés en amont.
- **Un chemin** d'un nœud source s vers un nœud terminal t est une suite finie (non vide) de sommets $(n_i)_{1 \leq i \leq l}$ telle que :

$$\begin{cases} n_1 = s \\ n_l = t \\ \forall i \in \{1, \dots, l-1\}, (n_i, n_{i+1}) \in \mathcal{A} \end{cases}$$

Un tel chemin est dit **sans boucle** si : $\forall (i, j) \in \{1, \dots, l\}^2, i \neq j \implies n_i \neq n_j$

Il est appelé **cycle** lorsque $s = t$.

Dans ce cadre, des algorithmes classiques de parcours de graphe pourront être appliqués pour obtenir un ensemble de chemins, éventuellement sans boucle, reliant deux nœuds donnés du graphe.

3.3.2 Le graphe *direct*

Définition

Un premier graphe orienté peut-être défini directement en fonction des éléments descriptifs de l'aéroport détaillés en annexe B.2.1. Ce graphe sera qualifié, pour cette raison, de *direct* :

- **Les nœuds du graphe** sont les éléments ponctuels de l'aéroport, c'est-à-dire les points nommés et les parkings ;
- **Les arcs** sont les éléments de liaison entre ces éléments ponctuels, c'est-à-dire les transitions, les voies d'accès aux parkings et les bretelles d'entrée ou de sortie de piste.
- **Le coût** $c(\mathcal{L})$ associé à un arc représentant une liaison \mathcal{L} correspond au temps de parcours de la liaison, calculé en fonction de la vitesse nominale de roulage (modélisée dans la partie précédente), éventuellement pondéré par les poids associés aux sens préférentiels de parcours donnés dans la description de l'aéroport (annexe B).

Taille du graphe

Le tableau suivant donne les caractéristiques des graphes directs construits à partir de la description des aéroports d'Orly et de Roissy.

Aéroport	Sommets (ordre)	Arcs	Degré moyen	Degré max.
Roissy	2181	3883	3.56	39
Orly	798	1716	4.3	42

Les ordres respectifs de ces graphes, qui peuvent paraître élevés, reflètent la taille des deux aéroports. Ces valeurs restent cependant tout à fait accessibles aux algorithmes de recherche de chemins pour les performances actuelles des ordinateurs : le temps de calcul de l'algorithme de Dijkstra (annexe D.3) devrait par exemple être inférieur à la seconde.

On peut remarquer la contradiction entre les faibles valeurs du degré moyen des graphes (autour de 4) et les fortes valeurs du degré maximal (autour de 40) : ceci est dû à la description des voies

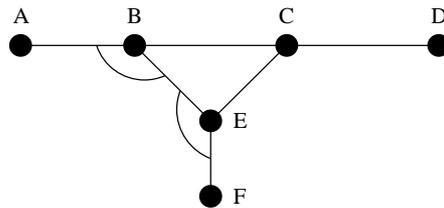


FIG. 3.4 – Virages non pris en compte

d'accès aux parkings, qui relie souvent des groupes de parkings voisins à un même point d'accès. Les nœuds concernés par ces degrés élevés sont donc rares et toujours proches des nœuds terminaux : la pénalisation engendrée (en complexité de calcul de chemins) devrait donc être réduite.

Problèmes posés

Bien que cette définition du graphe associé à l'aéroport découle logiquement de sa description topologique et aboutisse à une taille relativement faible, elle pose un certain nombre de problèmes pour la mise en œuvre des contraintes liées au cheminement des avions :

Les aires de piste ne sont pas prises en compte : des détours passant par ces aires pourraient être considérés comme des chemins alternatifs alors que cela n'est pas envisageable dans un contexte opérationnel : des coefficients de pénalité doivent être ajoutés à ces voies pour éviter leur utilisation lorsqu'elle n'est pas nécessaire.

L'interdiction de traverser un parking n'est pas formalisée : les parkings admettant des sorties en marche avant sont en effet des nœuds ordinaires du graphe et peuvent donc servir de point de passage de manière illicite. Ce problème peut être résolu en supprimant certains arcs du graphe en fonction du mouvement concerné (pour un départ, les parkings n'auraient plus de nœuds pères, et pour une arrivée, les parkings n'auraient plus de nœuds fils).

L'angle de virage entre deux liaisons n'est pas pris en compte, alors qu'il peut pénaliser la vitesse de roulage. Sur la figure 3.4 par exemple, l'enchaînement autorisé (A, B, E, F) sera associé à la vitesse de roulage en ligne droite (la vitesse maximale) et le temps de parcours du chemin sera donc sous-estimé.

L'impossibilité d'effectuer un demi-tour n'est pas formalisée : les transitions de l'aéroport sont en effet partitionnées en matrices et seuls les enchaînements de transitions de matrices distinctes sont admis, comme précisé en annexe B.2.1. Sur la figure 3.4, l'enchaînement (B, C, E) est normalement rendu illicite par cette règle.

Les deux derniers problèmes soulevés font apparaître que les arcs réellement accessibles en amont d'un nœud dépendent en réalité du nœud précédent dans le chemin courant. Ceci incite à considérer un nouveau graphe pour mieux représenter les voies de circulation de l'aéroport, comme décrit ci-après.

3.3.3 Le graphe *dual*

Définition

Pour formaliser toutes les contraintes de cheminement des avions, il s'avère nécessaire d'inclure les règles d'enchaînement des liaisons dans la définition même du graphe. Pour cela, chaque portion de taxiway de l'aéroport doit être considérée comme un nœud du graphe et non plus comme un arc.

Le graphe résultant se rapproche d'un graphe dual du précédent, si ce n'est qu'il sera restreint aux enchaînements autorisés :

- **Les nœuds du graphe** sont :
 - toutes les *liaisons* décrivant les voies de circulation (les transitions, les accès aux parkings et les bretelles de pistes).
 - les éléments ponctuels qui correspondent à des origines ou des destinations de chemins à chercher, c'est-à-dire tous les parkings et tous les points situés sur un axe de piste ;
- **Les arcs** sont définis comme suit :
 - Un couple de *liaisons* $(\mathcal{L}_1, \mathcal{L}_2)$ est un arc si le point de destination de \mathcal{L}_1 correspond au point d'origine de \mathcal{L}_2 et si l'enchaînement $(\mathcal{L}_1 \rightarrow \mathcal{L}_2)$ est autorisé :
 - Si \mathcal{L}_1 et \mathcal{L}_2 sont des transitions, leurs matrices doivent être distinctes.
 - Si \mathcal{L}_1 est une sortie de parking, le point destination de \mathcal{L}_2 ne doit pas faire partie des points interdits après \mathcal{L}_1 (selon les règles spécifiées dans la description de l'aéroport, fournies en annexe B.2.1). De même, si \mathcal{L}_2 est une entrée de parking, le point origine de \mathcal{L}_1 ne doit pas être un point interdit avant cet accès.
 - Un couple (p, \mathcal{L}) (resp. (\mathcal{L}, p)) est un arc si p est un parking ou un point de l'axe de piste et s'il est le point d'origine (resp. le point de destination) de la liaison \mathcal{L} .
- **Le coût** associé à un arc dépend de la nature du nœud source de cet arc : si ce nœud est un élément ponctuel, le coût sera nul, tandis que pour une liaison \mathcal{L}_1 , le coût représentera le temps de parcours de cet arc, pondéré par un coefficient multiplicatif $w_{\mathcal{L}_1}$ et augmenté d'un coût d'enchaînement des deux éléments lorsque le nœud fils de l'arc est une liaison \mathcal{L}_2 :
 - Si la liaison \mathcal{L}_1 interfère avec les aires de piste, $w_{\mathcal{L}_1}$ devra être suffisamment élevé pour pénaliser le passage dans ces aires lorsque d'autres chemins (qui seraient plus coûteux sans ces coefficients) permettent de les éviter. Le caractère multiplicatif de la pénalité permet de favoriser les chemins qui minimisent le temps d'occupation des aires de piste (ce qui favorisera l'utilisation des sorties rapides de piste).
 - Pour les autres liaisons (les transitions), $w_{\mathcal{L}_1}$ sera logiquement le poids associé à la transition si l'on veut prendre en compte les sens préférentiels de circulation définis sur l'aéroport.
 - Enfin, le coût d'enchaînement de $(\mathcal{L}_1 \rightarrow \mathcal{L}_2)$ est défini par une fonction linéaire croissante de l'angle de virage mesuré entre les deux liaisons.

Taille du graphe

Le tableau suivant donne les caractéristiques du graphe dual pour les deux aéroports étudiés :

Aéroport	Sommets (ordre)	Arcs	Degré moyen	Degré max.
Roissy	4286	7224	3.37	31
Orly	1962	3756	3.83	25

On constate que la taille du graphe (nombre de nœuds et nombre d'arcs) est approximativement le double de celle du graphe *direct*. Cependant, les degrés moyens et maximaux ont diminué, car seuls les enchaînements autorisés sont modélisés.

Les figures 3.5, 3.6, 3.7 et 3.8 donnent une représentation des graphes de Roissy et d'Orly, avec les sens préférentiels correspondant à leurs configurations Est ou Ouest.

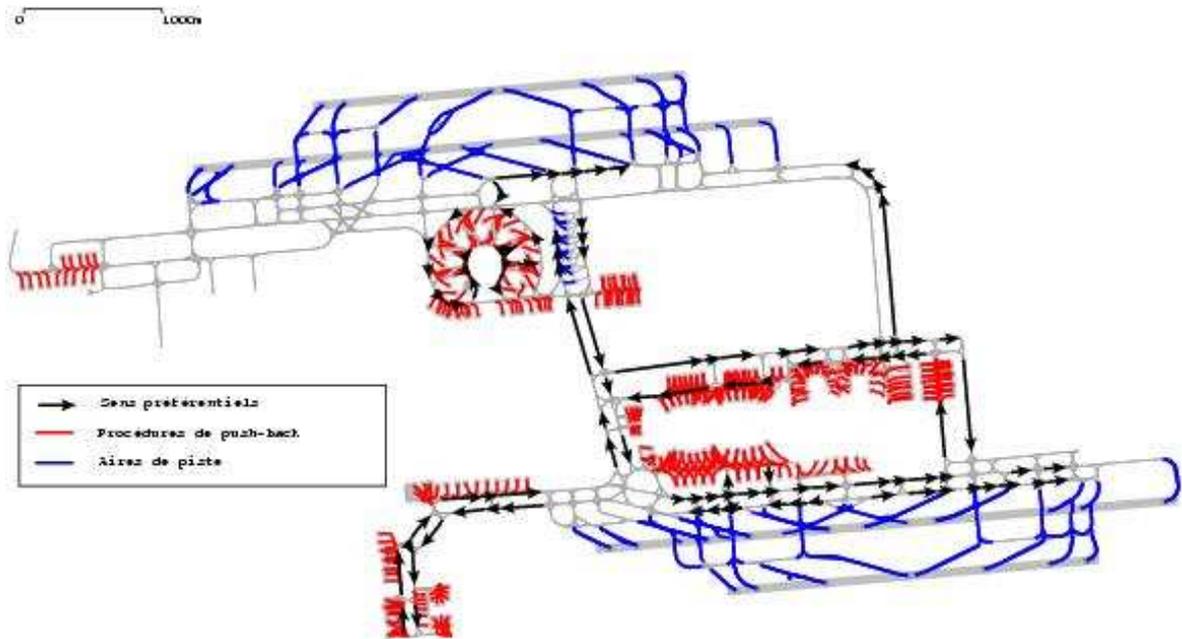


FIG. 3.5 – Roissy face à l'Ouest

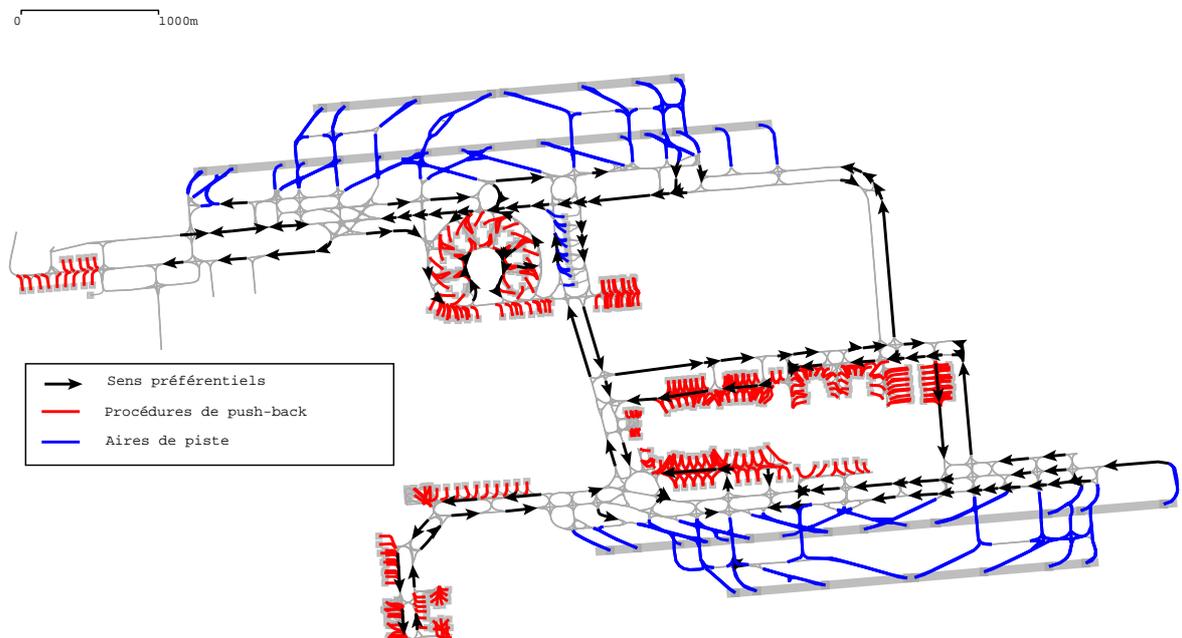


FIG. 3.6 – Roissy face à l'Est

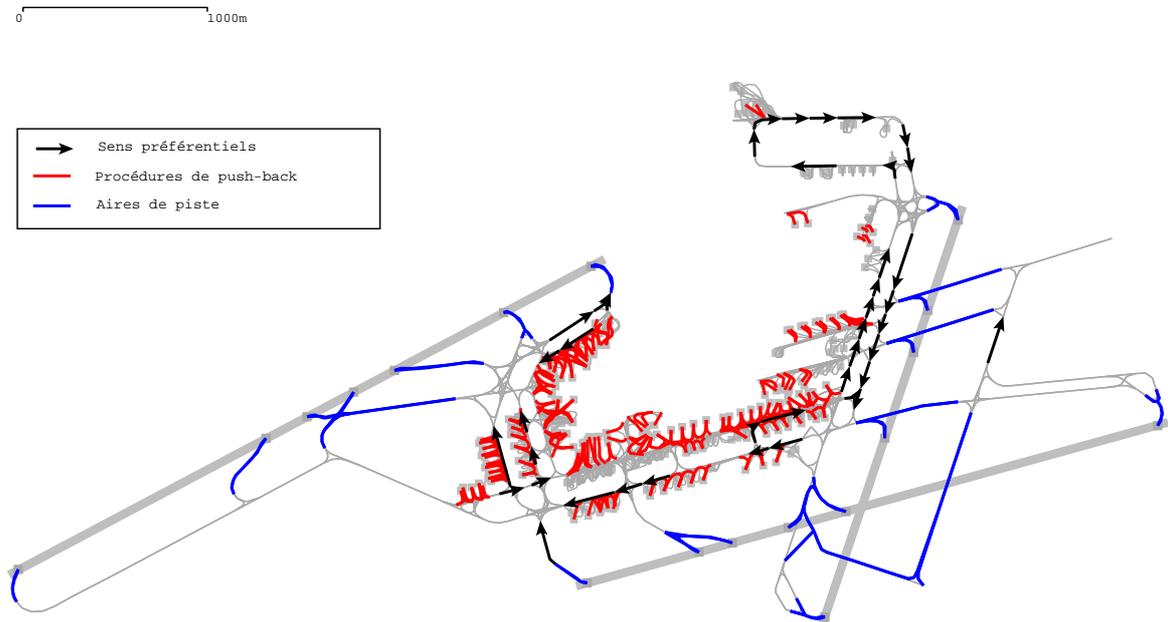


FIG. 3.7 – Orly face à l'Ouest

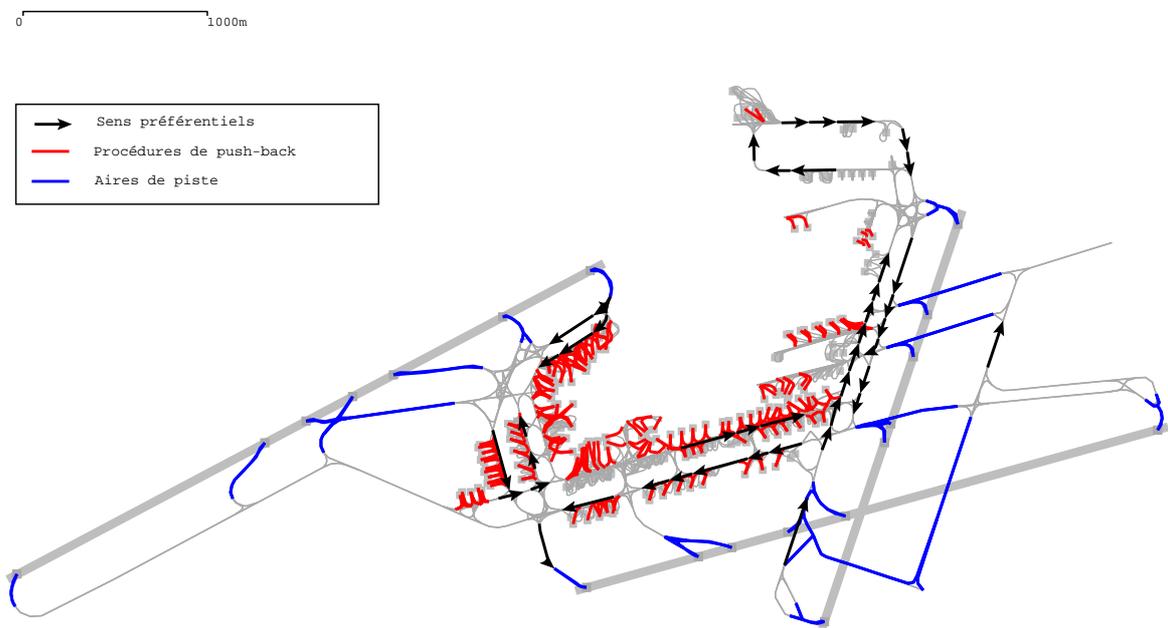


FIG. 3.8 – Orly face à l'Est

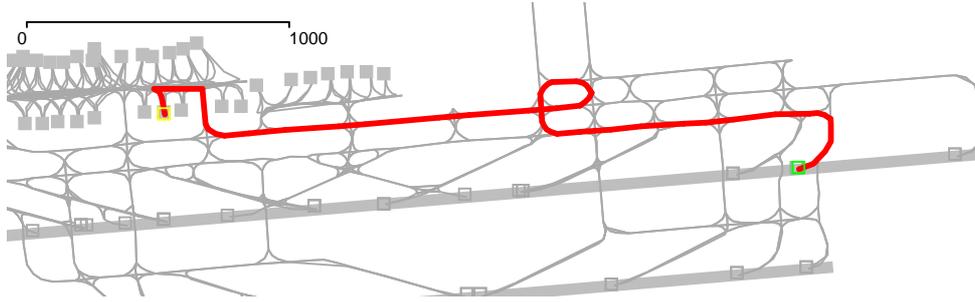


FIG. 3.9 – Exemple de boucle ne passant jamais deux fois par le même nœud

3.3.4 Contraintes de validité des chemins

Tel qu'il est défini, le graphe *dual* de l'aéroport formalise directement la plupart des contraintes de validité des chemins :

- Les enchaînements illicites de liaisons ne font pas partie des arcs du graphe : tous les chemins trouvés seront opérationnellement acceptables, s'ils ne contiennent pas de boucle.
- La gestion des sens préférentiels et l'évitement des aires de piste sont modélisés par les coefficients multiplicatifs w_i appliqués aux temps de parcours nominaux t_i des liaisons concernées. Ces coefficients devront être suffisamment pénalisants pour que les sens défavorables et les aires de pistes ne soient empruntés que lorsqu'aucun autre chemin ne permet de les éviter :

$$w_i t_i \gg t_M \quad \text{où } t_M \text{ est le temps de parcours maximal d'un chemin}$$

Cependant, l'élimination des chemins contenant des boucles ne peut se faire que par l'analyse géométrique des *transitions* de l'aéroport, car les nœuds sont des liaisons et non plus des points de passage : un chemin peut contenir une boucle au sens intuitif (géométrique) du terme, sans réellement passer deux fois par le même nœud du graphe, comme illustré figure 3.9. La contrainte de validité des chemins s'exprime donc en fonction des aspects géométriques de chaque liaison : un chemin sera valide si l'intersection géométrique (notée \cap) de chaque couple de ses liaisons non adjacentes est vide :

$$(n_i)_{1 \leq i \leq l} \text{ valide} \iff \forall (i, j) \in \{1, \dots, l\}, i < j - 1 \implies n_i \cap n_j = \{\}$$

La prise en compte de cette contrainte dans la recherche de chemins peut être implémentée de deux façons : soit en "épurant" les chemins trouvés (après coup), soit en adaptant les algorithmes de façon à restreindre leur recherche aux chemins valides. La deuxième solution semble naturellement plus efficace, bien qu'elle ne soit pas toujours réalisable. Elle utilisera une relation booléenne de compatibilité (géométrique) entre un nœud n du graphe et un chemin γ supposé valide, qui sera notée $n \cap \gamma = \{\}$: elle signifiera que le nœud n peut être ajouté en aval du chemin et former ainsi un nouveau chemin, noté $(n \cdot \gamma)$ valide :

$$n \cap \gamma = \{\} \iff (\gamma \text{ valide} \implies (n \cdot \gamma) \text{ valide})$$

Dans ce cadre, les algorithmes construisant les chemins nœud par nœud pourront parfois être adaptés : il suffira de n'autoriser l'ajout d'un nœud à un chemin que lorsque ce nœud est compatible avec le chemin. Cette opération de vérification sera cependant coûteuse en temps de calcul : elle nécessitera $\mathcal{O}(l)$ calculs d'intersections géométriques.

3.4 Conclusion

L'étude des traces radar des avions montre que leur vitesse de roulage peut être modélisée par une fonction linéaire décroissante du taux de virage. Les voies de circulation de l'aéroport sont alors représentées par un graphe orienté, dont les coûts représentent initialement des temps de parcours. Ces coûts peuvent être pondérés pour modéliser les sens préférentiels de circulation et la gêne occasionnée par le passage d'un avion dans les aires de piste.

La structure même du graphe permet ainsi de représenter presque toutes les contraintes liées à la circulation des avions : seule l'absence de boucle, au sens géométrique du terme, est définie comme une contrainte supplémentaire que devront respecter les chemins qui leur seront proposés. La recherche de chemins proprement dite demandera l'application d'algorithmes adaptés au problème, comme cela sera décrit au chapitre 6.

Chapitre 4

Modélisation du trafic

4.1 Introduction

La modélisation du trafic au sol se définit en fonction des éléments de l'aéroport décrits au chapitre précédent et sera à la base de toutes les mesures qui pourront être effectuées par simulation. Deux objectifs principaux déterminent le cadre dans lequel cette modélisation doit s'inscrire :

- Le premier est l'élaboration de méthodes d'optimisation pour la gestion du trafic au sol. Ce point s'inscrit directement dans le cadre du ASMGCS (Advanced Surface Movement Guidance and Control Systems), qui donne une priorité au développement d'outils d'aide au contrôle au sol et à la mise en œuvre de systèmes automatiques de surveillance du trafic. À plus ou moins long terme, les méthodes d'optimisation doivent être intégrables dans le système opérationnel, ce qui ne sera possible que si la modélisation du trafic représente correctement toutes les contraintes et les inconnues relatives à un environnement en temps réel.
- Le second est la simulation en temps accéléré d'une journée de trafic sur l'aéroport : il s'agit de représenter l'évolution du trafic de manière aussi réaliste que possible, pour effectuer des mesures globales sur la performance du système (débits, chemins les plus empruntés, points de congestion, délais, etc...). Ces mesures, qui seront souvent spécifiques à chaque simulation, peuvent servir par exemple à évaluer l'impact de nouvelles infrastructures prévues sur l'aéroport ou à comparer entre elles des organisations opérationnelles différentes.

D'une manière générale, le trafic est initialement décrit par les intentions des compagnies, données sous forme de plans de vol déposés en avance. Les données décrivant ces plans de vol (détaillées en annexe B) ne sont pas toujours complètes mais des traitements permettent d'en extraire un échantillon de trafic relativement réaliste (annexe C). Dans ce chapitre, les intentions des compagnies sont donc supposées connues et cohérentes avec la description de l'aéroport. Les premières parties élaborent des modèles relativement classiques de prévision de trajectoires et de détection de conflits pour le trafic au sol. La dernière partie décrit l'assemblage de ces différents modèles dans un environnement de simulation, permettant de représenter l'écoulement du trafic pendant toute une journée.

4.2 Prédiction de trafic

4.2.1 Définition et paramètres de la prédiction

Dans un souci de réalisme et d'intégration potentielle dans un système en temps réel, le modèle de prédiction doit traiter des situations de trafic telles qu'elles se présentent au contrôle au sol : une

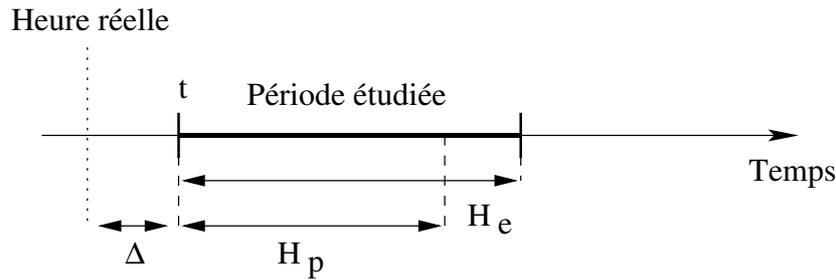


FIG. 4.1 – Paramètres de la prédiction

situation est décrite par les positions initiales des avions et par l'ensemble des trajectoires envisagées dans un futur proche, qui sera appelé l'horizon de prédiction.

Résoudre la situation consistera à choisir les *trajectoires* des avions sur la période considérée, de telle sorte qu'elles respectent les *règles de séparation* entre avions. La qualité de la *résolution* se mesurera en fonction d'un *critère global d'optimisation* qui sera défini au prochain chapitre.

En temps réel, les avions continuent d'avancer pendant que le système effectue la résolution de la future situation : le système doit donc toujours être en avance sur la réalité et doit se mettre-à-jour régulièrement avec les nouvelles positions des avions. Ce modèle classique de prédiction [Durand 96a], intégrable dans un système en temps réel, fait ainsi intervenir deux paramètres principaux (illustrés figure 4.1) :

- **L'horizon de prédiction** H_p définit l'anticipation du système : il fixe la largeur de l'intervalle de temps sur lequel chaque situation est optimisée. Ce paramètre doit être choisi judicieusement, comme un compromis entre le réalisme et la qualité des solutions qui seront trouvées : une valeur trop élevée aboutit à des situations complexes (difficiles à optimiser) et généralement caduques, car relatives à des événements qui seraient imprévisibles dans un contexte opérationnel. En revanche, une valeur trop faible correspond à une vision restreinte du trafic et biaise le problème d'optimisation. En pratique, H_p dépendra de l'incertitude qui sera envisagée sur les vitesses de déplacement : sa valeur sera comprise entre 5 minutes (pour une incertitude élevée) et 10 minutes (incertitude très faible).
- **L'avance Δ du système sur la réalité** sera également son pas de rafraîchissement. Ce paramètre doit être compatible avec l'horizon de prédiction :

$$\Delta < H_p$$

Dans un contexte opérationnel, Δ doit être un majorant du temps nécessaire à chaque résolution (il limite donc la profondeur de la recherche d'une solution). Par contre, plus sa valeur est faible, plus les décisions prises au cours des résolutions passées pourront être remises en cause. Ce paramètre fixe donc le compromis entre la réactivité du système et la qualité des solutions qu'il propose (en pratique, sa valeur sera comprise entre 1 et 3 minutes).

4.2.2 Effet de l'horizon

Le caractère borné de l'horizon de prédiction est une conséquence des incertitudes inhérentes au trafic aérien. Cet aspect peut pourtant causer un problème pour la résolution des situations : le système n'a qu'une vision à court terme du trafic et l'optimisation de la situation peut en être biaisée.

- Ce phénomène, appelé *effet de l'horizon*, peut notamment engendrer des situations sans solution. Pour le trafic au roulage, il peut s'agir par exemple de deux avions amenés l'un en face de

l'autre, à la fin de la fenêtre temporelle étudiée (le système ne détecte pas le problème mais par la suite, le problème ne pourra plus être résolu).

- Plus généralement, l'effet de l'horizon est accentué par des événements majeurs (c'est-à-dire qui modifient considérablement l'évaluation des solutions) risquant de se produire juste après la période étudiée. Au niveau de l'aéroport, l'atterrissage d'un avion en est un bon exemple : la piste doit être impérativement libre et les séparations avec le décollage précédent doivent être respectées. La prise en compte de cet atterrissage peut donc modifier totalement la séquence de piste optimale pour la situation courante.

Pour palier l'effet de l'horizon, il est souvent nécessaire de considérer les situations à venir sur un horizon de prédiction étendu, mais de manière moins détaillée : ceci permet d'ajuster le critère d'optimisation pour la situation courante ou d'ajouter des contraintes supplémentaires au problème (ce point sera détaillé dans la partie 5.2.2).

Dans ce cas, l'intervalle de temps supplémentaire qui sera étudié de manière plus macroscopique sera délimité par un horizon de prédiction *étendu* H_e (figure 4.1) :

$$\mathcal{I}_{supp} = [t + H_p; t + H_e]$$

Sur cet intervalle de temps, seuls seront considérés les événements majeurs (départs du parking et atterrissages par exemple) : les positions précises des avions ne seront pas calculées.

4.2.3 Modèle de trajectoire

Discretisation du temps

Deux principales approches peuvent être employées pour représenter la situation sur l'horizon de prédiction :

- L'approche événementielle consiste à calculer et à classer l'ensemble des événements susceptibles de survenir pendant cette période. Chacun d'eux est daté et leur séquence ordonnée donne une description de la situation.
- L'approche discrète consiste à diviser le temps en périodes élémentaires et à décrire l'état de la situation à chaque pas de temps ainsi formé.

La représentation détaillée du trafic au roulage servira en particulier à vérifier le respect des séparations entre avions à chaque instant (et cette opération s'avérera la plus pénalisante, comme cela sera abordé dans la partie 4.3). La deuxième approche semble pour cela la plus adaptée, puisqu'elle fournit directement toutes les positions des avions à chaque pas de temps, ce qui permet d'accélérer et de simplifier le calcul des distances.

Le temps sera donc discrétisé en périodes élémentaires : un paramètre δ définit la largeur des pas de temps considérés. Pour assurer la pertinence du calcul des distances entre avions par rapport aux normes de séparation, les pas de temps doivent être suffisamment fins :

$$\delta < \frac{D}{V_{\max}}$$

Où D est la distance minimale imposée entre avions et V_{\max} leur vitesse maximale de roulage, définie dans la partie 3.2.3.

Trajectoire initiale

Dans une situation, différents chemins sont envisageables pour chaque avion (la recherche de ces chemins fait partie de la résolution de la situation, comme cela sera décrit au chapitre 6). Initialement,

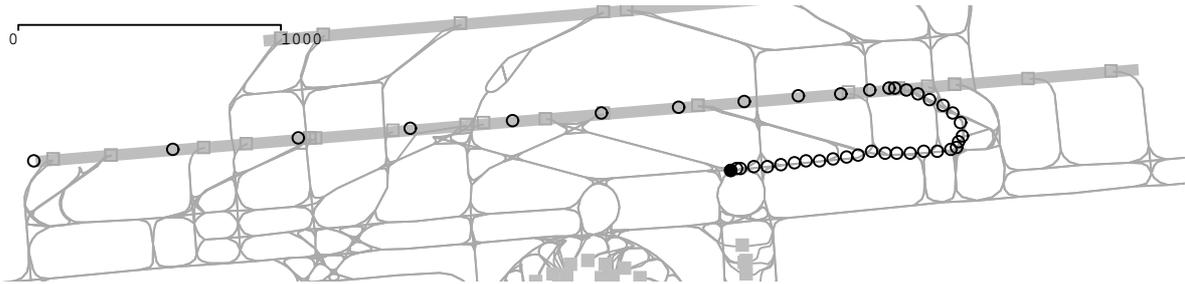


FIG. 4.2 – Trajectoire discrétisée sans attente

l'avion est supposé suivre le plus court d'entre eux, sans s'arrêter : ceci définit la *trajectoire initiale* de l'avion, qui est la trajectoire qu'il est censé avoir sur l'horizon de prédiction sans qu'aucun ordre de contrôle ne lui soit donné.

La trajectoire initiale d'un avion doit rester cohérente avec le critère d'optimisation : c'est en effet à partir de cette trajectoire que seront détectés les problèmes à résoudre pour la suite de la simulation. Elle doit donc correspondre à la *meilleure* trajectoire pour l'avion (celle qu'il devrait suivre si aucun autre avion n'était présent). Dans le modèle, toutes les décisions prévues par la résolution précédente (à l'heure $t - \Delta$ et sur l'intervalle de temps $[t; t + H_p - \Delta]$) sont donc systématiquement annulées dans la prédiction de trajectoire relative à la nouvelle situation, concernant la période $[t + \Delta, t + H_p]$.

En accord avec la discrétisation du temps définie dans la partie précédente, la trajectoire d'un avion est une suite finie $(p_t)_{0 \leq t \leq H_p}$ de positions ponctuelles (où l'horizon de prédiction H_p est exprimé en nombre de pas de temps élémentaires). L'espacement de ces positions ponctuelles est naturellement proportionnel à la vitesse de déplacement en chaque point (figure 4.2) :

- Au roulage, la trajectoire est calculée en fonction des vitesses nominales définies dans la partie 3.2.
- Pendant les phases de décollage et d'atterrissage proprement dites, la position exacte des avions sur la piste a une importance pour les autres avions, comme cela sera décrit dans la partie 4.3.1. Les trajectoires pendant ces phases sont donc modélisées en fonction de deux paramètres a_{dec} et a_{att} qui fixent respectivement l'accélération pendant le décollage et la décélération pendant l'atterrissage.

Manœuvres envisageables

Les manœuvres envisagées pour chaque avion permettent de générer l'ensemble des trajectoires susceptibles de lui être imposées pour assurer les séparations et optimiser la situation. Trois catégories de manœuvres sont modélisées. Elles sont calquées sur les ordres classiques utilisés par les contrôleurs au sol :

- **Une modification du chemin à suivre** : dans ce cas, un nouveau chemin est imposé à l'avion, choisi parmi l'ensemble des chemins restant encore envisageables pour cet avion.
- **Un ordre d'attente**, sur une position particulière du chemin, jusqu'à une heure de fin d'attente :
 - Si l'avion atteint la position après l'heure de fin d'attente, il n'attend pas. Cette particularité prendra toute son importance pour la prévision de trafic avec incertitudes.
 - La position d'attente peut également être au parking pour un départ, ce qui correspond à retarder l'heure effective du départ. Dans ce cas, l'occupation du parking par l'avion est prolongée.
 - Certaines positions correspondent à des procédures qu'il est impossible d'interrompre et

ne peuvent donc pas être prises pour position d'attente : le décollage, l'atterrissage ou une procédure de *push-back*.

- **Un ordre de ralentissement** en finale pour une future arrivée : cette action est très importante sur les grands aéroports pour l'optimisation des séquences de piste. Cependant, elle est limitée et doit être anticipée pour être applicable : elle correspond à avancer la sortie des « pleins volets » avant l'atterrissage et elle doit être considérée comme définitive (l'avion ne pourra plus accélérer par la suite). Un paramètre λ définit pour cela le taux maximal de réduction de vitesse en finale (25% en pratique). Dans le modèle, ce ralentissement revient à retarder l'heure d'atterrissage. Si l'ordre de ralentissement est donné d minutes avant l'heure d'atterrissage initialement prévue, le retard maximal qui pourra être imposé à l'arrivée avant l'atterrissage sera donc borné par l'horizon de prédiction :

$$D_{\max}(d) = \lambda d \quad \text{donc : } D_{\max}(d) < \lambda H_p$$

Remarque : Dans la réalité, une véritable *attente en l'air* peut être imposée aux arrivées (procédures de *stack*), mais ces opérations sont à la charge du contrôle d'approche. Elles sont modélisées par le traitement initial des heures d'atterrissage (annexe C.4) : une fois en finale, si l'arrivée ne peut pas être retardée suffisamment pour bénéficier de la piste, seule une remise de gaz peut être envisagée (mais cette manœuvre d'urgence ne sera pas modélisée).

Les manœuvres ainsi définies peuvent naturellement être combinées (changement de chemin avec attente ou plusieurs attentes à la suite). Une limite pourra être fixée sur le nombre de positions d'attente impossibles par avion dans chaque situation, pour rendre exploitables les actions si elles devaient être entreprises dans un contexte opérationnel.

4.2.4 Incertitudes sur la vitesse

Définition

La prévision de trafic est nettement plus réaliste si elle tient compte des incertitudes liées aux futures positions des avions : les solutions trouvées seront plus robustes aux écarts de trajectoires (pour un système opérationnel, la prise en compte des incertitudes est une nécessité absolue).

Au roulage, chaque avion se voit contraint de suivre un chemin imposé par le contrôle au sol. Bien que des erreurs de cheminement puissent avoir lieu (surtout sur des grands aéroports comme Roissy), ces cas extrêmes ne peuvent pas être raisonnablement anticipés lors de l'optimisation des situations (il serait impossible d'assurer les séparations dans de telles conditions) : les avions seront supposés suivre rigoureusement leur chemin.

Par contre, aucune vitesse de roulage n'est réellement imposée, ce qui rend les vitesses réelles extrêmement variables, comme cela a été étudié à partir des traces radar dans la partie 3.2.

Un modèle d'incertitude sur la vitesse de roulage est donc élaboré : à chaque position p de l'avion, l'écart entre sa vitesse réelle $v(p)$ et la vitesse nominale $v_0(p)$ prévue à cet endroit est bornée par un taux d'incertitude δ_v . Les vitesses nominales étant considérées comme des vitesses maximales de roulage, le modèle d'incertitudes s'écrit :

$$\forall p, (1 - \delta_v)v_0(p) \leq v(p) \leq v_0(p)$$

Il faut cependant restreindre les incertitudes aux positions de roulage seulement : de telles incertitudes seraient inadaptées aux aires de piste, où les avions doivent « expédier » la libération de la piste

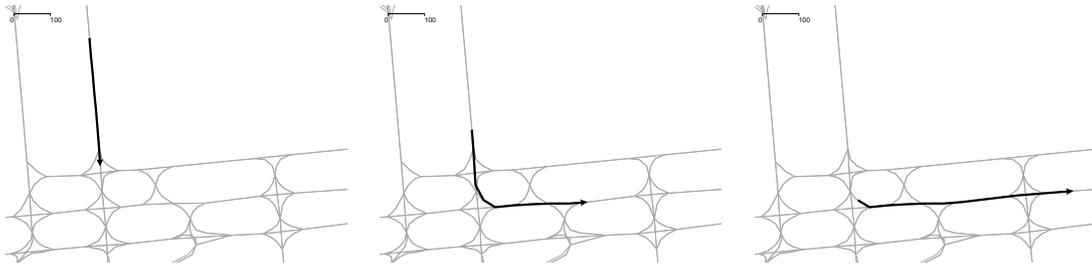


FIG. 4.3 – Positions d'incertitudes

(au décollage, à l'atterrissage, ou pendant la traversée d'une piste) pour garantir l'écoulement correct du trafic. De plus, les normes de séparation pour les séquences de piste sont définies en temps et sont fonction des turbulences de sillage (et non des distances entre avions). L'incertitude sur la vitesse sera donc considérée comme nulle dans les aires de piste.

Position d'un avion avec incertitudes

Dans la prévision de trafic, le taux d'incertitude sur la vitesse transforme la position d'un avion en une ligne brisée de positions possibles, le long d'une portion de son chemin (figure 4.3). Cette ligne brisée sera appelée la zone d'incertitude de l'avion :

- La position la plus avancée correspond à la position qu'aurait l'avion s'il avait toujours roulé à la vitesse nominale prévue, tandis que la position la plus reculée correspond à l'adoption de la vitesse minimale.
- La longueur totale de cette ligne brisée varie en fonction du temps et de la vitesse nominale : sur les portions à vitesse constante (une ligne droite par exemple), la zone d'incertitude augmente linéairement avec le temps. Elle peut au contraire diminuer lorsque l'avion ralentit pour tourner.
- Au début de la période de prédiction (état initial de la situation), les zones d'incertitude des avions ne sont pas forcément ponctuelles car le système étudie toujours la situation Δ minutes en avance [Granger 02]. La longueur maximale l_{\max}^0 des zones d'incertitude initiales est donc :

$$l_{\max}(0) = \Delta \delta_v V_{\max}$$

Plus généralement, la longueur maximale $l_{\max}(t)$ de la zone d'incertitude d'un avion à l'instant relatif t est donnée par :

$$l_{\max}(t) = (\Delta + t) \delta_v V_{\max}$$

La discrétisation du temps transforme cette relation en un nombre maximum de positions possibles, si Δ et t sont exprimés en nombre de pas de temps élémentaires :

$$n_{\max}(t) = 1 + E(\delta_v(\Delta + t)) \quad (\text{où } E \text{ désigne la partie entière})$$

Cette relation signifie que pour garder une qualité de prédiction constante, la taille de l'horizon de prédiction doit être considérée comme inversement proportionnelle à l'incertitude δ_v .

- Dans les aires de piste, l'absence d'incertitude rend la longueur de la ligne brisée uniquement dépendante de la vitesse nominale : une arrivée conservera une position ponctuelle pendant toute la phase d'atterrissage, jusqu'à la libération de la piste. Au contraire, un avion au départ conservera, pendant toute la durée du décollage, l'amplitude (en temps) de la zone d'incertitude qu'il avait au moment d'entrer sur la piste. Il en sera de même pour un avion traversant une piste.

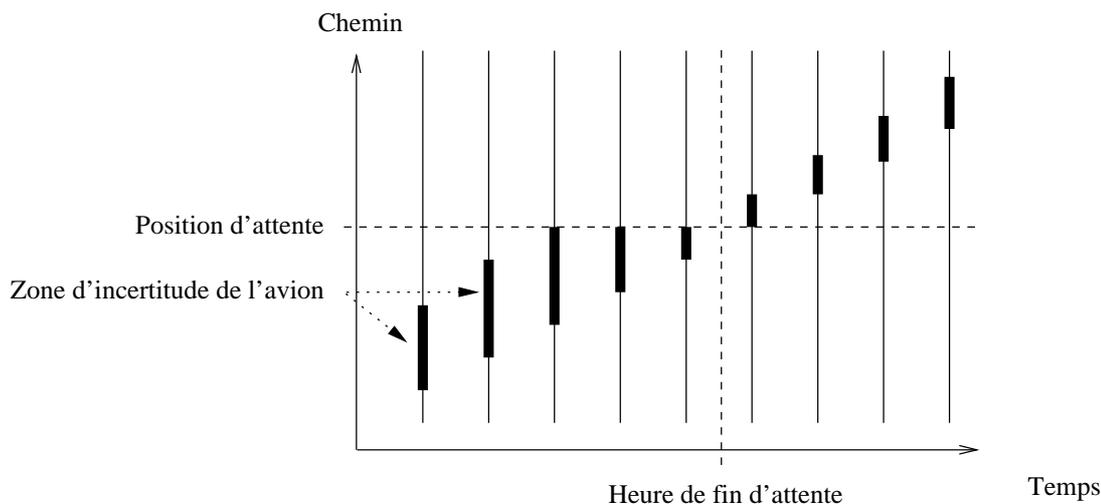


FIG. 4.4 – Réduction des incertitudes sur les points d'attente

Évolutions de l'incertitude pendant les attentes

Un ordre d'attente imposé à un avion réduit naturellement sa zone d'incertitude pendant la phase d'attente, car l'heure de fin d'attente ne dépend pas de l'heure d'arrivée sur le point d'attente (figure 4.4). Ceci modélise la double utilité des ordres d'attente pour le trafic au sol : ils servent d'une part à assurer les séparations entre avions, et d'autre part à rendre la prédiction de trafic plus précise.

La réduction de l'incertitude pendant les attentes est en réalité indispensable pour l'optimisation des séquences de piste : la mise en attente des départs devant la piste (au point d'arrêt) rend la zone d'incertitude de l'avion ponctuelle avant le décollage, ce qui est la seule façon de maximiser l'utilisation de la piste.

4.3 Détection de conflits

4.3.1 Règles de séparation

Dans cette partie, les notions classiques de conflits entre une paire d'avions et de détection de ces conflits par un modèle prédictif ([Durand 96a], [Granger 02]) sont reprises et adaptées au trafic au sol sur un aéroport.

L'objectif est donc de détecter l'ensemble des violations de normes de séparation prévisibles, étant donné une description de la situation, ou plus précisément un ensemble de trajectoires et de manœuvres imposées sur l'horizon de prédiction. Ceci nécessite la définition d'un modèle de séparation :

- Le modèle de séparation est généralement une simplification des règles réelles : certaines de ces règles peuvent en effet dépendre de phénomènes non modélisés comme les conditions météorologiques (les règles de séparation liées aux turbulences de sillages des avions peuvent par exemple être adaptées en fonction du vent) ou de certaines pratiques opérationnelles qui seraient difficiles à définir rigoureusement. Pour éviter que cette simplification rende les solutions caduques, il faut veiller à ce que le modèle reste toujours plus pénalisant que les règles réelles.
- Les méthodes de résolution doivent faire le moins d'hypothèses possible sur le modèle de séparation, afin que l'affinement ou l'adaptation de ce dernier ne modifie pas leur efficacité.

- Le modèle de séparation dépend du cadre étudié : pour le trafic au sol, on peut notamment distinguer les séparations au roulage seulement (assurée par la *fréquence SOL*, en dehors de toute aire de piste) et les séparations imposées pour les séquences de piste (gérées par la *fréquence LOC*). Il est cependant difficile et pénalisant de séparer les deux problèmes, qui ne sont pas indépendants :
 - Pendant les périodes chargées de la journée, les files d’attente d’avions au décollage peuvent atteindre des tailles conséquentes et bloquer certaines voies de circulation éloignées des aires de piste.
 - L’ordre dans lequel les départs se présentent à la piste est le résultat de toute leur phase de roulage et modifie considérablement les retards engendrés.

Pour ces raisons, toutes les règles de séparation définies pour le trafic au sol seront modélisées et classées en trois catégories : les règles au parking, au roulage et dans les aires de piste. En fonction des objectifs d’une simulation, chacune des catégories pourra éventuellement être occultée.

Séparations au parking

Les avions garés sont supposés isolés du reste du trafic (dans la plupart des cas, les voies de circulation sont prévues pour éviter tout problème avec ces avions). Seule l’occupation du parking est donc considérée : deux avions ne peuvent occuper un parking au même moment.

Les dépendances pouvant exister entre parkings ne seront donc pas modélisées par des règles de séparation (car ces dépendances ne sont pas décrites, comme cela est analysé en annexe B.2.2). La cohérence des parkings affectés aux avions ne sera donc pas remise en cause et sera supposée éviter ces problèmes.

Séparations au roulage

Au roulage, une norme de séparation en distance D est requise : la distance entre deux avions doit toujours être supérieure à cette norme (la valeur communément admise dans les simulateurs d’aéroport étant de 60 mètres). En réalité, la norme de séparation peut être réduite sur certaines voies de circulation particulières, en fonction du type des avions (annexe A.2). Cet aspect pourrait facilement être modélisé (le type des avions est connu) mais ces voies ne sont pas décrites.

Seule la norme de séparation D est donc considérée dans la version actuelle du simulateur, ce qui modélise une règle générale de séparation plus restrictive que la réalité.

Séparations dans les aires de piste

Dans les aires de piste, les règles de séquençement suivantes viennent s’ajouter à la norme de séparation en distance D définie dans la partie précédente :

1. **Un seul décollage ou atterrissage à la fois** est autorisé sur chaque piste.
2. **Une séparation en temps** est imposée entre chaque décollage ou atterrissage. Ce temps sera défini en fonction du type (décollage ou atterrissage) et de la catégorie de turbulence de sillage (léger, moyen ou lourd) des deux avions impliqués, comme cela est décrit par le tableau 4.1. Le modèle correspond aux règles « standard » de séparation : les mesures minimales observées lui sont inférieures, sauf pour les catégories L , dont le nombre est trop faible pour que les observations soient représentatives.
3. **La circulation dans les aires de piste** est autorisée derrière l’éventuel décollage ou atterrissage en cours. Ceci concerne les départs rejoignant l’axe de piste ou les avions traversant la piste au

1 ^{er} avion		Observées (24/03/2000) / Modèle (secondes)					
		2 ^{ème} avion					
		Atterrissage			Décollage		
	L	M	H	L	M	H	
Att.	L	105 / 60	80 / 60	125 / 120	— / 60	55 / 60	70 / 120
	M	80 / 120	65 / 60	50 / 60	65 / 120	50 / 60	55 / 60
	H	— / 180	85 / 90	95 / 90	— / 60	60 / 60	70 / 90
Dec.	L	— / 60	55 / 60	— / 60	95 / 60	50 / 60	60 / 60
	M	80 / 120	55 / 60	45 / 60	50 / 60	45 / 60	55 / 60
	H	70 / 180	75 / 120	85 / 120	130 / 180	115 / 120	80 / 120

TAB. 4.1 – Séparations minimales (sec.) pour les séquences de piste

roulage. Cette règle autorise donc l'alignement multiple des avions sur la piste, comme cela est souvent pratiqué par les contrôleurs pour optimiser une séquence de décollage.

4.3.2 Conflits entre avions

Deux avions seront dit *en conflit* à chaque fois que leurs trajectoires prévues ne respectent pas l'une des règles de séparation. Chaque *conflit* peut être caractérisé par :

- La paire d'avions concernée ;
- L'instant (le pas de temps) du conflit ;
- La nature du conflit (au parking, au roulage ou sur la piste).

Ces informations pourront être utiles pour orienter la recherche de solutions (par l'intermédiaire d'heuristiques) ou adapter le critère d'optimisation aux situations conflictuelles.

4.3.3 Gestion des incertitudes

Présentation

La prise en compte des incertitudes sur la vitesse augmente la complexité de la détection de conflits mais ne modifie en rien les règles de séparation : pour être séparés, deux avions doivent pouvoir se trouver n'importe où dans leur zone d'incertitude sans jamais être en conflit. Cette notion permet de transposer les règles de séparation ponctuelles en règles de séparation avec incertitudes : deux avions sont séparés si leurs zones d'incertitude le sont.

La figure 4.5 illustre dans ce cadre les trois catégories de conflits détectés dans le modèle avec incertitudes.

Gestion des avions qui se suivent

Une exception doit cependant être considérée pour les avions qui se suivent : il s'agit d'un cas particulier où les règles de séparation classiques ne peuvent être appliquées, car le deuxième avion serait contraint de rouler à la vitesse minimale, ce qui n'est ni réaliste, ni approprié à l'optimisation.

Dans ce cas précis, il faut donc considérer que le deuxième pilote adaptera sa vitesse en fonction de celle du premier. Les zones d'incertitude ne sont pas modifiées mais elles deviennent conditionnelles (la vitesse du second est fonction de celle du premier). L'identification des cas où deux avions *se suivent* permet de prendre en compte ces incertitudes conditionnelles :

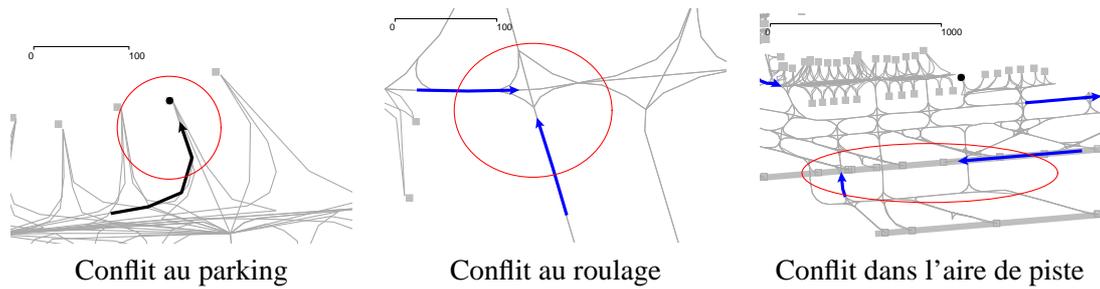


FIG. 4.5 – Conflits avec incertitudes

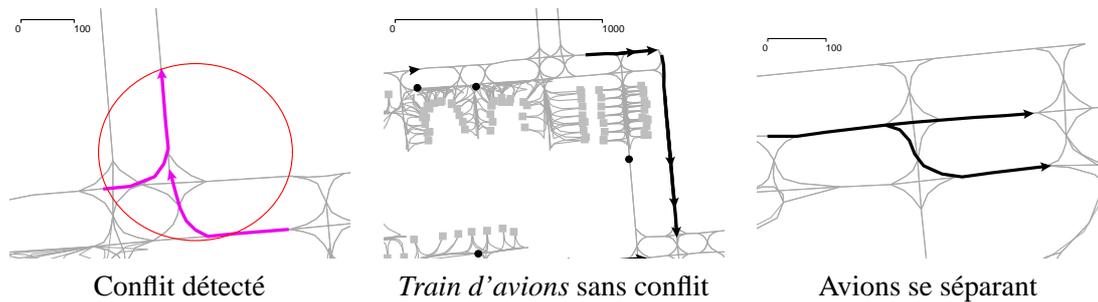


FIG. 4.6 – Conflits entre avions qui se suivent

1. Les positions principales des avions sont séparées et ne correspondent pas à des positions de décollage ou d'atterrissage.
2. Leurs zones d'incertitude ne sont pas totalement séparées.
3. Le deuxième avion est passé par la position la plus reculée de la zone d'incertitude du premier. Plus rigoureusement, cette position particulière du premier avion doit être *en conflit* avec la zone d'incertitude du second : cette condition assure rétrospectivement que les avions sont l'un derrière l'autre et que leur ordre ne peut être autrement (même si le premier avion a toujours adopté la vitesse minimale).

Sous ces conditions, les avions qui se suivent seront déclarés séparés. La figure 4.6 montre des exemples d'application de ces règles, notamment dans le cas d'un *train* d'avions (plusieurs avions se suivent mutuellement), comme cela peut souvent être observé sur le trafic réel.

Simplification de la détection de conflits

L'analyse des évolutions des zones d'incertitude des avions permet par ailleurs de simplifier considérablement la détection des conflits avec incertitudes, en altérant toutefois leur définition : ces zones sont continues et évoluent dans le sens de déplacement des avions, le long de leur chemin. La comparaison des positions de deux avions peut donc être réduite à la comparaison symétrique de la position la plus avancée de l'un des avions avec la zone d'incertitude de l'autre : dans ce cas, le conflit ne sera détecté qu'aux premiers pas de temps où les zones d'incertitude ne sont plus totalement séparées (le nombre de conflits détectés entre deux avions sera donc diminué) mais les paires d'avions en conflit sont identiques dans les deux modèles de détection.

La figure 4.7 donne un exemple d'application de ces règles simplifiées.

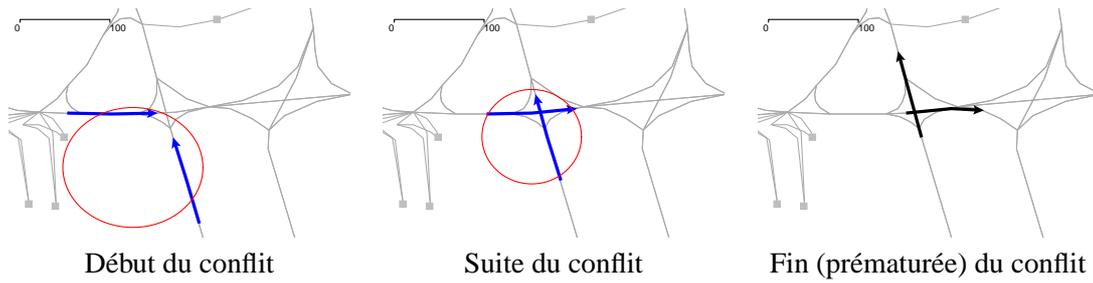


FIG. 4.7 – Simplification des conflits avec incertitudes

4.3.4 Optimisation et complexité de la détection de conflits

La détection des conflits entre une paire d'avions consiste à relever tous les pas de temps où ces deux avions ne sont pas séparés :

Sans incertitude, cette opération demande H_p comparaisons de positions ponctuelles (où H_p est exprimé en nombre de pas de temps) ;

Avec incertitudes, chaque zone d'incertitude des avions peut être constituée au maximum de $1 + E(\delta_v(\Delta + H_p))$ positions ponctuelles à chaque pas de temps (comme cela a été décrit dans la partie 4.2.4) et chacune de ces zones doit être comparée avec la position la plus avancée de l'autre avion (comme cela a été décrit dans la partie 4.3.3).

Si l'on considère que les paramètres Δ et δ sont fixes, la détection demandera donc $\mathcal{O}(\delta_v H_p)$ comparaisons de positions à chaque pas de temps, soient $\mathcal{O}(\delta_v H_p^2)$ comparaisons sur l'horizon de prédiction.

Pour N avions (soient $\frac{N(N-1)}{2}$ paires) la complexité de la détection de conflits est donc :

$$\mathcal{O}(N^2 H_p) \text{ sans incertitude} \quad \text{et} \quad \mathcal{O}(\delta_v N^2 H_p^2) \text{ avec incertitudes.}$$

Comparée à la prévision de trajectoire (qui demande NH_p opérations dans tous les cas), la détection de conflits est l'opération la plus pénalisante. Même si les simulations avec incertitudes sont effectuées avec un horizon de prédiction plus faible, la complexité de la détection de conflits augmente avec le carré de l'horizon et le carré du nombre d'avions. Ceci rend nécessaire son optimisation : en dehors des aires de piste, la détection peut être accélérée car seules les distances entre avions sont considérées alors que leurs vitesses sont bornées. La mesure de la distance $d(t)$ entre deux avions à un pas de temps donné, lorsqu'elle est supérieure à la norme de séparation D , permet de calculer le prochain pas de temps t' où les avions risquent d'être en conflit :

$$t' = t + 1 + \frac{d(t) - D}{2\delta V_{\max}}$$

L'algorithme II.1 utilise cette relation pour optimiser la détection des conflits au roulage d'une paire d'avions.

Detect_roulage(a_1, a_2):

soit $\gamma : \{ \}$ soit $t : \max(0, \min(H_p, \max(\text{début}(a_1), \text{début}(a_2))))$ soit $t_1 : \max(0, \min(H_p, \min(\text{fin}(a_1), \text{fin}(a_2))))$ tant que $t \leq t_1$ soit $d : \text{distance}(\text{position}(a_1, t), \text{position}(a_2, t))$ si $d < D$ alors $\gamma \leftarrow \gamma \cup \{t\}$ $t \leftarrow t + 1$ sinon $t \leftarrow t + 1 + E\left(\frac{d - D}{2\delta V_{\max}}\right)$ retourner γ	(Heures des conflits détectés) (Début du roulage) (Fin du roulage)
--	--

Algorithme II.1 – Détection des conflits au roulage

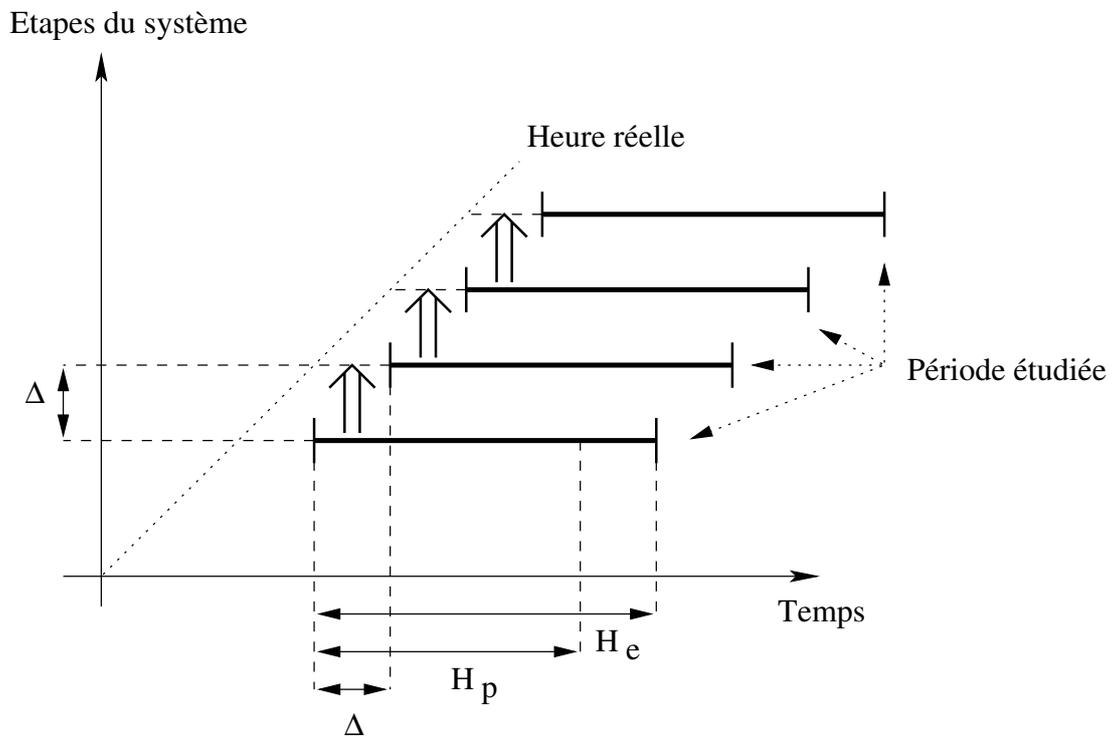


FIG. 4.8 – Simulation tabulée

4.4 Environnement de simulation

4.4.1 Itérations du système

En accord avec le modèle de prédiction, la simulation d'une journée de trafic consiste à résoudre successivement les situations, en déduisant chaque nouvelle situation de la dernière résolution (figure 4.8). Une simulation se déroule donc en plusieurs étapes :

1. Le traitement des plans de vol est tout d'abord effectué et le trafic sur l'aéroport est considéré comme nul (aucun avion n'est en mouvement). L'heure simulée t correspond à l'heure d'activation du premier plan de vol de la journée de trafic (qui peut être la veille avant minuit).
2. À chaque itération, la situation de trafic doit être étudiée et résolue sur l'horizon de prédiction. Dans un contexte opérationnel, ces opérations doivent être commencées à l'heure réelle $t - \Delta$ et terminées avant t . Dans le contexte de la simulation, ces opérations se déroulent de la façon suivante :
 - (a) **La nouvelle situation** est tout d'abord construite :
 - Les nouveaux mouvements, dont les plans de vol commencent avant la fin de l'horizon, doivent être activés. Lors de cette activation, un ensemble de chemins doit être envisagé pour chaque mouvement et un de ces chemins (le plus court) lui est initialement attribué. Les méthodes de recherche de ces chemins sont décrites au chapitre 6.
 - Inversement, les mouvements terminés avant l'heure courante t sont supprimés de l'ensemble des mouvements actifs.
 - Si la situation précédente contenait des mouvements non terminés, leur trajectoire prévue par la dernière résolution entre $t - \Delta$ et t est considérée comme définitive à cette étape : leur future position (à l'heure t) est prise en compte, ce qui restreint généralement l'ensemble des chemins qui peuvent encore leur être proposés.
 - (b) **La situation ainsi élaborée peut être résolue** : la formulation rigoureuse du problème et les méthodes envisagées pour le résoudre seront décrites dans les chapitres suivants. Les manœuvres résultantes devront être appliquées rigoureusement entre t et $t + \Delta$, alors que le système étudiera la situation suivante (à $t = t + \Delta$).
 - (c) **La simulation s'arrête** lorsque tous les plans de vol ont été traités et qu'il n'y a plus de mouvements actifs.

4.4.2 Passage à la situation suivante avec incertitudes

La prise en compte des incertitudes sur la vitesse laisse plusieurs possibilités pour effectuer le passage à l'étape suivante de la simulation, une fois que la situation courante est résolue : dans un environnement en temps réel, le système serait rafraîchi à chaque nouvelle étape (les positions réelles des avions à $t - \Delta$ seraient prises en compte pour étudier la situation sur $[t; t + H_p]$). Dans la simulation, ceci revient à rendre ponctuelle la position initiale de chaque avion avant de construire la prochaine situation. Deux principales stratégies peuvent être employées :

- La position initiale de chaque avion est réduite à sa position principale (celle qui correspond au roulage à la vitesse nominale).
- La position initiale de chaque avion est choisie aléatoirement dans sa zone d'incertitude initiale.

Ces deux stratégies devraient être équivalentes à partir du moment où les méthodes de résolutions n'anticipent pas ce choix.

La première stratégie sera employée pour comparer les méthodes de résolution, car elle assure que le résultat des simulations n'est pas influencé par le choix aléatoire de la position réelle des avions.

La deuxième solution permet de simuler concrètement les effets des variations de vitesses sur l'évolution des situations et elle sera donc implémentée, en traitant le cas des zones d'incertitude conditionnelles entre avions qui se suivent.

4.5 Conclusion

Un modèle de trajectoire, basé sur une représentation discrète du temps, est associé à un modèle général de prédiction de trafic pouvant prendre en compte les incertitudes sur les vitesses de roulage des avions.

Dans chaque situation, les règles de séparations permettent de détecter les conflits entre chaque paire d'avions et identifient ainsi les avions dont la trajectoire doit être modifiée, en fonction d'un modèle de manœuvre représentant les ordres de contrôle.

L'environnement de simulation permet de générer dynamiquement l'ensemble des situations de trafic d'une journée. Le réalisme de la simulation dépendra alors de la résolution de chacune de ces situations, ce qui est l'objet des prochains chapitres.

Chapitre 5

Formulation du problème

5.1 Introduction

L'environnement de simulation et la modélisation du trafic présentés dans le chapitre précédent permettent de générer des situations de trafic au sol. Cependant, pour que ces situations soient vraiment représentatives du trafic réel, elles doivent être *résolues* à chaque étape de la simulation :

1. Un ensemble de chemins alternatifs doit être affecté à chaque avion : cette étape définira en fait l'espace de recherche du problème posé par chaque situation de trafic et sera effectuée en fonction du graphe des voies de circulation de l'aéroport, lors de l'activation des plans de vol.
2. Les séparations entre avions doivent être assurées sur l'horizon de prédiction : il faut trouver les manœuvres à imposer à chaque avion pour que leurs trajectoires soient représentatives des espacements réglementaires. Ces notions apparaissent comme des contraintes du problème.
3. Les trajectoires résultantes doivent être optimales : l'arrêt de tous les avions, qui est une solution triviale pour assurer les séparations, n'est pas acceptable. Il faudra donc définir un ou plusieurs critères d'optimisation, pour évaluer les solutions et extraire la ou les meilleures d'entre elles, selon ce critère.

La résolution d'une situation de trafic peut ainsi être formulée comme un problème d'optimisation sous contraintes, comme cela sera précisé dans une première partie. Une analyse de la complexité du problème montrera, dans les parties suivantes, que ce dernier est fortement combinatoire et qu'il sera nécessaire de le décomposer en sous problèmes indépendants, à chaque fois que cela est possible.

5.2 Le problème à résoudre

5.2.1 Variables

Les variables du problème doivent fixer la trajectoire de chaque avion impliqué dans la situation. L'espace \mathcal{E} de l'ensemble de ces variables doit donc correspondre à toutes les combinaisons de toutes les trajectoires envisageables. Rapporté aux modèles de trajectoires et de manœuvres élaborés dans le chapitre précédent, ces variables doivent décrire, pour chaque avion :

Le chemin à suivre : ce chemin devra être choisi parmi l'ensemble des chemins possibles pour l'avion (la façon d'obtenir cet ensemble est décrite au chapitre 6).

Les attentes qui lui sont imposées, chacune d'elles se définissant par la position d'attente et l'heure jusqu'à laquelle l'avion doit attendre (pour plus de réalisme opérationnel, le nombre de positions d'attente peut être limité).

L'ordre de ralentissement en finale pour les futures arrivées, qui correspond à un décalage de l'heure effective d'atterrissage.

Plusieurs descriptions de ces manœuvres peuvent être envisagées pour définir précisément les variables du problème. L'ordre de ralentissement en finale peut être modélisé comme une position d'attente avant l'atterrissage, dont la durée est bornée. Pour N avions $(a_i)_{1 \leq i \leq N}$ ayant respectivement η_i chemins envisageables et limités à Ω positions d'attente, on peut alors définir :

$$\forall i \in \{1, \dots, N\}, \begin{cases} \nu_i \in \{1, \dots, \eta_i\} : \text{indice du chemin à suivre} \\ (p_{ik}, t_{ik})_{1 \leq k \leq \Omega} \in (\{0, \dots, H_p - 1\}^2)^\Omega : \text{suite des positions et heures de fin d'attente} \end{cases}$$

Dans ce cadre, p_{ik} est exprimé en nombre de pas de temps pour atteindre la position d'attente dans la trajectoire principale et t_{ik} est exprimé en nombre de pas de temps depuis l'heure courante. Pour chaque avion i , les suites $(p_{ik})_k$ et $(t_{ik})_k$ seront supposées strictement croissantes.

Cette formulation générale des variables sera utilisée dans toute cette partie : le terme *solution* désignera une instanciation particulière de ces variables, qu'elle soit ou non dans le domaine admissible défini par les contraintes du problème.

5.2.2 Contraintes

Contraintes élémentaires

Les contraintes élémentaires sont relatives aux domaines de définition des variables : pour correspondre à un ordre d'attente cohérent, la $n^{\text{ème}}$ attente de l'avion i , décrite par les variables (p_{in}, t_{in}) doit être fixée en fonction des attentes précédentes (pour $n > 1$), décrites par les variables $(p_{ik}, t_{ik})_{k < n}$. Pour cela, il faut considérer le retard cumulé maximal de l'avion i après sa $k^{\text{ème}}$ attente, qui est donnée directement par $t_{ik} - p_{ik}$, puisque les positions d'attente sont supposées classées. Les domaines de variation de ces variables sont donc liés entre eux de la façon suivante :

$$\forall i \in \{1, \dots, N\} \quad 0 \leq p_{i1} \leq t_{i1} \leq H_p$$

$$\forall k \in \{2, \dots, \Omega\}, \begin{cases} p_{i(k-1)} \leq p_{ik} \\ p_{ik} + t_{i(k-1)} - p_{i(k-1)} \leq t_{ik} \leq H_p \end{cases}$$

Des contraintes supplémentaires concernent la limite de « l'attente » avant atterrissage, pour les arrivées non encore posées au début de la période de prédiction. Si l'heure d'atterrissage initialement prévue est $t_{a_i} > 0$, l'ordre de ralentissement en finale est donné par $p_{i1} = t_{a_i}$ et la contrainte supplémentaire s'écrit :

$$0 \leq t_{i1} - p_{i1} < \lambda t_{a_i}$$

Où λ désigne le taux maximal de réduction de vitesse en finale (défini dans la partie 4.2.3).

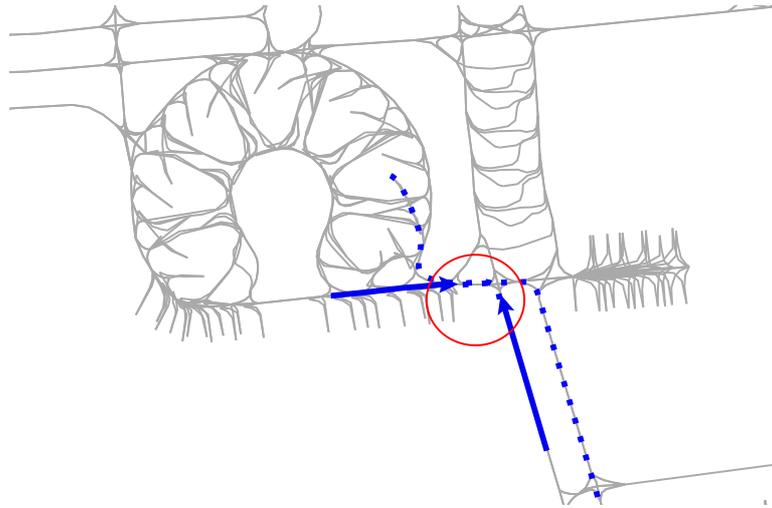


FIG. 5.1 – Face-à-face à la fin de l’horizon

Contraintes de séparation

L’absence de conflit entre avions peut être formulé par les $H_p \frac{N(N-1)}{2}$ contraintes de séparation suivantes :

$$\forall t \in \{0, \dots, H_p - 1\} \forall i \in \{2, \dots, N\} \forall j \in \{1, \dots, i - 1\} c_{ij}(t) = 0$$

Où $c_{ij}(t) \in \{0, 1\}$ indique si la paire d’avions (a_i, a_j) est en conflit au pas de temps t , en fonction des variables $\nu_i, \nu_j, (p_{ik}, t_{ik})_{1 \leq k \leq \Omega}$ et $(p_{jk}, t_{jk})_{1 \leq k \leq \Omega}$.

Ceci peut naturellement être ramené à une seule contrainte :

$$\sum_{t=0}^{H_p-1} \sum_{i=2}^N \sum_{j=1}^{i-1} c_{ij}(t) = 0$$

La vérification d’admissibilité d’une solution nécessite :

1. Le calcul des trajectoires des avions sur l’horizon de prédiction ;
2. La détection des conflits entre chaque paire d’avions.

Contraintes relatives à l’effet de l’horizon

Une analyse de la situation sur l’horizon de prédiction étendu H_e (défini dans la partie 4.2.2) permet d’éviter deux types de problèmes liés à l’effet de l’horizon :

1. **La mise en face-à-face de deux avions** à la fin de l’horizon peut être caractérisée par une règle simple : la position finale de chacun des avions (à $t = H_p$) est en conflit avec le futur chemin (sur l’horizon étendu) de l’autre avion, comme illustré figure 5.1.
2. **L’indisponibilité de la piste** à la fin de l’horizon correspond soit à l’occupation de la piste par un avion, soit à un décollage à forte turbulence de sillage. Cette situation, dangereuse lorsque des atterrissages sont prévus, peut être détectée simplement en étudiant les futures arrivées, sur l’horizon étendu.

Dans les deux cas, la taille de l'horizon étendu H_e a une importance : plus elle est grande, plus la règle est pénalisante. Pour être adaptée au problème, sa valeur doit correspondre :

- au temps de parcours de la plus grande ligne droite sans intersection pour la première règle ;
- et au temps maximal de séparation sur la piste pour la seconde.

En pratique, H_e sera de l'ordre de quelques minutes dans les deux cas.

Ces règles pourront être considérées comme des contraintes, selon la méthode de résolution employée. Dans l'affirmative, le non respect d'une de ces règles sera modélisé (et caractérisé) par un conflit à la fin de l'horizon, entre les paires d'avions concernés : dans le cas de la disponibilité de la piste, la prochaine arrivée prévue est considérée en conflit avec tout avion occupant la piste à la fin de l'horizon, (même si cette arrivée n'est pas réellement active dans la situation étudiée).

5.2.3 Critère d'optimisation

Le critère d'optimisation doit évaluer globalement chaque solution admissible. La mise au point de ce critère fait nécessairement intervenir des notions subjectives :

- Des critères rigoureusement définis, comme la longueur ou le nombre de virages du chemin, peuvent être jugés plus représentatifs que les temps de parcours, qui dépendent de la vitesse de roulage et donc des habitudes des pilotes.
- Un détour (rallongement souvent irréversible du temps de roulage) doit être mis en concurrence avec une attente provisoire le long d'un chemin plus court.
- Les attentes au parking peuvent être privilégiées, pour éviter la congestion des voies de circulation et rendre l'embarquement des passagers moins contraignant.
- Certains avions peuvent être jugés plus prioritaires, parce qu'ils ont déjà un retard conséquent, ou parce qu'ils sont contraints de respecter un créneau de décollage.

Loin d'être exhaustive, cette liste montre que le critère d'optimisation doit absolument rester évolutif pour être affiné en fonction du type de trafic de chaque aéroport. Les méthodes d'optimisation qui seront développées doivent donc faire le moins d'hypothèses possible sur ce critère, qui sera vu initialement comme une fonction à minimiser, ne possédant aucune propriété particulière.

Cette liste montre également que beaucoup de notions plus ou moins mesurables peuvent intervenir et qu'il sera intéressant de trouver plusieurs solutions admissibles : le choix final d'une de ces solutions peut revenir aux opérationnels, en fonction de leur expertise des situations. Dans ce cadre, la définition d'un critère multi-objectifs (une fonction multi-dimensionnelle à optimiser) peut sembler attrayante.

Critère mono-objectif

Un premier critère mono-dimensionnel relativement simple peut être défini, en fonction de la somme des temps de roulage supplémentaires $\mathcal{L}(\nu)$ due aux choix des chemins suivis par les avions et des attentes $\mathcal{D}(p, t)$ qui leurs sont imposées :

$$f(\nu, p, t) = \alpha \mathcal{L}(\nu) + \mathcal{D}(p, t)$$

Où $\alpha > 1$ pondère la pénalité des détours par rapport aux retards. Si $l_i(\nu_i)$ désigne le temps de roulage supplémentaire impliqué par le chemin ν_i de l'avion i :

$$\mathcal{D}(p, t) = \sum_{i=1}^N t_{i\Omega} - p_{i\Omega} \quad \text{et} \quad \mathcal{L}(\nu) = \sum_{i=1}^N l_i(\nu_i)$$

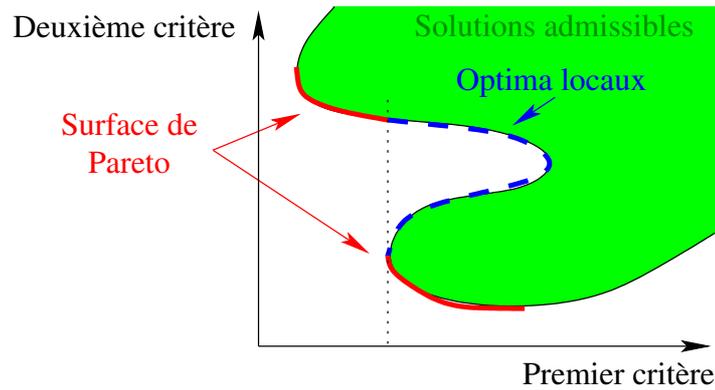


FIG. 5.2 – Surface de Pareto en recherche multi-objectifs

Dans ce cas, le retard imposé à l'avion sera jugé α fois plus pénalisant lorsqu'il est dû à un détour que lorsqu'il est dû à une attente. À retard égal, il sera donc toujours préférable d'attendre que d'emprunter un détour ($\alpha > 1$) : dans toute la suite, ce critère sera celui utilisé par défaut pour l'optimisation du trafic au sol, avec $\alpha = \frac{3}{2}$.

Critère multi-objectifs

Dans la recherche multi-objectifs, la fonction f est multi-dimensionnelle et chaque dimension représente un critère mono-objectif particulier f_i que l'on cherche par exemple à minimiser :

$$f = (f_1, \dots, f_i, \dots, f_n)$$

Les solutions sont comparées par la *relation de dominance*, définie par rapport à f sur l'espace de recherche :

$$\forall (s_1, s_2) \in \mathcal{E}^2, s_1 \text{ domine } s_2 \iff \begin{cases} s_1 \text{ est admissible} \\ s_2 \text{ non admissible ou } \forall i \in \{1, \dots, n\}, f_i(s_1) < f_i(s_2) \end{cases}$$

Cette relation de dominance n'est pas une relation d'ordre total, mais elle permet de caractériser l'ensemble des solutions qui ne sont *dominées* par aucune autre : cet ensemble de solutions admissibles est appelé la *surface de Pareto* du problème (figure 5.2). Chacune de ces solutions peut être considérée comme un optimum puisqu'elle correspond à un point pour lequel aucun critère ne peut être amélioré sans en pénaliser d'autres.

La recherche multi-objectifs est par définition plus complexe que la recherche mono-objectif (les optima locaux sont souvent plus nombreux) mais peut parfois apporter de meilleures solutions lorsque le problème est résolu par algorithmes génétiques, car la diversification de la population peut s'avérer meilleure.

Dans le cas de l'optimisation du trafic au sol, les critères envisagés par les opérationnels sont généralement classés par ordre de priorité et peuvent être pondérés aisément au sein d'un unique critère mono-dimensionnel : l'utilisation de la recherche multi-objectifs ne semble donc pas indispensable et ne sera pas développée par la suite, bien que cette voie mérite amplement d'être approfondie au cours d'études ultérieures.

5.3 Analyse du problème

5.3.1 Complexité

Tel qu'il est formulé dans la partie précédente et en supposant toujours qu'un nombre maximum de η chemins est envisagé par avion, le problème à N avions est décrit par :

- N variables entières $(\nu_i)_{1 \leq i \leq N}$ dans $\{1, \dots, \eta\}$;
- $2\Omega N$ variables entières $(p_{ij}, t_{ij})_{1 \leq i \leq N, 1 \leq j < \Omega}$ dans $\{0, \dots, H_p\}$.

La taille de l'espace de recherche peut donc être exprimée par :

$$|\mathcal{E}| = \mathcal{O}(\eta^N H_p^{2\Omega N})$$

Le problème est donc fortement combinatoire par rapport au nombre d'avions impliqués et à la taille de l'horizon de prédiction :

$$\text{Pour } \begin{cases} N & = 5 \text{ avions} \\ H_p & = 5 \text{ minutes de prédiction (soient 60 pas de 5 secondes)} \\ \eta & = 5 \text{ chemins par avion} \\ \Omega & = 2 \text{ positions d'attentes par avion} \end{cases}$$

On obtient : $|\mathcal{E}| > 10^{39}$ solutions.

5.3.2 Composantes connexes

Une analyse plus générale du problème de résolution de conflits montre que le nombre de composantes connexes du problème est lui-même combinatoire, car les contraintes de séparation divisent fortement l'espace des solutions admissibles :

- L'étude du problème à N avions pour le trafic en route ([Durand 96a]) montre que le nombre de composantes connexes est exponentiel par rapport au carré du nombre d'avions.
- Ce résultat reste valable pour le trafic au sol tel qu'il est modélisé :
 - Pour deux avions en conflit sur un de leur chemin particulier, deux espaces disjoints de solutions admissibles peuvent être distingués, selon que le premier avion s'arrête pour laisser passer l'autre ou non.
 - Si l'on considère tous les chemins envisagés pour les deux avions, on peut obtenir jusqu'à η^2 paires de chemins conflictuels, soient $2\eta^2$ composantes disjointes.
 - Pour N avions, on obtient $\frac{N(N-1)}{2}$ conflits potentiels, soient :

$$(2\eta^2)^{\frac{N(N-1)}{2}} = (\sqrt{2}\eta)^{N(N-1)} \text{ composantes connexes}$$

Cette expression est naturellement un majorant du nombre effectif de composantes disjointes (car beaucoup de composantes peuvent être vides ou confondues), mais il n'existe pas de méthodes connues pour le savoir à l'avance : même si à l'intérieur de chaque composante, des méthodes de recherche peuvent être développées pour trouver très rapidement l'optimum local, le nombre de ces composantes est lui-même rapidement prohibitif :

$$\text{Pour } \begin{cases} N & = 5 \text{ avions} \\ \eta & = 5 \text{ chemins par avion} \end{cases}$$

On peut obtenir plus de 10^{16} composantes connexes.

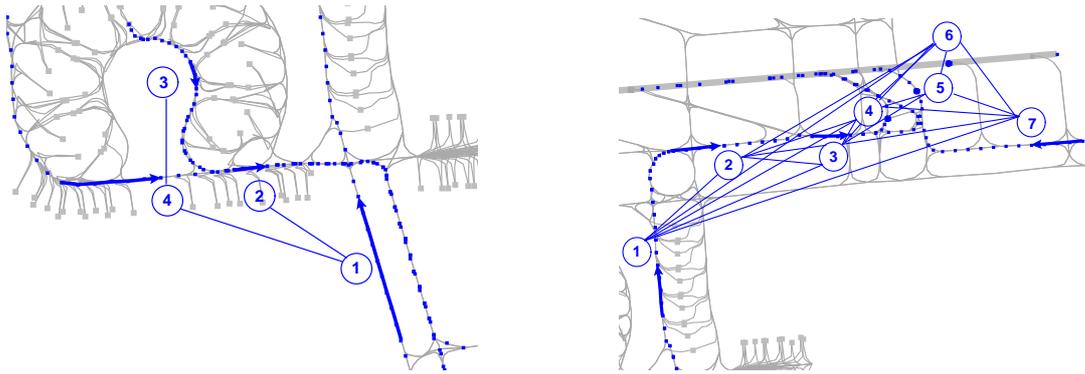


FIG. 5.3 – Clusters d'avions en conflit

5.3.3 Domaine d'admissibilité et optimum

Comme dans tout problème de minimisation sous contraintes, les solutions optimales se trouvent souvent sur les frontières du domaine admissible : dans le cas du critère global défini au 5.2.3 en fonction des délais et des détours, une diminution élémentaire du délai d'un seul avion a_i dans une solution optimale représente nécessairement une violation de contrainte, alors qu'une seule variable booléenne t_{ij} aura été modifiée.

5.4 Décomposition en problèmes indépendants

5.4.1 Principes

La complexité élevée du problème posé par chaque situation de trafic incite à le diviser autant que possible en sous problèmes indépendants. Dans cette partie, une méthode simple pour diviser le problème, très utilisée pour la résolution des conflits en route ([Durand 96a]) est reprise et appliquée au trafic au roulage : l'ensemble des avions est divisé en groupes d'avions en conflits, appelés *clusters*. Chaque cluster peut initialement être résolu séparément des autres.

5.4.2 Clusters d'avions en conflit

La relation *est en conflit avec* définie sur l'ensemble \mathcal{A} des N avions est trivialement symétrique. Une fermeture transitive peut donc être appliquée dans l'ensemble \mathcal{P} des paires d'avions en conflit sur l'horizon de prédiction. Le résultat est une partition $(\mathcal{A}_i)_{1 \leq i \leq m}$ de l'ensemble des avions qui ont au moins un conflit. Les trajectoires initiales de deux avions appartenant à deux sous-ensembles distincts ne peuvent donc pas être en conflit :

$$\forall i \in \{1, \dots, m\}, \forall a \in \mathcal{A}_i, \begin{cases} \exists a' \in \mathcal{A}_i, (a, a') \in \mathcal{P} \\ \forall b \in \mathcal{A}/\mathcal{A}_i, (a, b) \notin \mathcal{P} \end{cases}$$

Les éléments (\mathcal{A}_i) de cette partition sont les *clusters* d'avions en conflit. Chaque cluster forme un sous problème a priori indépendant des autres : si une solution optimale est trouvée pour chacun d'eux, sans que de nouveaux conflits n'apparaissent entre avions de différents clusters, la juxtaposition de ces solutions forme une solution optimale au problème des N avions.

La figure 5.3 montre deux exemples de clusters :

- Dans le premier, trois conflits réunissent quatre avions (exemple classique d'avions réunis pendant la phase de roulage proprement dite, lors du passage d'une intersection).
- Dans le deuxième, sept avions sont presque tous en conflits deux à deux à cause des séparations imposées sur la piste (le cluster contient 19 conflits).

Ces deux exemples font apparaître l'éventuelle limite de la séparation du problème en clusters indépendants, dans le cas spécifique du trafic au sol : la piste peut en effet être vue comme une ressource partagée par tous les avions et les séparations en temps imposées pour cette ressource sont suffisamment pénalisantes pour rendre conflictuelles toutes les paires d'avions, au sein d'un même cluster de grande taille : le nombre maximal de composantes connexes calculé dans la partie précédente (complexité du *pire cas*) risque d'être fréquemment atteint.

5.4.3 Résolution itérative

La résolution du problème global à N avions se définit en fonction de la résolution séparée de chaque cluster, de manière itérative et éventuellement parallélisée :

1. Initialement, chaque avion constitue un cluster particulier (d'un seul élément) et se voit attribué sa trajectoire initiale définie dans la partie 4.2.3.
2. À chaque itération :
 - (a) Les paires d'avions en conflit sont détectées, à partir des trajectoires courantes des avions.
 - (b) Les nouveaux clusters sont formés, en appliquant la fermeture transitive définie par les paires d'avions en conflits sur les clusters précédents (chaque conflit entre avions de différents clusters unifie leur cluster).
 - (c) Chaque cluster de plus d'un avion est résolu séparément (cette opération peut être parallélisée par clusters).
3. La résolution s'arrête lorsque l'ensemble des clusters n'évolue plus :
 - S'il n'y a plus de conflit, les trajectoires courantes des avions forment la solution trouvée.
 - Sinon, aucune solution ne peut être trouvée pour le problème global.

Cette forme de résolution permet de simplifier et d'accélérer significativement la résolution de conflits (la plupart des problèmes impliquant un grand nombre d'avions ne pourraient d'ailleurs pas être résolus sans cette approche). Un point important doit cependant être souligné : le caractère optimal de la solution finale peut être remis en cause si les trajectoires initiales des avions ne sont pas en accord avec le critère global à optimiser. En effet, les trajectoires des avions sans conflit ne sont jamais remises en cause dans cette méthode de résolution.

5.5 Conclusion

Le problème global posé par chaque situation de trafic s'exprime sous la forme d'un problème de minimisation sous contraintes qui peut être décomposé en plusieurs sous-problèmes initialement indépendants, en fonction des clusters d'avions en conflit. Chaque sous-problème reste néanmoins fortement combinatoire, dès lors que plusieurs chemins et plusieurs attentes sont envisagés pour chaque avion.

La résolution du problème nécessite par conséquent soit sa simplification, pour être traitée par des méthodes déterministes, soit la mise-en-œuvre de méthodes non déterministes, plus adaptées aux problèmes de grande taille.

Troisième partie

Méthodes de résolution

Chapitre 6

Affectation de chemins

6.1 Introduction

Dans ce chapitre, le graphe représentant les voies de circulation de l'aéroport est utilisé pour affecter un ensemble de chemins à chaque mouvement sur l'aéroport. Il faut souligner que la recherche de cheminements admissibles est en réalité indissociable de l'optimisation du trafic au sol, puisqu'elle en définit justement l'espace de recherche : la qualité des chemins proposés aux mouvements influencera directement le réalisme et la perspicacité des solutions trouvées. Le principal dilemme pour la recherche de chemins découle de ces considérations et concerne le nombre de chemins à envisager :

- Ce nombre doit être suffisamment grand pour que toutes les déviations susceptibles d'éviter la congestion de certaines zones de l'aéroport soient étudiées. Ceci sera illustré tout au long de cette partie, où l'on pourra constater que certains cheminements, régulièrement imposés par le contrôle au sol, représentent en fait un détour conséquent.
- Cependant, plus ce nombre sera grand, plus l'optimisation des situations de trafic sera difficile : chaque chemin de chaque mouvement définit une composante connexe particulière de l'espace de recherche et la taille de ce dernier croît exponentiellement avec le nombre de chemins proposés.

L'objectif est donc de trouver une méthode d'énumération des chemins qui ne soit pas seulement efficace (en terme de temps de calcul), mais qui soit également suffisamment adaptée au cheminement des avions sur un aéroport. Sur ce sujet, l'absence de base de données répertoriant plus directement les chemins opérationnellement valides (même sur des grands aéroports comme Roissy ou Orly) est une justification supplémentaire à l'élaboration d'une telle méthode (les enregistrements des traces radar des avions pourraient servir à former une telle base de données, mais risquerait d'aboutir à des échantillons de chemins trop restrictifs, qui seraient de plus inutilisables pour évaluer l'impact de nouvelles procédures ou de nouvelles infrastructures).

6.2 Analyse préliminaire

6.2.1 Références

Le problème de la recherche des meilleurs chemins dans un graphe orienté s'applique en réalité à de nombreux domaines aussi vastes que celui des transports, des télécommunications ou encore de la robotique. Ce problème est donc fortement documenté et une bibliographie très fournie, réalisée par D. Eppstein, peut être trouvée en ligne à l'adresse suivante :

<http://www.ics.uci.edu/~eppstein/bibs/kpath.bib>

Les méthodes proposées s'inspirent le plus souvent des principes décrits dans [Yen 71], en leur ajoutant divers types de raffinements, tels que l'utilisation de structures de données optimisées pour la gestion de files à priorités (*priority queues*) [Eppstein 98].

Ce chapitre est consacré à la description des quelques méthodes qui semblent les plus adaptées à la description et à la taille du graphe des voies de circulation des aéroports.

6.2.2 Notations principales

On désigne toujours le graphe orienté par $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, avec $\mathcal{A} \subset \mathcal{N}^2$ et par c la fonction de coût associant aux arcs $(n, p) \in \mathcal{A}$ un réel positif ou nul.

Pour tout couple de nœuds (s, t) de \mathcal{N}^2 , \mathcal{C}_{st} désigne l'ensemble des chemins du graphe allant de s à t . La relation (\cdot) désignera la composition entre nœuds et chemins, qui permet d'ajouter un nœud à un chemin (aussi bien en aval qu'en amont) : un chemin γ de n_1 à n_l sera donc écrit $(n_1 \cdot n_2 \cdot \dots \cdot n_l)$ ou encore $(n_i)_{1 \leq i \leq l}$. Le coût d'un tel chemin sera noté \bar{c}_γ et défini par :

$$\bar{c}_\gamma = \sum_{i=1}^{l-1} c_{n_i n_{i+1}}$$

Si le graphe admet des cycles, \mathcal{C}_{st} peut ne pas être fini. Mais lorsque \mathcal{C}_{st} est non vide, la fonction de coût c étant positive (ou nulle) et l'ensemble \mathcal{N} des nœuds du graphe étant fini, l'ensemble des coûts des chemins de \mathcal{C}_{st} admet un minimum positif atteint, noté \bar{c}_{st} :

- $\forall n \in \mathcal{N}$, $\bar{c}_{nn} = 0$ (le chemin trivial (n) ayant un coût nul)
- $\forall (s, t) \in \mathcal{N}^2$ tel que $s \neq t$, **le coût minimal** \bar{c}_{st} sera défini par :

$$\text{Si } \mathcal{C}_{st} \neq \{\}, \text{ alors } \bar{c}_{st} = \text{Min}_{\gamma \in \mathcal{C}_{st}} \{\bar{c}_\gamma\}; \quad \text{sinon, } \bar{c}_{st} = \infty;$$

L'ensemble des plus courts chemins de s à t est alors défini par : $\bar{c}^{(-1)}(\bar{c}_{st}) = \{\gamma \in \mathcal{C}_{st} \mid \bar{c}_\gamma = \bar{c}_{st}\}$

6.2.3 Coûts minimaux depuis un nœud

Toutes les méthodes de recherche de chemins entre s et t décrites dans les parties suivantes nécessitent la connaissance de l'ensemble $(\bar{c}_{sn})_{n \in \mathcal{N}}$ des coûts minimaux depuis le nœud s . Parmi les algorithmes les plus connus permettant d'obtenir ces coûts minimaux (détaillés en annexe D), il faut citer :

- **L'algorithme de Dijkstra** [Ahuja 93] (annexe D.3), qui fournit tous les meilleurs chemins (et les coûts associés) allant d'un sommet donné vers tous les autres sommets du graphe, sous la condition que le graphe ne comporte pas de coût négatif. Selon son implémentation, le résultat peut être obtenu en $\mathcal{O}(mn)$ ou $\mathcal{O}((m+n) \log n)$ opérations ($\mathcal{O}(m+n)$ pour certains graphes particuliers), où m est le nombre d'arcs et n le nombre de sommets du graphe.
- **L'algorithme de Floyd** (annexe D.2), qui permet d'obtenir tous les meilleurs chemins et leurs coûts respectifs entre toutes les paires de nœuds du graphe. Sa complexité élevée ($\mathcal{O}(n^3)$ au minimum) le rend parfois plus coûteux (en temps de calcul) que l'itération de l'algorithme de Dijkstra sur tous les nœuds.

6.3 Méthode de la déviation maximale

6.3.1 Description de la méthode

Une première méthode pour énumérer des chemins reliant deux nœuds s et t est celle implémentée dans [Pescic 00] : elle consiste à rechercher *tous* les chemins entre s et t , dont le coût est inférieur au coût minimal \bar{c}_{st} augmenté d'un sur-coût Δ fixé.

Une fois que l'ensemble $\{\bar{c}_{sn}\}_{n \in \mathcal{N}}$ des coûts minimaux depuis s est connu, un algorithme de parcours de graphe *en profondeur d'abord* permet d'énumérer ces chemins : l'algorithme explore les nœuds du graphe de pères en pères, en partant du nœud terminal t et dans n'importe quel ordre. Lorsque qu'un nœud n est exploré, le chemin γ^t et le coût c_{nt} pour relier n à t est connu, ainsi que le coût minimal \bar{c}_{sn} de s à n . L'exploration est donc abandonnée lorsque ce coût minimal devient supérieur à la borne $\bar{c}_{st} + \Delta - c_{nt}$.

Cet algorithme a l'avantage de pouvoir être adapté facilement pour ne développer que les chemins qui satisfont la contrainte de validité donnée au 3.3.4 : il suffit pour cela d'interrompre l'exploration d'une branche lorsque le nœud courant n n'est pas compatible avec le chemin courant γ^t . L'algorithme III.1 en présente la version récursive.

```

Dev_Max( $\mathcal{G}, (\bar{c}), s, t, \Delta$ ) :
  soit  $\mathcal{C} : ()$  et  $c_{\max} : \bar{c}_{st} + \Delta$  (Liste des chemins trouvés et coût maximal)
  Explorer( $c_{nt}, n, \gamma^t$ ) :
    si  $n \neq s$  et  $\bar{c}_{sn} < c_{\max} - c_{nt}$  et  $n \cap \gamma^t = \{\}$  alors
      pour chaque  $m \in \mathcal{P}ères(n)$  Explorer( $c_{mn} + c_{nt}, m, (n \cdot \gamma^t)$ )
    sinon si  $n = s$  et  $n \cap \gamma^t = \{\}$  et  $c_{nt} < c_{\max}$  alors Ajouter ( $s \cdot \gamma^t$ ) à  $\mathcal{C}$ 
  Explorer( $0, t, ()$ )
  retourner  $\mathcal{C}$ 

```

Algorithme III.1 – Dev_Max : recherche des déviations bornées

6.3.2 Complexité

Sans la contrainte de validité définie sur les chemins, l'algorithme ainsi formulé visiterait chaque nœud du graphe exactement autant de fois qu'il y a de chemins trouvés passant par ce nœud : ceci découle de la connaissance exacte des coûts minimaux (\bar{c}). Dans ce cas, le nombre d'itérations de Explorer pourrait donc être majoré par kl , où k désignerait le nombre de chemins trouvés et l le nombre maximal de nœuds par chemin.

L'ajout de la contrainte de validité ne peut cependant que diminuer le nombre d'itérations de Explorer, qui pourra donc toujours être majoré par kl , à condition que k désigne le nombre de chemins **de coût admissible** (et non le nombre de chemins réellement trouvés).

Au cours de chaque itération de Explorer, l'opération la plus pénalisante consiste à vérifier que le nœud courant n'interfère pas avec le chemin emprunté depuis t : ceci demande $\mathcal{O}(l)$ calculs d'intersections géométriques.

Dev_Max s'effectue en $\mathcal{O}(kl^2)$ opérations.

Où : $\begin{cases} k \text{ désigne le nombre de chemins de coûts admissibles,} \\ l \text{ le nombre maximal de nœuds par chemin.} \end{cases}$

Il faut cependant remarquer que le nombre k de chemins de coût admissible dépend du paramètre Δ et ne peut pas être connu à l'avance. Il n'est par ailleurs pas souhaitable de limiter l'algorithme à un nombre fixé de chemins, puisque leur énumération n'est pas ordonnée (on n'obtiendrait donc plus les *meilleurs* chemins).

6.3.3 Résultats

La figure 6.1 illustre quelques exemples d'application de l'algorithme aux voies de circulation de Roissy et d'Orly, pour différentes valeurs de Δ , avec ou sans sens préférentiels. Il en ressort tout d'abord une grande variation du nombre de chemins (et donc du temps nécessaire à leur calcul) en fonction des paramètres et de l'aéroport.

- A Roissy, les deux premiers exemples (1.1 et 1.2) montrent qu'une déviation maximale de deux minutes (qui semble minimaliste pour un tel aéroport) rapporte déjà un nombre de chemins beaucoup trop élevé (1352 chemins sans sens préférentiels). Les deux exemples suivants (2.1 et 2.2) mettent en évidence le caractère extrêmement variable et très rapidement absurde du nombre de chemins obtenus.
- A Orly, les voies de circulation moins complexes donnent des résultats plus exploitables (car plus constants). Les exemples sélectionnés (3.1 et 3.2) font cependant apparaître une particularité : le non respect des sens préférentiels peut faire diminuer le nombre de chemins trouvés (à Δ fixé). En effet, les sens préférentiels dénaturent la notion de *plus court* chemin, qui peut devenir beaucoup plus court sans sens préférentiels. Il peut naturellement en résulter un nombre moins important de chemins situés dans la plage de coûts correspondante.

Ces exemples montrent que les chemins trouvés sont difficilement satisfaisants, car beaucoup trop nombreux ou trop peu diversifiés : les ramifications des voies de circulation rendent le nombre de chemins combinatoire et il est nécessaire de limiter le nombre de combinaisons, en utilisant d'autres méthodes d'énumération.

6.4 Recherche des k meilleurs chemins

6.4.1 Formulation du problème

Pour pouvoir limiter le nombre de chemins à chercher, il est nécessaire de les énumérer dans l'ordre de leur coût (du moins au plus coûteux). Le problème devient alors la recherche de k meilleurs chemins reliant s à t .

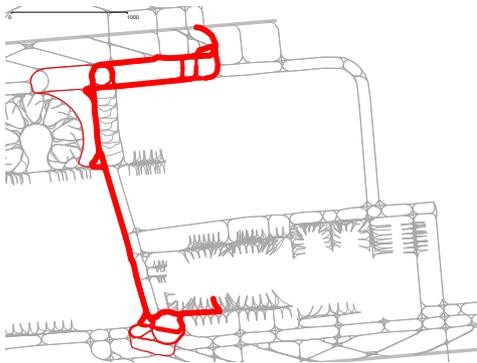
Pour chaque nœud n du graphe, \mathcal{C}_{sn} désigne toujours l'ensemble de tous les chemins reliant s à n et \bar{c}_γ le coût d'un chemin γ . Dans le cas particulier du nœud origine s , le chemin élémentaire (s) est toujours considéré comme valide et de coût nul ($\bar{c}_{ss} = 0$).

Un sous ensemble \mathcal{C}_{sn}^k ($k \geq 0$) de \mathcal{C}_{sn} est appelé ensemble de k meilleurs chemins entre s et n si il est de cardinal k et que tout chemin plus court que l'un de ses chemins lui appartient :

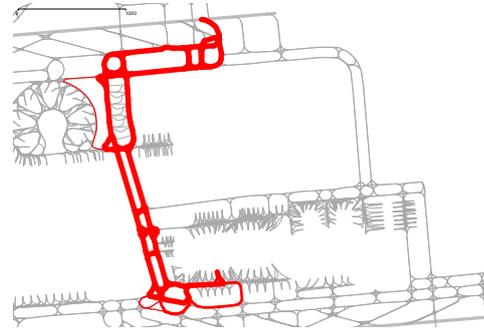
$$\forall \gamma \in \mathcal{C}_{sn}^k, \forall \gamma' \in \mathcal{C}_{sn}, \bar{c}_{\gamma'} < \bar{c}_\gamma \implies \gamma' \in \mathcal{C}_{sn}^k$$

Il peut naturellement exister plusieurs ensembles de k meilleurs chemins de s à n de même cardinal, puisque différents chemins peuvent avoir le même coût. Par contre, le coût maximal des éléments de ces ensembles est unique et sera noté $c_k(n)$:

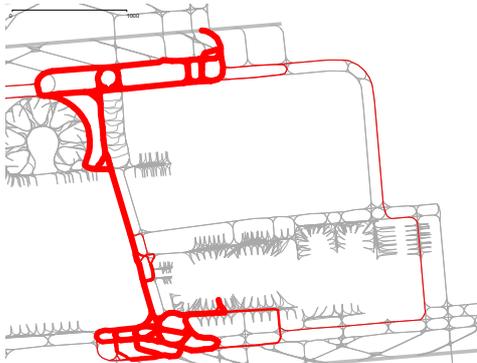
$$\forall \mathcal{C}_{sn}^k \text{ ensemble de } k \text{ meilleurs chemins de } s \text{ à } n, \text{ avec } k \geq 1, \max_{\gamma \in \mathcal{C}_{sn}^k} \bar{c}_\gamma = c_k(n)$$



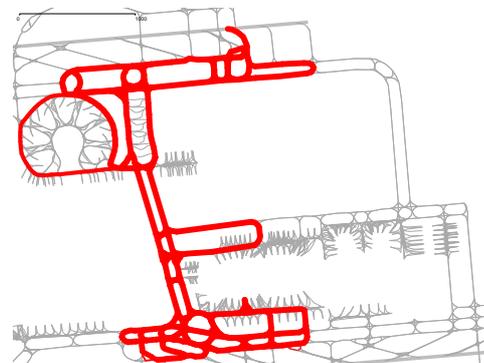
1.1 Roissy : $\Delta = 120s$
Avec sens préférentiels
92 chemins (trouvés en 80 ms)



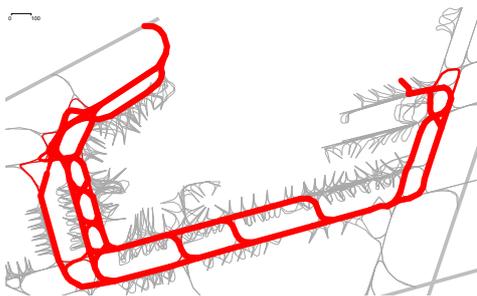
1.2 Roissy : $\Delta = 120s$
Sans sens préférentiels
1352 chemins (trouvés en 1.2s)



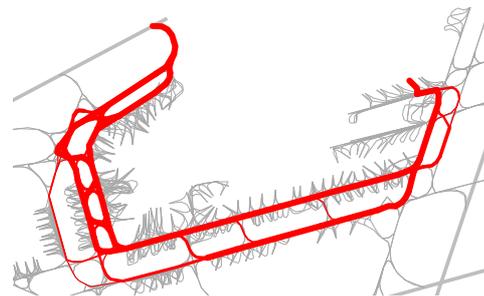
2.1 Roissy : $\Delta = 200s$
Avec sens préférentiels
948 chemins (trouvés en 825ms)



2.2 Roissy : $\Delta = 200s$
Sans sens préférentiels
15850 chemins (trouvés en 143s)



3.1 Orly : $\Delta = 60s$
Avec sens préférentiels
264 chemins (trouvés en 145ms)



3.2 Orly : $\Delta = 60s$
Sans sens préférentiels
113 chemins (trouvés en 68ms)

FIG. 6.1 – Applications de Dev_Max

$$\text{Il en découle : } c_{k+1}(n) = \text{Min}_{\gamma \in \mathcal{C}_{sn}/\mathcal{C}_{sn}^k} \bar{c}_\gamma$$

Le problème est donc de trouver un ensemble \mathcal{C}_{st}^k de k meilleurs chemins entre s et t , si un tel ensemble existe. Dans la négative, le problème revient à chercher \mathcal{C}_{st} .

6.4.2 Relation de récurrence

Une relation de récurrence, décrite pour la première fois par Dreyfus [Dreyfus 69] permet de calculer la suite $(c_k(n))_{k \geq 1}$ des coûts des meilleurs chemins de s à n , à partir des coûts des meilleurs chemins de s à chacun des pères de n dans le graphe :

$$\text{Soient } \left\{ \begin{array}{l} \mathcal{C}_{sn}^k \text{ un ensemble de } k \text{ meilleurs chemins de } s \text{ à } n, \text{ avec } k \geq 1 \\ \forall m \in \mathcal{P}ères(n), \Gamma_m^k = \{\gamma_m \in \mathcal{C}_{sm} | (\gamma_m \cdot n) \in \mathcal{C}_{sn}^k\} \quad \text{et} \quad j_m^k = |\Gamma_m^k| \end{array} \right.$$

$$\text{La relation de récurrence s'écrit alors : } c_{k+1}(n) = \text{Min}_{m \in \mathcal{P}ères(n)} c_{j_m^k+1}(m) + c_{mn}$$

Cette relation sert de base à l'élaboration des méthodes itératives permettant d'obtenir k meilleurs chemins entre deux nœuds. Elle se démontre comme suit :

Lemme 1 $\forall m \in \mathcal{P}ères(m), \Gamma_m^k$ est un ensemble de j_m^k meilleurs chemins de s à m .

En effet, $\forall \gamma_m \in \Gamma_m^k, \forall \gamma'_m \in \mathcal{C}_{sm}/\Gamma_m^k$, on a par construction de Γ_m^k :

$$(\gamma_m \cdot n) \in \mathcal{C}_{sn}^k \quad \text{et} \quad (\gamma'_m \cdot n) \in \mathcal{C}_{sn}/\mathcal{C}_{sn}^k$$

Et comme \mathcal{C}_{sn}^k est un ensemble de k meilleurs chemins : $\bar{c}_{(\gamma_m \cdot n)} \leq \bar{c}_{(\gamma'_m \cdot n)}$

$$\text{soit : } \bar{c}_{\gamma_m} + c_{mn} \leq \bar{c}_{\gamma'_m} + c_{mn} \quad \text{et donc : } \bar{c}_{\gamma_m} \leq \bar{c}_{\gamma'_m}$$

Lemme 2 $\forall m \in \mathcal{P}ères(m), \mathcal{C}_{sn}/\mathcal{C}_{sn}^k = \bigcup_{m \in \mathcal{P}ères(n)} \bigcup_{\gamma_m \in \mathcal{C}_{sm}/\Gamma_m^k} \{(\gamma_m \cdot n)\}$

Ceci découle naturellement de la structure même du graphe, qui impose :

$$\forall n \neq s, \mathcal{C}_{sn} = \bigcup_{m \in \mathcal{P}ères(n)} \bigcup_{\gamma_m \in \mathcal{C}_{sm}} \{(\gamma_m \cdot n)\}$$

Pour le nœud particulier s , il faut ajouter à cette réunion le chemin trivial (s) . Dans tous les cas, la définition des Γ_m^k donne alors directement le résultat.

On peut alors écrire :

$$\begin{aligned} c_{k+1}(n) &= \text{Min}_{\gamma_n \in \mathcal{C}_{sn}/\mathcal{C}_{sn}^k} \bar{c}_{\gamma_n} \\ &= \text{Min}_{m \in \mathcal{P}ères(n)} \text{Min}_{\gamma_m \in \mathcal{C}_{sm}/\Gamma_m^k} \bar{c}_{(\gamma_m \cdot n)} \quad (\text{d'après le lemme 2}) \\ &= \text{Min}_{m \in \mathcal{P}ères(n)} \text{Min}_{\gamma_m \in \mathcal{C}_{sm}/\Gamma_m^k} \bar{c}_{\gamma_m} + c_{mn} \end{aligned}$$

Et comme Γ_m^k est un ensemble de j_m^k meilleurs chemins de s à m (lemme 1) :

$$\text{Min}_{\gamma_m \in \mathcal{C}_{sm}/\Gamma_m^k} \bar{c}_{\gamma_m} = c_{j_m^k+1}(m)$$

La relation de récurrence peut également être exprimée en terme de chemins :

$$\forall m \in \mathcal{P}ères(n), \forall \gamma_m \in \mathcal{C}_{sm}/\Gamma_m^k \text{ tel que } \bar{c}_{\gamma_m} + c_{mn} = c_{k+1}(n),$$

$$\mathcal{C}_{sn}^{k+1} = \mathcal{C}_{sn}^k \cup \{\gamma_m \cdot n\} \text{ est un ensemble de } k+1 \text{ meilleurs chemins de } s \text{ à } n.$$

6.4.3 Énumération récursive des k meilleurs chemins

Description

L'algorithme d'énumération récursive de k meilleurs chemins [Jiménez Pelayo 99] est un des plus efficace en terme de temps de calcul : il traduit la relation de récurrence décrite ci-dessus par une fonction récursive calculant un *chemin suivant* pour tout nœud du graphe, défini par la suite directe d'un *chemin suivant* d'un de ses pères.

Pour cela, un $k^{\text{ème}}$ chemin de s à n (avec $s \neq n$ ou $k > 1$) sera décrit par un $i^{\text{ème}}$ chemin de s à m , où m est un nœud père de n : un tel chemin sera caractérisé par son coût, le nœud père m , et l'indice i de ce chemin pour m .

La structure de données utilisée par l'algorithme associe, à chaque nœud du graphe, **un tableau** de chemins connus et **un ensemble** de chemins candidats pour le chemin suivant. Il se décompose en trois phases :

1. **Une phase d'initialisation**, pendant laquelle un plus court chemin (lorsqu'il en existe un) est calculé pour chaque nœud du graphe (avec l'algorithme de Dijkstra par exemple.) Chaque plus court chemin de chaque nœud $n \neq s$ est décrit par un triplet :
 - le coût \bar{c}_{sn} du chemin,
 - le père p_n de n dans ce chemin (tel que $\bar{c}_{sp_n} + c_{p_n n} = \bar{c}_{sn}$),
 - et l'indice $i = 1$.
2. **Une phase itérative** pendant laquelle un ensemble de meilleurs chemins de s à t est élaboré par des appels successifs à une fonction récursive `Chemin_Suivant`. Appliquée à un nœud n , cette fonction :
 - (a) complète les chemins candidats du nœud n :
 - Lors de la recherche d'un second chemin, tous les plus courts chemins entre s et les nœuds pères de n autres que le nœud source p_n (défini dans l'initialisation) forment des candidats pour le second chemin. Dans le cas particulier $n = s$, il n'y a pas de nœud source p_n donc cet ensemble des chemins candidats n'est pas restreint.
 - si $n \neq s$ ou si le chemin cherché n'est pas le second, un autre chemin candidat est ajouté : le chemin suivant du nœud père de n dans le dernier chemin trouvé. Si ce chemin n'est pas encore connu, la fonction `Chemin_Suivant` se rappelle récursivement sur ce nœud père.
 - (b) extrait de l'ensemble des candidat du nœud n celui de coût minimal, qui sera alors le chemin recherché.

Les itérations de `Chemin_Suivant` s'arrêtent lorsque le nombre de chemins trouvés est suffisant, ou lorsqu'il n'y a plus de chemin candidat pour t .

3. Une phase de développement des chemins, à partir des nœuds pères et des indices trouvés.

L'algorithme maintient ainsi, pour chaque nœud exploré, autant de chemins candidats de coût minimal que ce nœud possède de pères (tant que de tels chemins existent). Le meilleur candidat forme donc bien un meilleur chemin suivant, d'après la relation de récurrence.

La terminaison correcte de la fonction récursive est assurée si le graphe ne comporte pas de cycle négatif : les coûts minimaux font référence à des plus courts chemins sans boucle, donc si un $i^{\text{ème}}$ chemin contient plusieurs fois un nœud n , ce chemin sera décrit comme étant la suite d'un $j^{\text{ème}}$ chemin de n , avec $j < i$. La fonction récursive `Chemin_Suivant` ne peut donc pas se rappeler indéfiniment sur ce nœud.

REA(\mathcal{G} , (\bar{c}) , s , t , k) :

soient \mathcal{Q}, \mathcal{K} : Tables sur \mathcal{N}	(Candidats et nombre de chemins par nœuds)
soient $\mathcal{C}, \Gamma, \mathcal{I}$: Tables sur $\mathcal{N} \times \{1, \dots, k\}$	(Coût, nœuds et indice source des $i^{\text{ème}}$ chemins des nœuds)
Chemin_Suivant (n) :	
si $\mathcal{K}(n)$ non défini alors	(Initialisation)
$\mathcal{Q}(n) \leftarrow \{(\bar{c}_{sm} + c_{mn}, m, 1) \mid m \in \mathcal{P}ères(n) \text{ et } \bar{c}_{sm} \neq \infty\}$	
si $n = s$ alors $\mathcal{K}(s) \leftarrow 2$; $\mathcal{C}(s, 1) \leftarrow 0$; $\Gamma(s, 1) \leftarrow (s)$; $\mathcal{I}(s, 1) \leftarrow 0$	
sinon $\mathcal{K}(n) \leftarrow 1$	
sinon	(Recherche du $i^{\text{ème}}$ chemin, avec $i > 1$)
soit $i : \mathcal{I}(n, \mathcal{K}(n)) + 1$ et p : père de n dans $\Gamma(n, \mathcal{K}(n))$	
si $\mathcal{K}(p) < i$ alors <code>Chemin_Suivant</code> (p)	
si $\mathcal{C}(p, i) \neq \infty$ alors	
$\mathcal{Q}(n) \leftarrow \mathcal{Q}(n) \cup \{(\mathcal{C}(p, i) + c_{pn}, p, i)\}$	
$\mathcal{K}(n) \leftarrow \mathcal{K}(n) + 1$	
si $\mathcal{Q}(n) \neq \{\}$ alors	(Extraction et développement d'un chemin)
soit $(c_p, p, i_p) : \text{Extraire_min}(\mathcal{Q}(n))$	
si $\mathcal{K}(p)$ non défini alors <code>Chemin_Suivant</code> (p)	
$\mathcal{C}(n, \mathcal{K}(n)) \leftarrow c_p$; $\Gamma(n, \mathcal{K}(n)) \leftarrow (\Gamma(p, i_p) \cdot n)$; $\mathcal{I}(n, \mathcal{K}(n)) \leftarrow i_p$	
sinon $\mathcal{C}(n, \mathcal{K}(n)) \leftarrow \infty$	
répéter <code>Chemin_Suivant</code> (t) tant que $\mathcal{K}(t) < k$ et $\mathcal{C}(t, \mathcal{K}(t)) \neq \infty$	
retourner $\{\Gamma(t, i) \mid i \in \{1, \dots, \mathcal{K}(t)\} \text{ et } \mathcal{C}(t, i) \neq \infty\}$	

Algorithme III.2 – REA : énumération récursive des k meilleurs chemins entre s et t .

La figure III.2 en décrit une version légèrement adaptée : les trois phases sont réunies en une seule dans la fonction `Chemin_Suivant`, ce qui permet de n'initialiser que les nœuds impliqués dans les k meilleurs chemins de s à t et de ne développer qu'une seule fois chaque portion de chemin : on considère alors que seuls les coûts minimaux $(\bar{c}_{sn})_{n \in \mathcal{N}}$ ont été calculés précédemment. Cette adaptation présente également l'avantage de pouvoir être appliquée à des graphes pour lesquels l'énumération exhaustive de tous les nœuds n'est pas envisageable.

Cette version utilise des structures de données, appelées *tables* : certains cas d'application de l'algorithme pourraient nécessiter l'utilisation de tables associatives ou de hachage, mais dans les cas plus simples (comme celui du graphe des voies de circulation de l'aéroport), ces *tables* peuvent être implémentées par des tableaux :

- \mathcal{Q} associe à chaque nœud du graphe la description de son ensemble de chemins candidats, chacun d'eux étant décrit par un triplet, composé d'un coût, d'un nœud père et de l'indice i du chemin pour ce père. Cet ensemble est manipulé comme une file à priorité : l'opération `Extraire_Min` consiste à enlever et à renvoyer l'élément de coût minimal de cet ensemble.

- \mathcal{K} donne le nombre de chemins connus pour chaque nœud. Lorsqu'un nœud n n'a jamais été parcouru, $\mathcal{K}(n)$ est considéré *non défini* (ce qui pourra être implémenté par $\mathcal{K}(n) = 0$ avec des tableaux).
- \mathcal{C} , Γ et \mathcal{I} décrivent le $i^{\text{ème}}$ chemin de chaque nœud n , pour $i \in \{1, \dots, \mathcal{K}(n)\}$: $\mathcal{C}(n, i)$ en donne le coût, $\Gamma(n, i)$ la suite de nœuds et $\mathcal{I}(n, i)$ l'indice de ce chemin pour son nœud source.

Complexité

La complexité de l'algorithme peut alors être évaluée de la façon suivante :

- `Chemin_Suivant(t)` est appliquée au plus k fois et elle ne peut se rappeler récursivement qu'une fois par appel. Dans le pire des cas, elle se rappelle récursivement jusqu'à arriver au nœud s , donc autant de fois qu'il y a de nœuds dans le chemin cherché, ce qui représente kl itérations.
- Les opérations les plus pénalisantes, à chaque appel de `Chemin_Suivant` sont l'insertion et l'extraction d'un candidat. Elles peuvent être réalisées en $\log d$ opérations pour un ensemble de d éléments, s'il est implémenté par un arbre binaire par exemple. Or, le nombre maximal de candidats par nœuds est le nombre de ses pères, qui peut donc être majoré par d_M , le degré maximal de \mathcal{G} .

REA s'effectue en $\mathcal{O}(kl \log d_M)$ opérations.

$$\text{Où : } \begin{cases} k \text{ désigne le nombre de chemins cherchés,} \\ l \text{ le nombre maximal de nœuds par chemin,} \\ d_M \text{ le degré maximal des nœuds de } \mathcal{G}. \end{cases}$$

Il faut remarquer que le nombre d'appels récursifs de `Chemin_Suivant` est généralement bien inférieur à l , puisque la plupart des chemins sont composés de portions de plus courts chemins : pour ces raisons, l'énumération récursive est souvent beaucoup plus rapide que d'autres méthodes, qui pourraient avoir le même ordre de complexité.

En contrepartie, l'algorithme n'est pas directement adaptable à la recherche des chemins sans boucle, et encore moins à la contrainte de validité formulée au 3.3.4 : dans le cas où un chemin candidat ne serait pas valide pour le nœud considéré, il faudrait chercher le candidat suivant venant du même nœud, ce qui supprimerait la condition d'arrêt de la récursivité.

Cependant, l'efficacité de l'algorithme laisse pleinement envisager la sélection des chemins valides en post-traitement, bien que cela devienne dans ce cas l'opération la plus pénalisante, puisqu'elle nécessite $\mathcal{O}(kl^2)$ opérations. Quelques exemples d'application au graphe des voies de circulation sont fournis et commentés dans la partie 6.4.5.

6.4.4 Recherche de chemins sans boucle

L'algorithme MPS

Le problème de l'énumération des k meilleurs chemins sans boucle dans un graphe orienté est également un problème classique et fortement documenté [Clarke 63, Perko 86, Ringelsen 87]. Dans [Pascoal 98], les auteurs présentent l'algorithme MPS (dont le nom est tiré de leurs initiales), qui s'inspire largement d'une énumération par A* classique : il utilise un unique ensemble de chemins candidats et chaque itération consiste :

- à sélectionner un meilleur chemin parmi tous les candidats,
- puis à rajouter les nouveaux candidats dérivés de ce chemin.

La seule particularité de l'algorithme MPS est d'éliminer en même temps les chemins contenant des boucles. Pour cela, les coûts minimaux (\bar{c}) depuis s sont supposés connus, ainsi qu'un arbre $\mathcal{P} = (p_n)$ de plus court chemins depuis s : chaque nœud $n \neq s$ tel que $\bar{c}_{sn} \neq \infty$ est associé à un nœud père p_n tel que $\bar{c}_{sp_n} + c_{p_n n} = \bar{c}_{sn}$.

Chaque chemin $\gamma_{st} = (s \cdot \dots \cdot t)$ sera alors décomposé en deux parties : un plus court chemin $\gamma_{sn} = (s \cdot \dots \cdot n)$ décrit dans l'arbre \mathcal{P} et un chemin $\gamma_{pt} = (p \cdot \dots \cdot t)$ non décrit par \mathcal{P} . Le nœud particulier n est appelé le *nœud de déviation* du chemin : c'est le dernier nœud n du chemin tel que $(s \cdot \dots \cdot n)$ soit un plus court chemin décrit par \mathcal{P} . Dans l'algorithme, ce *nœud de déviation* est en fait celui qui a permis de générer le nouveau chemin :

1. Initialement, l'ensemble des chemins candidats ne contient qu'un seul élément : le plus court chemin entre s et t décrit par \mathcal{P} . Le nœud de déviation de ce chemin est trivialement t .
2. A chaque itération, un chemin γ_{st} , de coût minimal et de nœud de déviation n est extrait de l'ensemble des chemins candidats :
 - Le nœud de déviation scinde le chemin en deux parties γ_{sn} et γ_{pt} . Si ces deux parties n'ont aucun nœud en commun, alors le chemin est sans boucle et il est le chemin suivant recherché. Dans la négative, aucun nouveau chemin ne sera trouvé lors de cette itération.
 - Dans tous les cas, de nouveaux chemins candidats sont ajoutés : ils sont formés à partir des nœuds pères des nœuds de γ_{sn} , tant qu'ils ne font pas partie de γ_{st} : chacun de ces nœuds m est associé au chemin formé par le plus court chemin entre s et m (décrit dans \mathcal{P}) et par la portion correspondante de γ_{st} . Le nœud m est alors le nœud de déviation de ce nouveau candidat.
3. L'algorithme s'arrête lorsque le nombre de chemins trouvés est k , ou lorsqu'il n'y a plus de chemin candidat.
 - Dans cette version, seuls les coûts minimaux (\bar{c}) depuis s sont supposés connus (l'arbre des plus courts chemins (p_n) n'étant pas réellement nécessaire).

MPS($\mathcal{G}, (\bar{c}), s, t, k$) :

soit $Q : \{(\bar{c}_{st}, t, ())\}$	(Chemins à explorer, classés par coûts)
soit $\mathcal{C} : ()$	(Chemins trouvés)
tant que $ \mathcal{C} < k$ et $Q \neq \{\}$	
soit $(c, n, \gamma^t) : \text{Extraire_Min}(Q)$	
tant que $n \neq s$ et $n \cap \gamma^t = \{\}$	
$\gamma^t \leftarrow (n \cdot \gamma^t)$	
soit p_n : un élément de $\mathcal{P}ères(n)$ tel que $\bar{c}_{sm} + c_{mn} = \bar{c}_{sn}$	
pour chaque $m \in \mathcal{P}ères(n)$ tel que $m \neq p_n$ et $\bar{c}_{sm} \neq \infty$	
$Q \leftarrow Q \cup \{(c + \bar{c}_{sm} + c_{mn} - \bar{c}_{sn}, m, \gamma^t)\}$	
$n \leftarrow p_n$	
si $n = s$ et $s \cap \gamma^t = \{\}$ alors Ajouter $(s \cdot \gamma^t)$ à \mathcal{C}	
retourner \mathcal{C}	

Algorithme III.3 – MPS : énumération des k meilleurs chemins sans boucle.

L'algorithme III.3 est une adaptation de MPS :

- Q est l'ensemble des chemins candidats. Chacun d'eux est décrit par un triplet : son coût, son nœud de déviation n et la portion γ_{pt} permettant de relier n à t (la partie γ_{sn} n'a pas besoin

d'être précisée, puisqu'elle peut-être trouvée directement pendant la génération des nouveaux candidats, en remontant les arcs du graphe de pères en pères).

- L'absence de boucle est remplacée par la contrainte de validité définie dans la partie 3.3.4. Dans la mesure où cette contrainte est plus forte que l'interdiction de repasser plusieurs fois par le même nœud, l'algorithme risque d'effectuer plus d'itérations mais ne perd pas sa validité.

Présenté de cette façon, la similitude entre cet algorithme et celui de Dev_Max (algorithme III.1) est assez évidente : le développement des chemins de pères en pères est en tout point identique, mais il est cette fois-ci effectué dans l'ordre de leurs coûts, selon le principe de développement d'un algorithme A^* .

Complexité

La complexité de l'algorithme MPS dépend naturellement des cycles que possède le graphe :

- Le nombre d'itérations principales \bar{k} sera le nombre de chemins trouvés, augmenté du nombre de cycles qui peuvent faire partie d'un chemin de moindre coût entre s et t , que l'on ne peut pas connaître a priori.
- À chaque itération principale, les nœuds situés entre le nœud de déviation du chemin courant et s sont visités, ce qui représentera au maximum l itérations (l étant la longueur maximale des chemins).
 - Chacun de ces nœuds doit être comparé à la deuxième portion du chemin courant, ce qui demande $\mathcal{O}(l)$ opérations.
 - Pour chacun des pères de ces nœuds, un nouveau candidat est ajouté, ce qui demande $\mathcal{O}(d_M \log m)$ opérations, si d_M est le degré maximal du graphe, m le nombre d'arcs et si l'ensemble \mathcal{Q} est implémenté avec une structure de données appropriée (sous forme d'un arbre binaire par exemple).

Chaque itération principale nécessite donc $\mathcal{O}(l(l + d_M \log m))$ opérations.

On obtient alors :

$$\text{MPS s'effectue en } \mathcal{O}(\bar{k}l^2 + \bar{k}ld_M \log m) \text{ opérations.}$$

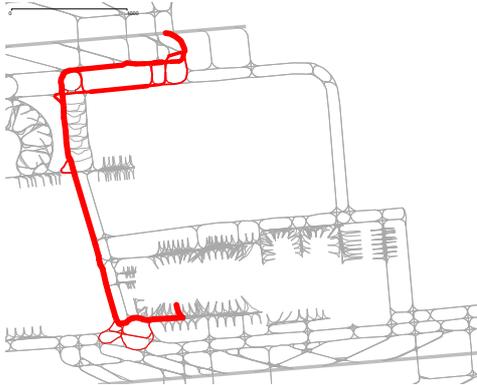
$$\text{Où : } \begin{cases} \bar{k} \text{ désigne le nombre de chemins parcourus } (\bar{k} \geq k), \\ l \text{ le nombre maximal de nœuds par chemin,} \\ d_M \text{ le degré maximal des nœuds de } \mathcal{G}, \\ m \text{ le nombre d'arcs de } \mathcal{G}. \end{cases}$$

La complexité ainsi formulée apparaît plus élevée que celle de l'énumération récursive (même si l'on ajoute à cette dernière le coût du traitement de sélection des chemins valides). Cependant, l'algorithme MPS semble plus adapté au problème du cheminement des avions, puisqu'il est le seul, parmi les trois étudiés, qui permette réellement d'obtenir un nombre de chemins fixé à l'avance.

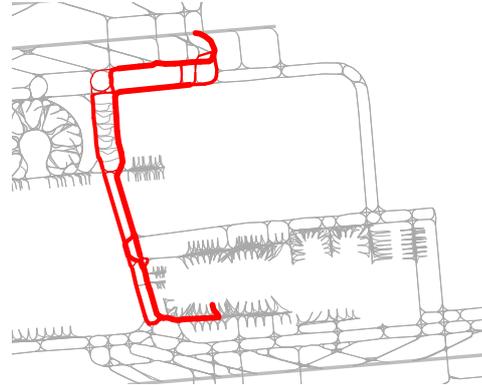
De plus, dans le cas des voies de circulation d'un aéroport, le degré des nœuds est relativement faible par rapport au nombre de nœuds par chemins : $\mathcal{O}(\bar{k}l^2)$ sera donc généralement le terme le plus pénalisant, ce qui revient à dire que les deux algorithmes (REA avec post-traitement et MPS) devraient être équivalents.

6.4.5 Résultats

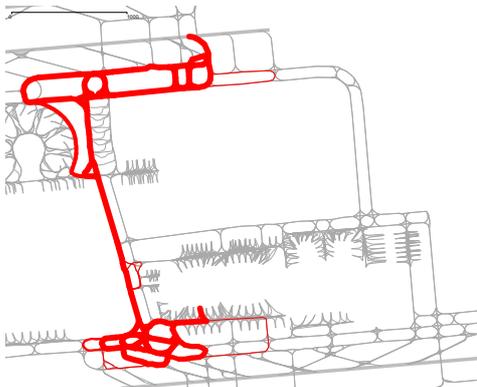
La figure 6.2 montre quelques exemples d'application des deux algorithmes d'énumération des k meilleurs chemins. Pour l'algorithme REA, les temps de calcul et les chemins représentés tiennent compte du post-traitement (l'élimination des chemins contenant des boucles) : on peut remarquer la différence entre k et le nombre effectif de chemins trouvés.



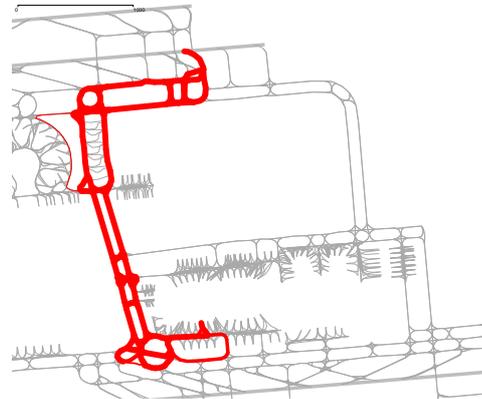
1.1 Roissy : REA avec $k = 50$
Avec sens préférentiels
36 chemins (trouvés en 39 ms.)



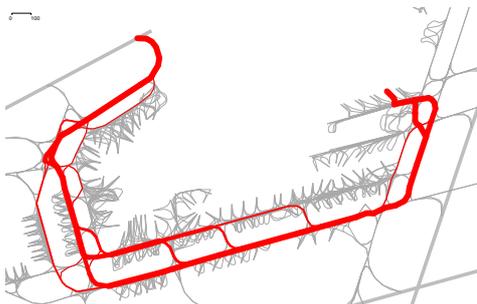
1.2 Roissy : MPS avec $k = 50$
Sans sens préférentiels
50 chemins (trouvés en 40 ms.)



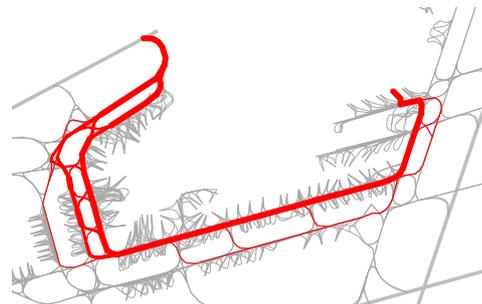
2.1 Roissy : REA avec $k = 2000$
Avec sens préférentiels
605 chemins (trouvés en 1.7 s.)



2.2 Roissy : MPS avec $k = 2000$
Sans sens préférentiels
2000 chemins (trouvés en 2 s.)



3.1 Orly : REA avec $k = 50$
Avec sens préférentiels
36 chemins (trouvés en 39 ms.)



3.2 Orly : MPS avec $k = 50$
Sans sens préférentiels
50 chemins (trouvés en 43 ms.)

FIG. 6.2 – Applications de REA et MPS

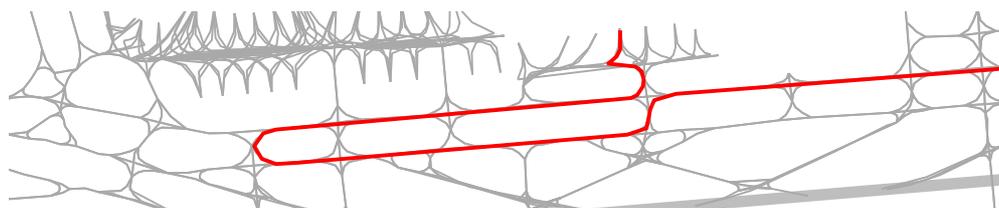


FIG. 6.3 – Exemple de détour difficilement utile

Dans les deux premiers exemples (1.1 et 1.2 à Roissy) ainsi que dans les deux derniers (3.1 et 3.3 à Orly), les ensembles de chemins trouvés sont composés de beaucoup moins de chemins que précédemment et semblent explorer les mêmes voies principales de l'aéroport : ces algorithmes permettent donc de diminuer le nombre de chemins tout en conservant un ensemble d'alternatives presque aussi diversifiées qu'avec l'algorithme *Dev_Max*.

Il faut également remarquer qu'en terme de temps de calcul, et pour des valeurs de k compatibles avec un nombre de chemins à proposer aux avions, ces deux algorithmes d'énumération sont finalement beaucoup moins coûteux et beaucoup plus constants (que *Dev_Max*) : ceci laisse pleinement envisager d'affiner la sélection des chemins si besoin en est (lors de post-traitements) en fonction d'autres critères que pourraient rajouter les opérationnels.

Cependant, la diversité des chemins est loin d'être suffisante : à Roissy, avec ou sans sens uniques, le *contournement Est* n'est pas envisagé, même avec $k = 2000$ chemins (exemples 2.1 et 2.2). Ce contournement est pourtant souvent proposé par les contrôleurs, car il permet d'éviter la congestion de l'axe central de l'aéroport. Dans le cas particulier de Roissy, les ensembles de chemins ne sont donc toujours pas satisfaisants : ceci est dû au nombre de chemins (beaucoup trop important) qu'il faudrait explorer avant d'espérer obtenir les détours intéressants.

C'est pourquoi il est nécessaire d'affiner encore les algorithmes de recherche de chemins : dans ce cadre, l'objectif ne sera plus obligatoirement de trouver un ensemble de *meilleurs chemins*, mais plutôt de trouver des chemins aussi *diversifiés* que possible.

6.5 Recherche avec variation des coûts

6.5.1 Objectifs

Les résultats trouvés précédemment montrent que l'ensemble des chemins envisageables pour un avion ne peut se définir aussi simplement qu'un ensemble de plus courts chemins valides (leur nombre est beaucoup trop élevé) : pour conserver un nombre réduit de chemins à envisager pour chaque avion, tout en utilisant au mieux les voies de circulation de l'aéroport, il s'avère nécessaire d'effectuer la sélection des chemins sur de nouveaux critères.

Certains détours peuvent effectivement sembler totalement absurdes aux yeux des opérationnels alors qu'ils sont moins coûteux que d'autres chemins acceptables (figure 6.3). Ceci s'explique par l'utilité potentielle que peut avoir un détour :

- Les petits détours peuvent permettre de faciliter localement certaines manœuvres d'évitement (croisement ou dépassement d'un autre avion par exemple).
- Des déviations plus grandes (ou contournements) permettent d'éviter toute une zone spécifique de l'aéroport (lorsque celle-ci est congestionnée, ou momentanément indisponible par exemple).
- En contrepartie, les grands détours n'évitant aucune zone particulière de l'aéroport seront jugés sans intérêt, bien que cela ne puisse être formulé plus rigoureusement tant que ces zones ne sont

pas explicitement décrites.

Ces considérations incitent à s'intéresser à un nouveau problème : la recherche de chemins suffisamment *différents*. On ne souhaite plus obtenir un ensemble de meilleurs chemins, mais plutôt plusieurs composantes de chemins, suffisamment distinctes. Ceci représente en réalité un compromis entre le coût des chemins (que l'on cherche malgré tout à minimiser) et leur diversité (que l'on cherche alors à maximiser). Ce compromis sera matérialisé par un coefficient de pondération, représentant le rapport entre le coût d'un nouveau chemin et sa redondance avec ceux déjà trouvés.

6.5.2 Variation des coûts

Description

Une solution simple pour trouver des chemins *différents* consiste à augmenter le coût des arcs empruntés par les chemins déjà trouvés. Pour conserver malgré tout un nombre fixé de *petits détours*, il suffit de rechercher un sous-ensemble de meilleurs chemins entre chaque mise à jour des coûts :

- Le nombre d'itérations ν du procédé correspondra au nombre de composantes recherchées.
- k sera le nombre de chemins (de coûts voisins) par composante.
- δ , le facteur d'augmentation des coûts des arcs déjà explorés, définit le compromis entre coût et diversité.

Deviations($\mathcal{G}, s, t, \nu, k, \delta$):

<p>pour chaque $(m, n) \in \mathcal{A}$, soit $c'_{mn} : c_{mn}$ soit $\mathcal{G}' : \mathcal{G}$ avec les coûts définis par c' soit $\mathcal{C} : \{\}$ pour i de 1 à ν, faire soit (\vec{c}') : Dijkstra(\mathcal{G}', s) $\mathcal{C} \leftarrow \mathcal{C} \cup \text{MPS}(\mathcal{G}', (\vec{c}'), s, t, k)$ si $i < \nu$ alors pour chaque (m, n) arc d'un chemin de \mathcal{C}_i, $c'_{mn} \leftarrow \delta \times c_{mn}$ retourner \mathcal{C}</p>	(Chemins trouvés)
--	-------------------

Algorithme III.4 – Deviations : recherche de chemins suffisamment différents.

La stratégie d'augmentation des coûts à chaque itération peut être envisagée de deux façons : soit cumulée (le coût d'un arc est augmenté autant de fois qu'il y a de chemins utilisant cet arc), soit unique (le coût d'un arc ne peut alors varier qu'une seule fois). L'ordre de grandeur de δ devra naturellement être choisi en fonction de l'option choisie (plus faible dans l'option cumulée). En ce qui concerne la recherche de chemins sur l'aéroport, l'augmentation unique des coûts semble mieux adaptée, car elle permet de borner plus facilement les coûts considérés, ce qui évite par exemple que certains *sens interdits* soient finalement proposés.

L'algorithme d'itération III.4, intitulé Deviations réalise cette recherche de chemins.

Mises à jour des coûts minimaux

Dans la formulation de l'algorithme III.4, la mise à jour des plus courts chemins depuis le nœud origine se fait par l'application de l'algorithme de Dijkstra, qui est ainsi recalculé entièrement à chaque itération principale.

De nombreux algorithmes permettent d'optimiser cette mise à jour, pour les graphes *dynamiques* dont les coûts varient (une version répandue est formulée dans [Ramalingam 96]) : ces algorithmes procèdent par exploration des arcs modifiés, en commençant par ceux situés en bordure de l'arbre des plus courts chemins à mettre à jour. Lorsque plusieurs arcs sont concernés par la mise à jour, la complexité de ces algorithmes reste du même ordre que celle du Dijkstra, bien qu'ils soient en pratique plus efficaces lorsque l'ordre du graphe augmente.

La taille du graphe des voies de circulation de l'aéroport étant relativement faible pour ce type d'exploration, ces algorithmes ne sont pas plus efficaces sur le problème que le Dijkstra lui-même, qui se révèle très peu coûteux en temps de calcul (moins de 10 millisecondes par appel, avec un processeur à 1.6 GHz).

6.5.3 Résultats

La figure 6.4 montre quelques exemples d'application de *Deviations* associée à MPS sur Roissy et Orly, avec un nombre de chemins réduit à 12 et un pourcentage de variation des coûts de 300%. Cet ordre de grandeur de δ signifie qu'une portion de chemin pourra être envisagée même si elle est trois fois plus coûteuse qu'une portion déjà explorée.

A Roissy (exemples 1.1 à 2.2), l'algorithme est appliqué avec $\nu = 3$ itérations de $k = 4$ chemins, avec ou sans sens préférentiels. Le *contournement Est* est systématiquement envisagé, même lorsqu'il représente un détour important. De plus, les chemins trouvés sont globalement plus satisfaisants car beaucoup moins tortueux.

A Orly, les voies de circulation sont beaucoup plus simples : les k meilleurs chemins obtenus avec l'algorithme MPS étaient déjà satisfaisants. Le premier exemple (3.1) montre que l'utilisation des mêmes paramètres ($\nu = 3$, $k = 4$, et $\delta = 300\%$) dénature les sens préférentiels de circulation. Ceci est dû aux poids associés aux transitions déconseillées, qui sont plus faibles qu'à Roissy. On pourra donc en rester à l'algorithme MPS, ce qui revient à n'effectuer qu'une seule itération de 12 chemins (exemple 3.2).

Pour les deux aéroports étudiés, on parvient ainsi à trouver un ensemble réduit et assez représentatif de chemins à envisager. Lorsqu'il s'agira des chemins associés à un mouvement, il faudra considérer toutes les bretelles de piste utilisables par l'avion, ce qui sera fait simplement en réitérant l'algorithme *Deviations* sur chaque bretelle (les temps d'exécution de l'algorithme étant tout à fait compatibles avec ces itérations).

6.6 Conclusion

Une première méthode générale de recherche de chemins appliquée aux voies de circulation de Roissy montre que le nombre de possibilités permettant de rejoindre deux points donnés de l'aéroport est beaucoup trop grand pour que chacune de ces possibilités puissent être raisonnablement envisagées.

Par contre, une méthode itérative de parcours de graphe avec variation des coûts permet d'obtenir un ensemble réduit de chemins, résultat d'un compromis entre leurs temps de parcours (que l'on cherche à minimiser) et leur diversité (que l'on cherche à maximiser).

Chapitre 7

Résolution par des méthodes déterministes

7.1 Introduction

Le problème global posé par chaque situation de trafic est fortement combinatoire. Tel qu'il est formulé au chapitre 5, il ne peut généralement pas être résolu de manière optimale par des méthodes déterministes (car les temps de calcul seraient prohibitifs). Dans ce chapitre, les solutions recherchées ne seront donc plus forcément optimales, mais elles n'en resteront pas moins des solutions admissibles pouvant s'avérer intéressantes :

- Pour être applicables, les méthodes déterministes doivent considérer un espace de recherche restreint, en ajoutant par exemple des règles de priorité entre avions. Les solutions obtenues sont alors des optima sur cet espace restreint, ce qui correspond généralement à des optima locaux du problème initial.
- L'obtention d'une solution admissible, même si elle ne correspond qu'à un optimum local, fournit une base de comparaison utile pour mesurer l'efficacité d'autres méthodes de résolution plus aléatoires.
- Lorsque la restriction opérée sur l'espace de recherche autorise une exploration rapide de cet espace, la méthode peut être améliorée par itération sur différents sous-espaces. Ceci est particulièrement intéressant lorsque la restriction initiale de l'espace ne contient pas de solutions admissibles (ce qui est le plus à craindre dans ce contexte) : le problème peut alors être reconsidéré (modification des priorités) pour explorer un autre sous-espace de solutions.

Dans une première partie, le concept général permettant de restreindre l'espace de recherche est présenté et analysé : les avions sont considérés classés par ordre de priorité. Ce concept sera tout d'abord mis en œuvre sur un problème simplifié, sans incertitude (deuxième partie). Il sera ensuite adapté et amélioré pour être finalement applicable au problème global avec incertitudes (dernières parties).

7.2 La méthode 1 contre n

7.2.1 Définition

La partie 5.3.1 montre que le problème global est combinatoire par rapport à la taille de l'horizon de prédiction et au nombre d'avions impliqués. Ceci laisse présager deux possibilités pour le

simplifier :

- La première consiste à séquencer la résolution sur l’horizon de prédiction : ce dernier serait divisé en plusieurs périodes, qui pourraient être traitées les unes après les autres. Cette éventualité permettrait de résoudre des situations définies sur un horizon de prédiction étendu, mais n’apporte pas réellement de solutions intéressantes pour les simulations, puisqu’elle revient simplement à diminuer l’horizon de prédiction sur lequel la situation est réellement optimisée : cette éventualité ne sera donc pas étudiée.
- La seconde consiste à séquencer la résolution par avions : les avions sont considérés les uns après les autres. Dans ce contexte, le problème élémentaire à résoudre est celui d’un seul avion devant en éviter d’autres : les trajectoires des autres avions, considérées comme fixes, forment les contraintes du problème et le critère à optimiser n’est relatif qu’à l’avion concerné.

La résolution du problème global se fera par itération des résolutions élémentaires du problème lié à chaque avion, dans l’ordre donné par un classement des avions en conflit : dans toute la suite, cette approche du problème sera appelée la méthode 1 contre n .

7.2.2 Analyse du problème

La méthode 1 contre n se décompose en réalité en deux sous problèmes :

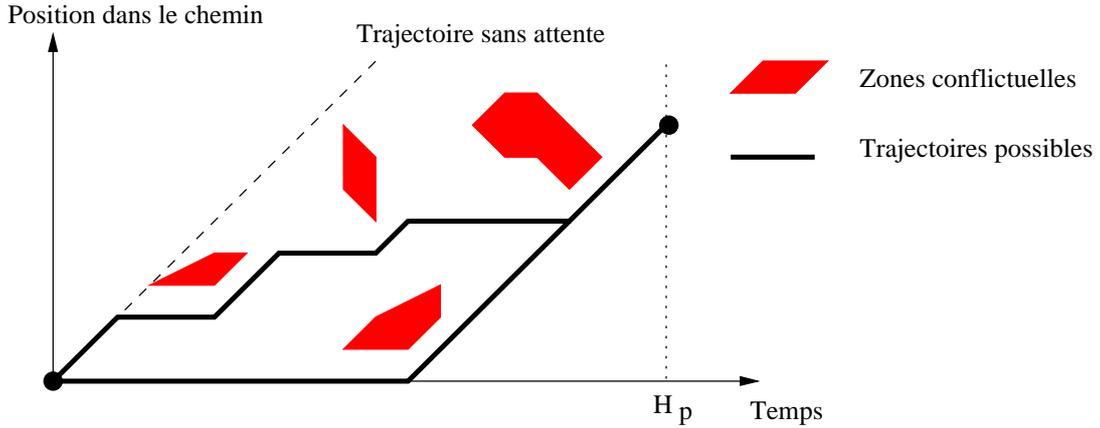
1. **Les avions doivent être classés.** Ce point sera déterminant pour la résolution du problème, car beaucoup de classements peuvent aboutir à un problème sans solution pour les derniers avions considérés. Plus généralement, la qualité de la solution qui sera trouvée dépendra largement de l’adéquation du classement avec la situation courante des avions. Une étude de la situation courante pourra orienter le classement à opérer, mais seule l’application de la méthode 1 contre n à ce classement pourra réellement évaluer sa qualité.
2. **Une trajectoire optimale doit être trouvée** pour chaque avion. Pour ce faire, l’ensemble des choix successifs qui s’offrent à l’avion sera représenté par un arbre, ou plus généralement par un graphe (car certaines séquences de choix peuvent mener à la même position finale) : chaque arc du graphe représente un choix particulier pour l’avion et son coût doit correspondre à la pénalité résultante selon le critère d’optimisation global. Le critère doit donc être décomposé en fonction de chaque possibilité qui s’offre à l’avion, ce qui implique que la méthode de résolution 1 contre n ne pourra pas être totalement indépendante du critère global : lorsque ce critère sera modifié, les coûts associés aux arcs du graphe relatif aux possibilités pour un avion devront être revus, ce qui pourra entraîner l’adaptation de l’algorithme de résolution lui-même.

7.2.3 Complexité

La complexité de la méthode 1 contre n est relative au nombre de nœuds du graphe \mathcal{G}_a représentant les possibilités pour un avion a :

- Sur un chemin particulier de l’avion, le problème apparaît comme la recherche d’une trajectoire optimale dans un sous-graphe de \mathcal{G}_a : cette trajectoire doit contourner les obstacles représentés par les autres avions. Plus précisément, les positions successives de l’avion ne doivent pas être conflictuelles avec les trajectoires des avions déjà traités (figure 7.1).

Dans ce graphe relatif à un chemin particulier, l’avion a deux choix à chaque pas de temps de l’horizon de prédiction : avancer ou attendre. Ceci représente à priori 2^{H_p} possibilités, mais beaucoup d’entre elles aboutissent à la même position :

FIG. 7.1 – Illustration du problème 1 contre n sur un chemin

Sans incertitude sur la vitesse de déplacement, le nombre de positions possibles de l'avion à chaque pas de temps t s'écrit : $p(t) = t + 1$.

Le nombre de nœuds du graphe sur un chemin particulier est donc :

$$P = \sum_{t=0}^{H_p} t + 1 = \mathcal{O}(H_p^2)$$

Avec incertitudes, le nombre de positions possibles de l'avion à chaque pas de temps t est fonction de la largeur maximale $n_{\max}(t)$ de sa zone d'incertitude à cet instant :

$$p(t) \leq n_{\max}(t)(t + 1)$$

Où $n_{\max}(t) = 1 + E(\delta_v(\Delta + t))$ (δ_v étant le taux d'incertitude sur la vitesse) comme cela a été décrit dans la partie 4.2.4.

Le nombre de nœuds du graphe sur un chemin particulier s'écrit donc :

$$P \leq \sum_{t=0}^{H_p} (1 + E(\delta_v(\Delta + t)))(t + 1) = \mathcal{O}(\delta_v H_p^3)$$

- Si l'on considère l'ensemble des η chemins d'un avion a , la taille de l'espace de recherche correspond au nombre total $|\mathcal{N}_a|$ de nœuds du graphe et s'écrit donc :

$$|\mathcal{N}_a| = \mathcal{O}(\eta H_p^2) \text{ sans incertitude} \quad \text{et} \quad |\mathcal{N}_a| = \mathcal{O}(\eta \delta_v H_p^3) \text{ avec incertitudes}$$

- Pour N avions classés, l'espace de recherche $|\mathcal{E}|$ est donc restreint à :

$$|\mathcal{E}| = \mathcal{O}(N \eta H_p^2) \quad (\text{sans incertitude})$$

$$|\mathcal{E}| = \mathcal{O}(N \eta \delta_v H_p^3) \quad (\text{avec incertitudes}).$$

La méthode 1 contre n apparaît ainsi compatible avec l'utilisation de méthodes déterministes :

$$\text{Pour } \begin{cases} N & = 5 \text{ avions} \\ \eta & = 10 \text{ chemins par avions} \\ \delta_v & = 20\% \text{ d'incertitude sur la vitesse} \\ H_p & = 5 \text{ minutes (ou 60 pas de 5 secondes)} \end{cases}$$

L'espace de recherche est réduit à environ deux millions de solutions.

7.3 Classement fixe sans incertitude

7.3.1 Principes

Dans cette partie, le problème 1 contre n est considéré sans incertitude, avec un classement absolu de tous les avions d'une journée de trafic (comme cela a été effectué dans [Pesic 00] avec une modélisation plus simple de l'aéroport). Cette approche du problème présente plusieurs intérêts :

- L'absence d'incertitude permet de s'affranchir de l'effet de l'horizon : la prédiction de trafic est exacte et l'horizon peut donc être considéré comme infini. Sous certaines conditions sur le classement des mouvements (détaillées dans la partie suivante), ceci diminue fortement les risques d'absence de solution pour un avion.
- Elle permet d'évaluer rapidement les délais induits par la globalité du trafic d'une journée. Même si elle ne pourrait pas s'appliquer dans un cadre opérationnel en temps réel (à cause de l'absence d'incertitude), elle fournit une version dynamique du trafic en donnant une première estimation des temps de roulage et en dévoilant les périodes réellement chargées de la journée.

Dans ce cadre, le modèle de simulation présenté dans la partie 4.4 devient caduque : les mouvements de la journée peuvent être traités intégralement les uns après les autres, indépendamment de tout horizon de prédiction.

7.3.2 Classement des avions

Plusieurs stratégies peuvent être envisagées pour classer les vols, la principale difficulté étant de trouver un classement qui n'aboutisse pas à un problème sans solution :

- Une contrainte forte doit notamment être prise en compte pour les arrivées : elles ne peuvent qu'être modérément retardées avant l'atterrissage et elles ne peuvent pas changer d'ordre. Les arrivées doivent donc être classées dans l'ordre des heures prévues d'atterrissage et doivent être prioritaires sur les décollages pour bénéficier de la piste libre.
- Pour les départs, la possibilité d'attendre au parking n'est pas limitée, ce qui permet d'assurer l'existence d'une solution dans la plupart des cas (les parkings sont supposés séparés des voies de circulation). Seuls les cas de partage du parking avec une arrivée pourront limiter cette attente initiale. Deux classements des départs peuvent donc être envisagés : par heure de début du mouvement ou par heure prévue à la piste.

Pour ces raisons, deux classements différents seront comparés :

1. **Le classement par heure début** : les arrivées sont prioritaires sur les départs et les mouvements de même type sont classés par heure de début du mouvement (heure d'atterrissage pour une arrivée et heure de début du roulage pour les départs).
2. **Le classement par heure piste** : les arrivées restent prioritaires sur les départs mais les mouvements de même type sont classés par heure d'accès à la piste (heure d'atterrissage pour une arrivée et heure de décollage prévu pour les départs).

7.3.3 Optimisation de la trajectoire d'un avion

Définition du graphe des possibilités

Le graphe des possibilités d'un avion a , débutant au pas de temps t_a et pour lequel η chemins sont envisageables se définit comme suit :

- **Chaque nœud du graphe** doit représenter une position datée de l'avion dans un de ses chemins. Un tel nœud est ainsi décrit par un triplet (ν, p, t) :
 - $\nu \in \{1, \dots, \eta\}$ est l'indice du chemin sur lequel se trouve l'avion ;
 - $p \in \{1, \dots, l(\nu)\}$ est la position de l'avion dans ce chemin ($l(\nu)$ est le temps de roulage sans attente associé au chemin ν , exprimé en pas de temps) ;
 - $t \geq t_a$ est l'heure à laquelle l'avion se trouve sur cette position, exprimée en pas de temps.
- **Les nœuds initiaux** correspondent à la première position de l'avion sur chacun de ses chemins, à l'heure t_a (cette position est située au parking pour un départ et en l'air pour une arrivée) :

$$\mathcal{I} = \{(\nu, 1, t_a) \mid 1 \leq \nu \leq \eta\}$$

- **Les nœuds fils** d'un nœud (ν, p, t) représentent les futures positions que l'avion peut atteindre : si la position datée est conflictuelle avec un des avions déjà traités, le nœud n'a aucun fils. Sinon, le nœud peut avoir un ou deux fils, selon que l'avion peut attendre ou non sur cette position :
 1. Soit il avance : le premier nœud fils s'écrit donc $(\nu, p+1, t+1)$ et le coût pour l'atteindre est $\alpha = \frac{3}{2}$, en accord avec le critère global, qui pénalise plus le temps de roulage que l'attente.
 2. Soit il maintient sa position, lorsque cette position peut-être maintenue : le second nœud fils est alors $(\nu, p, t+1)$ et le coût de cette attente élémentaire est 1.
- **Les nœuds terminaux** (ou solutions) sont les nœuds représentant une position finale non conflictuelle de l'avion sur un de ses chemins (en l'air pour un départ et au parking pour une arrivée) :

$$\mathcal{T} = \{(\nu, l(\nu), t) \mid 1 \leq \nu \leq \eta \text{ et } t_a \leq t\}$$

Résolution par A*

Dans le graphe des possibilités, les nœuds représentant des positions conflictuelles n'ont aucun fils : le graphe modélise les contraintes de séparations des avions. Tout chemin de ce graphe joignant un nœud initial à un nœud terminal représente donc une solution admissible pour l'avion. Cette solution sera optimale si ce chemin est de coût minimal dans le graphe.

Un algorithme A* permet donc de trouver une solution admissible optimale s'il en existe une et s'il est muni d'une heuristique minorante (annexe D.4). Cette heuristique peut se définir trivialement par le temps de roulage restant sans attente pondéré par α , soit :

$$h(\nu, p, t) = \alpha(l(\nu) - p)$$

De plus, l'étude du graphe des possibilités pour un avion permet d'optimiser l'implémentation de l'algorithme A* : dans ce graphe, tous les chemins menant à un même nœud sont nécessairement de même coût, en accord avec le critère d'optimisation. Pour parvenir à un nœud (ν, p, t) , l'avion a

```

A*_1n((ai)1≤i≤N, a) :
  soit Ψ : {(αl(ν), 0, (ν, 1, ta)) | 1 ≤ ν ≤ ηa} (Nœuds à développer)
  ∀ν ∈ {1, ..., ηa} ∀p ∈ {1, ..., l(ν)} soit tmax(ν, p) : ta - 1 (Caractérisation des nœuds développés)
  tant que Ψ ≠ {}
    soit (f, c, γ) : Extraire_Min(Ψ) (Un élément de c maximal parmi ceux minimisant f)
    soit (ν, p, t) : le dernier nœud de γ
    si tmax(ν, p) < t et (ν, p, t) non conflictuel avec (ai)1≤i≤N alors
      si p < l(ν) alors (Nœud non terminal)
        si tmax(ν, p + 1) < t + 1 alors Ψ ← Ψ ∪ {(f, c + α, (γ · (ν, p + 1, t + 1)))}
        Ψ ← Ψ ∪ {(f + 1, c + 1, (γ · (ν, p + 1, t)))}
      sinon retourner γ (Solution optimale trouvée)
    tmax(ν, p) ← t (Mémorisation des nœuds développés)
  retourner {} (Problème sans solution)

```

Algorithme III.5 – A* pour la résolution du problème 1 contre n

nécessairement avancé $p - 1$ fois et attendu $t - t_a - (p - 1)$ pas de temps. Le coût pour arriver à ce nœud est donc :

$$c(\nu, p, t) = \alpha(p - 1) + t - t_a - (p - 1) = (\alpha - 1)(p - 1) + t - t_a$$

Ajoutée à l'heuristique, cette expression permet de minorer le coût de tout chemin solution passant par un nœud (ν, p, t) , par la fonction f suivante :

$$f(\nu, p, t) = c(\nu, p, t) + h(\nu, p, t) = t - t_a - p + \alpha l(\nu) + 1 - \alpha$$

L'exploration des nœuds du graphe par l'algorithme A* se fait par f croissante : si $(\nu_i, p_i, t_i)_i$ désigne une suite de nœuds donnés dans l'ordre de leur développement, on aura :

$$\forall i < j, t_i - p_i + \alpha l(\nu_i) \leq t_j - p_j + \alpha l(\nu_j)$$

Rapportée à une position précise p sur un chemin particulier ν et à une suite de nœuds développés $(\nu, p, t_i(\nu, p))$, la relation précédente aboutit à une relation stricte car l'algorithme ne développe jamais deux fois le même nœud :

$$\forall i < j, t_i(\nu, p) < t_j(\nu, p)$$

Le caractère strictement croissant de la suite $(t_i(\nu, p))_i$ permet de caractériser directement les nœuds développés. Si $t_{\max}(\nu, p)$ désigne le pas de temps maximal des nœuds (ν, p, t) développés, on aura :

$$(\nu, p, t) \text{ développé} \iff t \leq t_{\max}(\nu, p)$$

Cette relation implique également que lors du développement d'un nœud, l'éventuel nœud fils correspondant à l'arrêt de l'avion n'est pas encore développé :

$$(\nu, p, t) \text{ non développé} \implies (\nu, p, t + 1) \text{ non développé}$$

L'algorithme III.5 donne la version ainsi optimisée de l'A*.

7.3.4 Application

L'application de la méthode 1 contre n avec classement fixe sans incertitude donne lieu à 4 simulations : chacune des deux méthodes de classement envisagée (par heure début ou par heure piste) est appliquée aux deux aéroports modélisés (Roissy et Orly), sur une journée de trafic chargée, avec les paramètres suivants :

- Journée du 22/03/2003 à Roissy (1461 mouvements)
- Journée du 18/06/1999 à Orly (807 mouvements) ;
- Vitesse de roulage en ligne droite : 10 m/s ;
- Sens préférentiels de circulation : appliqués ;
- Nombre maximum de chemins envisagés par avion et par bretelle de piste : 12 ;
- Norme de séparation au roulage : 60 mètres ;
- Normes de séparation pour les séquences de piste : 1, 2 ou 3 minutes ;
- Largeur des aires de piste : 90 mètres ;
- Attente après *push-back* : 1 minute.

Le tableau suivant donne les résultats globaux, obtenus avec un processeur à 1.6 GHz :

Aéroport	Classement	Temps de calcul	Pénalité moyenne (sec.)	Cas non résolus
Roissy	Heure début	8 min.	103	1
	Heure piste	8 min.	99	1
Orly	Heure début	3 min.	37	5
	Heure piste	3 min.	37	5

Les temps de calcul de chaque simulation montrent l'intérêt de la méthode 1 contre n : elle permet une analyse rapide du trafic d'une journée.

L'étude des cas non résolus montre qu'il s'agit d'une même cause et des mêmes mouvements dans les deux simulations de chaque aéroport : un départ ne peut libérer le parking avant que l'arrivée suivante n'y parvienne. En pratique, il suffirait de mettre en attente l'arrivée, mais cette solution ne peut être trouvée, puisque les arrivées doivent être prioritaires sur les départs au moment de l'atterrissage. Ceci souligne la principale limite de la méthode 1 contre n avec classement fixe : la nécessité de définir un ordre total sur les avions rend certaines situations (pourtant simples pour un opérationnel) sans solution.

La figure 7.2 donne la pénalité moyenne infligée à chaque mouvement en fonction du nombre d'avions qu'il doit éviter, dans chaque simulation.

- À Roissy, la méthode de classement par heure prévue à la piste s'avère légèrement moins pénalisante pour un nombre d'avions moyens (entre 30 et 45 avions). Ceci est dû à la grande diversité des temps de roulage sans attente : les avions devant parcourir un chemin plus court peuvent arriver plus vite devant la piste. Lorsque l'attente pour la piste est réduite, le classement par heure prévue à la piste les rend prioritaires par rapport aux autres départs et diminue ainsi leur retard, tandis que l'autre classement oblige ces mouvements à s'écarter pour laisser passer d'autres avions plus prioritaires. Lorsque les files d'attente pour la piste deviennent conséquentes (nombre d'avions plus élevé), les deux classements aboutissent finalement aux mêmes pénalités (les délais dus à la piste absorbent les délais dus au roulage).
- À Orly, les deux classements sont équivalents : les voies de circulation sont plus simples et la longueur des chemins n'est pas suffisamment variable pour que le même phénomène soit mesurable.
- Les domaines de variations des courbes illustrent la différence de taille entre Roissy et Orly : pour un même nombre d'avions, la pénalité est 3 à 4 fois plus importante à Orly (l'aéroport est

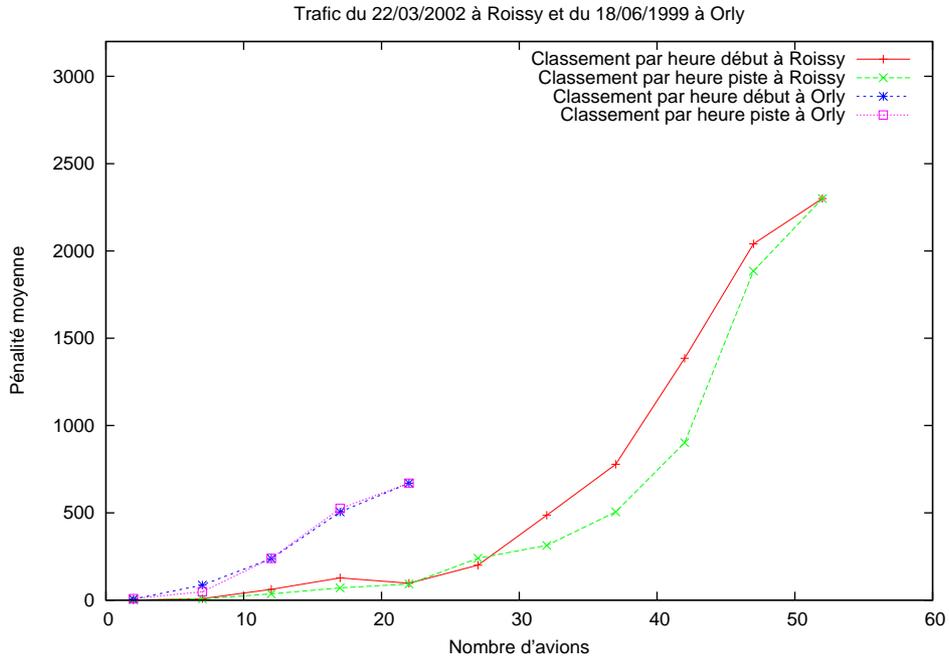


FIG. 7.2 – Pénalité en fonction du nombre d'avions sans incertitude

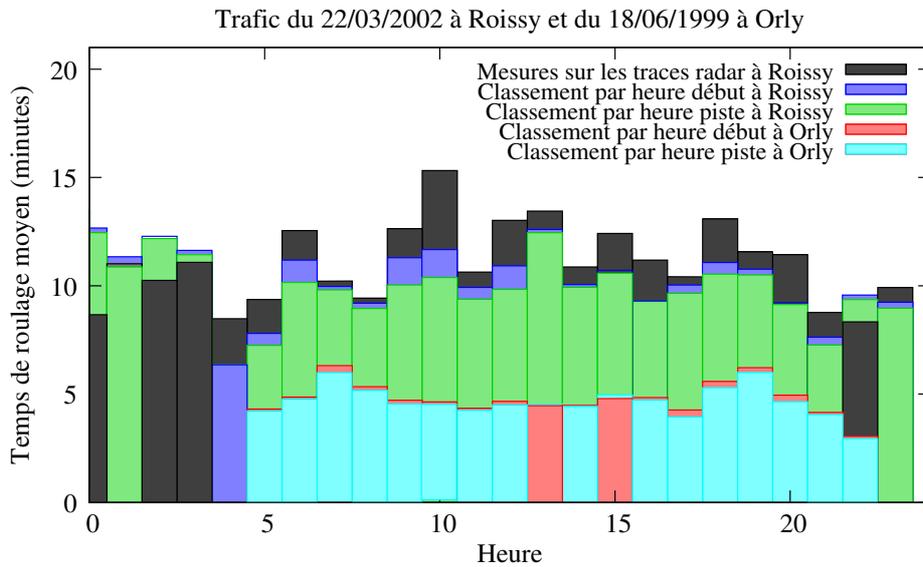


FIG. 7.3 – Temps de roulage moyens sans incertitude

moins capacitif), mais la pénalité moyenne ne dépasse jamais le tiers de celle de Roissy.

La figure 7.3 donne les variations des temps de roulage résultants des simulations, comparés aux temps de roulage réels à Roissy (mesurés sur les traces radar de la même journée) :

- Les ordres de grandeur entre les temps simulés et les temps observés sont comparables, mais les pointes de trafic sont mieux absorbées dans les simulations : ceci peut s'expliquer par l'absence d'incertitude sur la vitesse de roulage dans les simulations, mais indique également que les trajectoires réelles ne sont généralement pas optimales. On peut remarquer que les temps de roulage simulés sont supérieurs à ceux observés dans les premières et les dernières heures de la journée à Roissy (avant 5h et après 22h) : ceci concerne un nombre très faible de mouvements et peut s'expliquer par les vitesses de roulage réelles, qui sont beaucoup plus élevées lorsque l'aéroport est presque vide.
- Les résultats confirment les légères améliorations apportées par le classement par heure piste à Roissy (qui aboutit souvent à une minute de roulage en moins).

7.4 Classement fixe avec incertitudes

7.4.1 Problèmes posés

La prise en compte des incertitudes sur la vitesse de roulage nécessite l'adaptation de la méthode définie dans la partie précédente : les mouvements ne peuvent plus être considérés que sur l'horizon de prédiction limité (au delà, les zones d'incertitude des avions deviennent beaucoup trop importantes). Ceci pose deux principaux problèmes :

- Dans chaque situation, l'état courant est le résultat des résolutions précédentes mais il peut nécessiter un classement différent, car l'effet de l'horizon peut positionner un avion devant d'autres qui étaient initialement prioritaires.
- Le graphe des possibilités pour un avion doit être redéfini, car les chemins ne seront plus systématiquement explorés jusqu'à la fin : il faut donc modifier la notion de nœud terminal et intégrer les contraintes liées à l'effet de l'horizon. Pour rendre la méthode de résolution plus opérationnelle, une limite doit également être ajoutée sur le nombre de positions d'attente imposables à un même mouvement.

7.4.2 Classement des avions

À chaque étape de la simulation, les positions initiales des avions contraignent les classements envisageables : le classement reste fixe, dans le sens où il est établi avant la résolution de la situation et n'est pas remis en cause par cette dernière, mais il pourra varier d'une situation à la suivante. Ces variations posent toutefois un dilemme :

- Plus le classement est adapté à la situation courante, plus la résolution sera efficace. La priorité absolue donnée aux arrivées dans la méthode précédente pourra ainsi être modérée : seules les arrivées non encore posées doivent réellement être prioritaires.
- Plus les classements successifs seront différents d'une étape de la simulation à la suivante, plus les solutions trouvées seront instables, avec de nombreux changements de chemin et de points d'attente. Le risque d'aboutir à un problème sans solution sera donc augmenté.

Pour ces raisons, deux méthodes de classement de stabilité différentes sont envisagées :

1. **Le classement « stable »** est calqué sur le classement par heure prévue à la piste, mis en place dans la partie précédente. Il ne sera remis en cause que pour les avions se suivant de près :

les arrivées restent totalement prioritaires sur les départs, mais si un avion a suit un avion b au début de l'horizon de prédiction (selon la définition donnée dans la partie 4.3.3), a se verra classé immédiatement après b .

2. **Le classement « instable »** correspond également au classement par heure prévue à la piste, mais les arrivées deviennent moins prioritaires que les départs une fois posées.

7.4.3 Optimisation de la trajectoire d'un avion

Adaptation du graphe des possibilités

Le graphe représentant les trajectoires possibles d'un avion est défini en fonction du nombre η de chemins qui restent envisageables pour l'avion et du nombre maximal Ω de positions d'attentes imposables sur l'horizon de prédiction :

- **Les nœuds du graphe** représentent des zones d'incertitude datées de l'avion, associées à un nombre de positions d'attentes passées. Ils sont définis par des quintuplets $(\nu, z, t, \omega, \lambda)$:
 - ν est toujours l'indice du chemin suivi par l'avion ;
 - z décrit la zone d'incertitude de l'avion dans ce chemin : elle peut être représentée par le couple de ses positions extrêmes (p_{\min}, p_{\max}) avec $p_{\min}^0 \leq p_{\min} \leq p_{\max} \leq p_{\max}^0 + H_p$ (si $z_0 = (p_{\min}^0, p_{\max}^0)$ désigne la zone d'incertitude initiale de l'avion).
 - $t \in \{0, \dots, H_p\}$ est l'heure relative à la situation courante, en pas de temps ;
 - $\omega \in \{0, \dots, \Omega\}$ est le nombre de positions d'attentes déjà adoptées par l'avion ;
 - $\lambda \in \{0, 1\}$ indique si l'avion avance ($\lambda = 1$) ou est en attente ($\lambda = 0$).
- **Un nœud initial** unique, caractérisé par $\nu = 0$ décrit la position de l'avion au début de l'horizon ou à l'heure t_a de début du mouvement si cette dernière est supérieure à l'heure courante. Ce nœud particulier permet de générer les η fils $\{(\nu, z_0, \max(0, t_a), 0, 1) \mid n \in \{1, \dots, \eta\}\}$ décrivant le choix du chemin, chacun d'eux pouvant être atteint avec un coût $\alpha(l(\nu) - l(1))$, en accord avec le critère global.
- **Un nœud est terminal** (ou solution) s'il représente une zone d'incertitude de l'avion qui respecte les contraintes de séparation et les contraintes liées à l'effet de l'horizon définies dans la partie 5.2.2, à la fin de son chemin ($p_{\min} = l(\nu)$) ou à la fin de l'horizon ($t = H_p$).
- **Les fils d'un nœud** $(\nu, z, t, \omega, \lambda)$ (ni initial, ni terminal) représentent les choix possibles à l'instant t . Le nœud n'a aucun fils s'il représente une position conflictuelle de l'avion, sinon il peut avoir un ou deux fils :
 - L'avion peut avancer : le nœud fils correspondant $(\nu, z^+, t + 1, \omega, 1)$ est atteint avec un coût nul (z^+ est calculée en fonction du modèle d'incertitude décrit dans la partie 4.2.4).
 - Soit l'avion peut attendre sur la position la plus avancée de sa zone d'incertitude, ce qui porterait son nombre de positions d'attente à $\omega + \lambda$: si $\omega + \lambda \leq \Omega$, le nœud fils $(\nu, z^-, t + 1, \omega + \lambda, 0)$ représentant cette attente peut être atteint avec un coût unitaire (z^- est alors la nouvelle zone d'incertitude de l'avion, qui évolue même lorsque ce dernier est en attente).

Adaptation de l'algorithme A*

Pour trouver la meilleure solution pour chaque avion (lorsqu'une telle solution existe), l'algorithme A* doit être muni d'une heuristique minorante. En accord avec les coûts nuls associés à la progression de l'avion dans un de ses chemins, l'heuristique considérée est nulle pour tous les nœuds du graphe, ce qui réduit l'algorithme à un *Dijkstra* avec condition d'arrêt sur les nœuds terminaux.

Les nœuds développés ne peuvent pas être caractérisés aussi directement que dans la méthode précédente (car le nombre de positions d'attente est limité et l'évolution de la zone d'incertitude de l'avion est plus complexe). Cependant, beaucoup de branches de l'arbre de recherche peuvent être élaguées :

- Le nombre de positions d'attente étant limité, il n'est jamais intéressant d'atteindre une même position (sur un même chemin) à une même heure avec un nombre de positions d'attente plus élevé. Si $\omega_{\min}(\nu, z, t)$ désigne le nombre minimal de positions d'attente dans les nœuds développés ayant atteint la zone d'incertitude z à l'heure t :

$$\forall u = (\nu, z, t, \omega, \lambda) \text{ nœud non développé du graphe,}$$

$$\text{Si } \omega \geq \omega_{\min}(z, t), \text{ alors } u \text{ n'a pas besoin d'être développé.}$$

- De plus, les positions conflictuelles explorées à un instant t peuvent être mémorisées, pour éviter d'effectuer plusieurs fois la même détection de conflits (qui reste l'opération la plus pénalisante lors de l'exploration d'un nœud) : lorsqu'une zone d'incertitude z_i s'avère conflictuelle à l'instant t_i , tous les nœuds $(\nu, z_i, t_i, \omega, \lambda)$ seront considérés explorés, par l'annulation des valeurs $\omega_{\min}(\nu, z_i, t_i)$.

7.4.4 Application

Les deux méthodes de classement envisagées sont appliquées au trafic de Roissy et d'Orly, à partir des mêmes plans de vol que précédemment, avec les mêmes paramètres généraux, mais avec le modèle de simulation décrit dans la partie 4.4.1, définit comme suit :

- Horizon de prédiction : $H_p = 10$ minutes ;
- Pas de résolution : $\Delta = 2$ minutes ;
- Incertitude sur la vitesse : $\delta_v = 20\%$
- Vitesse maximale : $V_{\max} = 10$ m/s ;
- Nombre maximum de positions d'attente : $\Omega = 2$.

À chaque étape de la simulation, les *clusters* de mouvements actifs sont formés en fonction des paires conflictuelles. La méthode de résolution itérative décrite dans la partie 5.4.3 est appliquée (et chaque cluster est résolu séparément par classement de ses propres avions).

L'horizon de prédiction choisi est réduit (inférieur aux temps de roulage moyens) mais correspond déjà à une taille conséquente par rapport aux incertitudes considérées. La longueur maximale des zones d'incertitude des avions à la fin de l'horizon est en effet donnée par :

$$l_{\max} = \delta_v(\Delta + H_p)V_{\max} = 1440 \text{ mètres}$$

Le tableau suivant présente les résultats globaux des 4 simulations avec un processeur à 1.6 GHz :

Aéroport	Classement	Temps de calcul	Pénalité moyenne (sec.)	Cas non résolus
Roissy	stable	40 min.	134	3
	instable	40 min.	112	17
Orly	stable	25 min.	62	7
	instable	25 min.	61	10

Les temps de calcul restent accessibles car le modèle tabulé décompose la simulation en situations successives plus simples, dont le temps de résolution moyen est de l'ordre de quelques secondes.

De nouveaux cas d'échec de la résolution sont par contre identifiés :

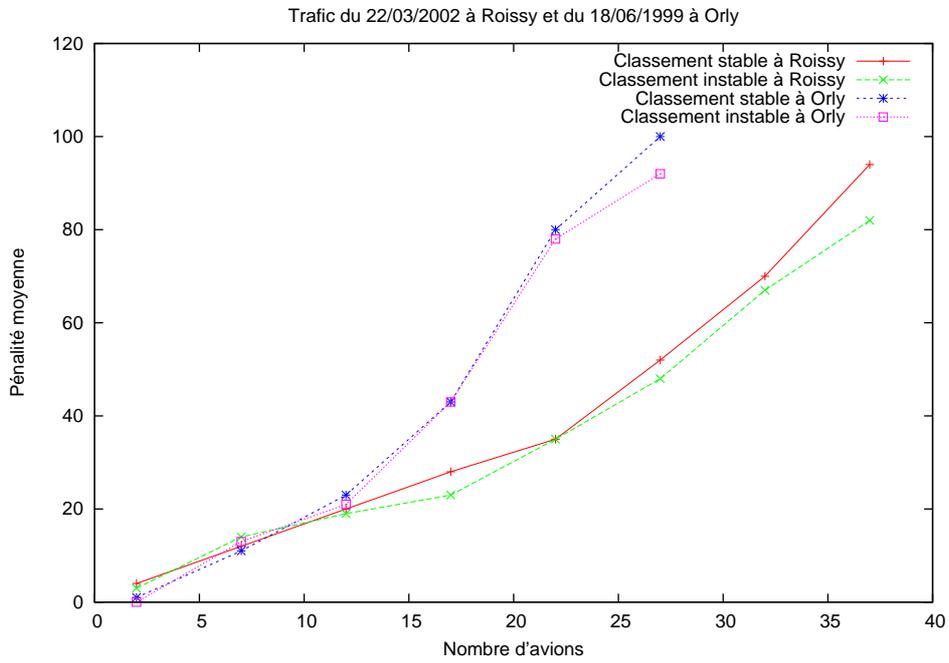


FIG. 7.4 – Pénalité en fonction du nombre d'avions avec incertitudes

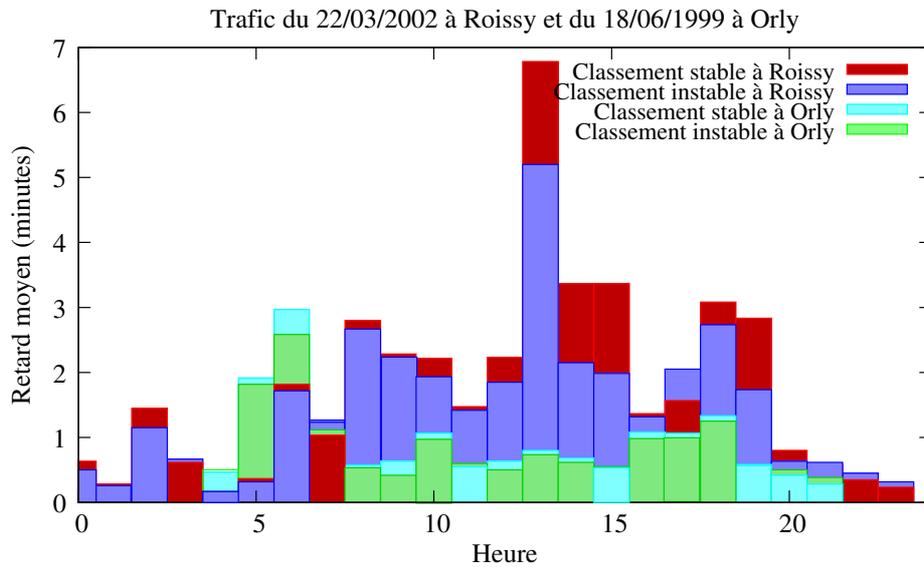


FIG. 7.5 – Retard moyen avec incertitudes

- Avec le classement « stable », les cas non résolus observés dans les simulations sans incertitude persistent : les arrivées sont toujours prioritaires et n’attendent pas nécessairement la libération de leur parking. L’étude précise des autres cas d’échec (à Roissy comme à Orly) révèle une cause commune, liée aux incertitudes sur la vitesse et à l’effet de l’horizon : les zones d’incertitude des avions se réduisent d’une résolution à la suivante et causent souvent des réductions d’attente ou des changements de chemin. Ces trajectoires, non prévues auparavant, rendent parfois insoluble le problème d’autres avions (moins prioritaires).
- Avec le classement « instable », les arrivées perdent leur priorité une fois posées et sont donc forcées d’attendre la libération de leur parking : ces problèmes sont donc résolus à Roissy comme à Orly. Par contre, les nouveaux cas d’échec sont plus nombreux et peuvent avoir des causes différentes : certains sont dus aux évolutions des zones d’incertitude (comme avec le classement stable), mais la plupart sont liés aux changements de classement des arrivées, qui doivent soudainement éviter des départs, alors que ces derniers n’étaient pas prioritaires dans la résolution précédente.
- L’effet des incertitudes et de l’horizon de prédiction limité peut se mesurer par la pénalité moyenne, supérieure à celle obtenue dans la partie précédente, alors que les cas d’échec, plus nombreux, ne sont pas comptabilisés.

La figure 7.4 donne la pénalité générée dans chaque situation en fonction du nombre total d’avions qui la composent :

- Les ordres de grandeur du nombre d’avions et des pénalités moyennes sont très différents des précédents car ils sont maintenant relatifs à l’horizon de prédiction limité.
- À Roissy, le classement instable s’avère généralement moins pénalisant et ceci est confirmé par les retards moyens donnés sur la figure 7.5. Ces résultats montrent que la méthode est améliorée lorsque le classement est souvent remis en cause, mais ceci doit être relativisé par le nombre de cas non résolus : il est difficile de trouver *a priori* un classement qui permette de résoudre toutes les situations.
- À Orly, les mêmes effets peuvent être constatés, mais avec des retards finalement similaires pour les deux classements, ce qui confirme ce qui a été mesuré sans incertitude : la simplicité des voies de circulation ne laisse pas beaucoup de possibilités pour la répartition du trafic.

7.5 Méthode avec variations du classement

7.5.1 Objectifs

Les résultats précédents montrent que la méthode 1 contre n est relativement peu coûteuse en temps de calcul, mais très sensible au classement sur lequel elle est appliquée. Dans cette partie, on s’intéresse donc à l’élaboration d’une méthode itérative permettant d’explorer plusieurs classements des avions. L’objectif principal est de trouver un classement compatible avec chaque situation, pour éviter autant que possible les échecs de la méthode, mais l’exploration de différents classements doit également aboutir à de meilleures solutions : les séquences de piste doivent notamment être améliorées par la possibilité d’insérer certains départs en décalant légèrement les arrivées. Pour cela, les atterrissages prévus ne doivent plus être systématiquement prioritaires.

- Dans un premier temps, l’algorithme A* permettant l’optimisation de la trajectoire d’un avion est remplacé par un algorithme d’exploration par *Branch & Bound* [Papadimitriou 82] qui s’avère plus rapide sur le problème (ceci permet de maximiser le nombre de classements envisagés).

- Dans un deuxième temps, un ordre d’exploration des classements est défini et un nouvel algorithme de *Branch & Bound* (utilisant le précédent) est mis en place pour rechercher le meilleur d’entre eux.

7.5.2 Résolution par *Branch & Bound*

Principes

L’étude plus précise du problème pour un avion fait apparaître que de nombreuses solutions sont équivalentes : sur chaque chemin, toutes les positions d’attentes non conflictuelles peuvent être utilisées avec la même durée, pour former autant de solutions différentes de même coût.

L’algorithme A^* développe systématiquement toutes ces possibilités en parallèle et par coût estimé croissant, alors qu’il peut être plus judicieux de n’en développer qu’une, *en profondeur d’abord* : si une solution admissible est trouvée, elle n’est pas nécessairement optimale, mais elle apporte des informations supplémentaires sur le problème, ce qui permet par la suite d’élaguer certaines branches de l’arbre de recherche. Dans ce cadre, l’ordre d’exploration du graphe peut être fixé d’avance, ce qui évite la sélection systématique d’un nœud généré de coût estimé minimal, qui est l’une des opérations les plus pénalisantes de l’ A^* .

Dans le cas d’un problème de minimisation, la connaissance d’une solution fournit notamment une borne supérieure de la pénalité minimale : si les premières branches explorées sont pertinentes (dans le sens où elles aboutissent à des solutions proches de l’optimum), la borne résultante est faible et le parcours de l’arbre de recherche en est fortement réduit. Une stratégie d’exploration de l’arbre doit donc sélectionner les premières branches les plus prometteuses à développer.

Cette forme de parcours correspond à un algorithme de type *Branch & Bound* (annexe D.5) :

- **La borne** correspond au coût de la meilleure solution trouvée (sa valeur est initialement $+\infty$ pour un problème de minimisation).
- **L’heuristique** évalue le coût du meilleur chemin permettant de joindre chaque nœud à un nœud terminal.
- Lorsqu’un nœud est exploré :
 - **Son coût, augmenté de l’heuristique, est comparé à la borne.** S’il lui est supérieur, l’exploration est interrompue : elle reprend plus haut dans l’arbre, vers d’autres branches non explorées (opération de *backtrack*).
 - **S’il est terminal**, la borne est mise-à-jour avec le coût de la solution trouvée et l’exploration des branches non encore explorées continue avec cette nouvelle borne.
 - **Dans les autres cas, la stratégie d’exploration** décide l’ordre dans lequel chaque nœuds fils doit être parcouru. Dans certaines implémentations, certains nœuds fils peuvent être explorés *en largeur*, mais ceci ne sera pas utile pour la méthode 1 contre n , car chaque nœud du graphe ne possède que deux fils.

Comme avec l’algorithme A^* , la solution trouvée sera optimale si l’heuristique est minorante.

Formulation de l’algorithme

L’algorithme de *Branch & Bound* pour la méthode 1 contre n est implémenté de la façon suivante :

- L’heuristique est nulle pour tous les nœuds (comme pour l’ A^*).
- La stratégie d’exploration est *meilleur en premier* : les arcs de moindre coût sont explorés en premier et *en profondeur d’abord*. Rapporté au problème, cette stratégie consiste à rechercher d’abord une solution sur le plus court chemin de l’avion, en explorant en premier les nœuds du graphe correspondants à l’avancée de l’avion.

La stratégie d'exploration décompose ainsi le problème pour un avion en autant de sous-problèmes qu'il reste de chemins envisageables pour cet avion : chaque sous-problème consiste à chercher une solution optimale sur un chemin particulier, de retard inférieur à une borne fixée.

Comme pour l'algorithme A*, il n'est jamais intéressant d'atteindre une même position sur le chemin avec un nombre plus élevé de positions d'attente. Si une position est caractérisée par un retard d à un instant t et si $w_{\min}(t, d)$ désigne le nombre de positions d'attente des nœuds ayant atteint cette position :

$$\forall u = (z, t, \omega, \lambda) \text{ nœud du graphe de retard } d, \\ \text{Si } \omega \geq w_{\min}(t, d), \text{ alors } u \text{ n'a pas besoin d'être exploré.}$$

De même que précédemment, les nœuds (z, t, ω, λ) représentant une position conflictuelle peuvent être mémorisés pour éviter d'effectuer plusieurs fois la même détection de conflit, en fixant $w_{\min}(t, d)$ à 0.

<pre> BB_chemin((a_i)_{1 \leq i \leq n}, a, \nu, P) : soit D : P - \alpha l(\nu) \forall (t, d) \in \{0, \dots, H_p\}^2, soit \omega_{\min}(t, d) : \Omega + 1 soit \gamma : \{\} Explorer(z, t, \omega, \lambda, d) : si d < D et \omega < \omega_{\min}(t, d) alors \omega_{\min}(t, d) \leftarrow \omega si z est conflictuelle à l'instant t alors \omega_{\min}(t, d) \leftarrow 0 sinon si t = H_p ou a est terminé à t alors \gamma \leftarrow la trajectoire de a D \leftarrow d sinon Explorer(z^+, t + 1, \omega, 1, d) si a peut attendre sur z alors Explorer(z^-, t + 1, \omega + \lambda, 0, d + 1) Explorer(z_0, \max(0, t_a), 0, 1, 0) retourner (\gamma, D + \alpha l(\nu)) </pre>	<p>(Exploration du $\nu^{\text{ème}}$ chemin de a avec pénalité maximale P) (Borne supérieure du retard) (Caractérisation des nœuds explorés) (Meilleure trajectoire trouvée) (Si le nœud est utile à explorer) (Mémorisation des nœuds explorés) (Mémorisation des conflits) (Une solution de retard d est trouvée) (Mise à jour de la borne) (Cas où l'avion avance à t) (Cas où l'avion attend à t)</p>
<pre> BB_1n((a_i)_{1 \leq i \leq n}, a) : soit \nu : 1 soit (\gamma, P) : BB_chemin((a_i)_{1 \leq i \leq n}, a, 1, +\infty) pour chaque \nu' \in \{2, \dots, \eta\} soit (\gamma', P') : BB_chemin((a_i)_{1 \leq i \leq n}, a, \nu', P) si \gamma' \neq \{\} alors (\gamma \leftarrow \gamma'; P \leftarrow P'; \nu \leftarrow \nu') retourner (\nu, \gamma, P) </pre>	<p>(Exploration de tous les chemins de a) (Chemin de la meilleure solution)</p>

Algorithme III.6 – *Branch & Bound* pour la résolution du problème 1 contre n

Dans ces conditions, l'algorithme III.6 résout le problème 1 contre n pour un avion :

- Un premier *Branch & Bound* (intitulé BB_chemin) cherche la meilleure solution avec borne sur un chemin. La recherche proprement dite s'effectue par une fonction récursive Explorer, qui retourne la trajectoire optimale pour l'avion et la nouvelle pénalité maximale correspondante. Lorsqu'aucune solution ne peut être trouvée, la trajectoire retournée est « vide » et la pénalité maximale est inchangée.
- L'optimisation de la trajectoire d'un avion (intitulée BB_1n) s'effectue par itérations de l'algorithme de *Branch & Bound* sur chaque chemin, en conservant la borne correspondante à la

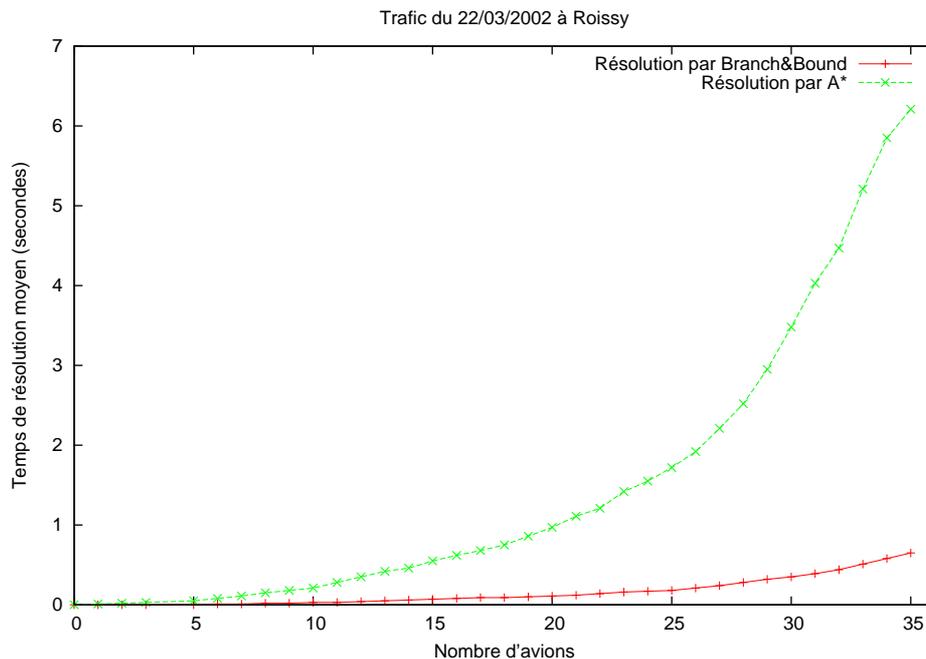


FIG. 7.6 – Temps de résolution moyens du problème 1 contre n

meilleure solution trouvée. Si aucune solution ne peut être trouvée sur aucun des chemins, la trajectoire « vide » est retournée, avec une pénalité infinie.

Résultats

Pour comparer la résolution par *Branch & Bound* à la résolution par *A**, deux simulations sont effectuées à Roissy, sur la même journée de trafic, avec le classement stable et les mêmes paramètres que précédemment, mais sans former les *clusters* d'avions en conflit (l'objectif est de comparer l'efficacité des algorithmes sur un grand nombre d'avions). Les temps de calculs de chaque méthode de résolution sont mesurés et corrélés au nombre d'avions impliqués dans la situation (la pénalité des solutions trouvées par les deux méthodes est naturellement identique).

La figure 7.6 montre que l'algorithme de *Branch & Bound* est environ dix fois plus rapide, même lorsque le nombre d'avions est élevé : cela signifie que l'ordre de parcours de graphe prédéfini par la stratégie d'exploration du *Branch & Bound* est particulièrement adapté au problème (les premières branches explorées contiennent des solutions proches de l'optimum). Cet algorithme permettra donc d'explorer en moyenne dix fois plus de classements pour un temps de résolution équivalent.

7.5.3 Recherche d'un classement optimal

Principe

Dans cette partie, l'algorithme de résolution du problème 1 contre n par *Branch & Bound* est utilisé itérativement pour chercher un classement minimisant la pénalité cumulée de tous les avions.

Cette méthode peut nécessiter l'étude des $N!$ permutations possibles des N avions d'un cluster, ce qui peut rapidement devenir trop coûteux en temps de calcul : l'exploration sera donc interrompue

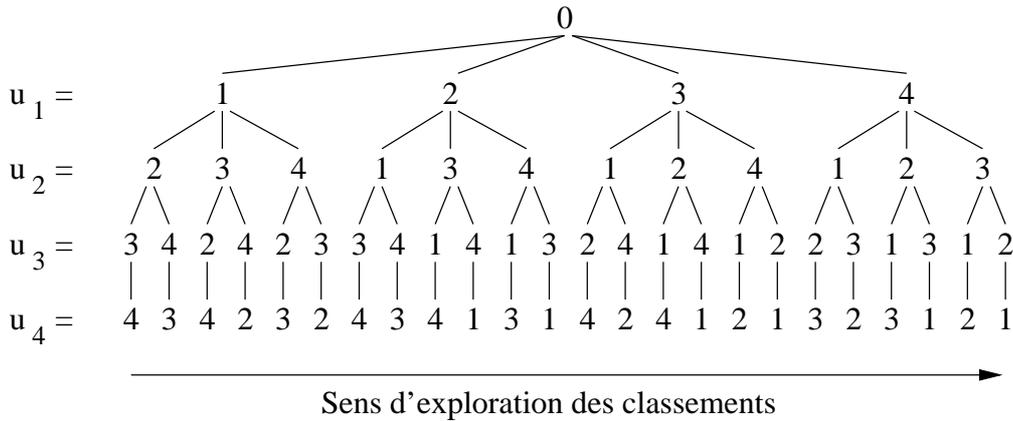


FIG. 7.7 – Exploration lexicographique des classements

au bout d’un temps d’exécution maximal fixé et l’objectif est d’étudier l’efficacité de cette méthode, en fonction du nombre d’avions composant chaque cluster.

Implémentation

Chaque classement est évalué par applications successives des résolutions du problème 1 contre n de chaque avion, dans l’ordre donné : lorsqu’un classement aboutit à une solution, la pénalité cumulée fournit un majorant de la pénalité minimale, ce qui permet ensuite de limiter la recherche aux classements moins coûteux. L’exploration des classements peut donc également être implémentée par un algorithme de type *Branch & Bound*, dans lequel la borne est la pénalité de la meilleure solution trouvée. Si les premiers classements explorés fournissent des solutions proches de l’optimum, la recherche sera fortement réduite : le classement « instable » définit dans la partie 7.4.2 pourra par exemple servir de classement initial, puis des transpositions élémentaires pourront être opérées pour générer tous les autres classements.

Dans ce cadre, un classement des N avions $(a_i)_{1 \leq i \leq N}$ est représenté par une instanciation de N variables entières $(u_i)_{1 \leq i \leq N}$, telle que :

$$\begin{cases} \forall i \in \{1, \dots, N\}, u_i \in \{1, \dots, N\} \\ \forall (i, j) \in \{1, \dots, N\}^2, i \neq j \implies u_i \neq u_j \end{cases}$$

L’arbre de recherche sur lequel sera appliqué le *Branch & Bound* doit explorer les différentes instanciations possibles des variables $(u_i)_{1 \leq i \leq N}$:

- **Chaque nœud de l’arbre** représente une instanciation des j premières variables $(u_i)_{1 \leq i \leq j}$. Le cas $j = 0$ correspond au nœud initial, pour lequel aucune variable n’est instanciée, tandis que les nœuds terminaux sont caractérisés par $j = N$.
- **Les fils d’un nœud** dans cet arbre (lorsque $j < N$) sont toutes les instanciations possibles de la variable suivante u_{j+1} , telles que le sous-problème des $j + 1$ avions classés $(a_{u_i})_{1 \leq i \leq j+1}$ admette une solution. Chaque fils représente ainsi le choix du $(j+1)^{ème}$ avion dans le classement : le coût de l’arc correspondant est la pénalité attribuée à cet avion dans la résolution du problème 1 contre n .

Le *Branch & Bound* se définit alors comme suit :

- **La stratégie d’exploration** de l’arbre consiste à étudier en premier le classement « instable », puis ses classements voisins. Pour cela, on suppose que les N avions $(a_i)_{1 \leq i \leq N}$ sont donnés

dans l'ordre correspondant au classement « instable », et les classements sont explorés dans l'ordre lexicographique, comme cela est illustré figure 7.7 : de cette façon, les premiers avions dans le classement sont ceux qui changent le moins souvent de priorité et leur trajectoire optimale peut être conservée pendant toute l'exploration des classements des derniers avions.

- **Une heuristique** minorante peut être définie : chaque avion non encore classé (c'est-à-dire non représenté dans les variables instanciées) sera au moins autant pénalisé que s'il était classé en $j + 1^{\text{ème}}$ position. La somme de ces pénalités minimales forme donc l'heuristique. De plus, si un des avions non encore classé n'a plus de solution, il ne pourra plus en avoir par la suite et l'exploration des nœuds fils peut être abandonnée.
- **Un temps maximal d'exécution** δ_s est fixé : passé cette durée, la recherche est interrompue dès qu'une solution admissible est trouvée.

```

Meilleur_Classement  $((a_i)_{1 \leq i \leq N})$ 
  soit  $t_0$  : l'heure courante
  soit  $P_{\max} : +\infty$  (Borne supérieure de la pénalité)
   $\forall i \in \{1, \dots, N\}$  soit  $u_i : i$  et  $v_i : i$  (Classement exploré et classement retenu)
  Explorer  $(j, p)$  :
    si  $P_{\max} = +\infty$  ou si l'heure courante est inférieure à  $t_0 + \delta_s$  alors
       $\forall k \in \{j, \dots, N\}$ , soit  $(\nu_k, \gamma_k, p_k) : \text{BB-1n}((a_{u_i})_{1 \leq i < j}, a_{u_k})$ 
      soit  $f : p + \sum_{k=j}^N p_k$  (Coût minimum des classements à explorer)
      si  $f < P_{\max}$  alors (La pénalité maximale n'est pas atteinte)
        si  $j = N$  alors (Un classement de pénalité  $f$  est trouvé)
           $P_{\max} \leftarrow f$  (Révision de la borne)
           $\forall i \in \{1, \dots, N\}, v_i \leftarrow u_i$  (Mémorisation du classement)
        sinon
          pour chaque  $k$  de  $j$  à  $N$  (Exploration de tous les classements des  $(a_{u_k})_{j \leq k \leq N}$ )
            Affecter la trajectoire  $(\nu_k, \gamma_k)$  à  $a_{u_k}$ 
            si  $k > j$  alors Échanger  $(u_k, u_j)$ 
            Explorer  $(j + 1, p + p_k)$ 
            si  $k > j$  alors Échanger  $(u_k, u_j)$ 
    Explorer  $(1, 0)$ 
    si  $P_{\max} = +\infty$  alors lever (Sans_solution) (Aucun classement admissible n'a pu être trouvé)
    sinon retourner  $(v_i)_{1 \leq i \leq N}$ 

```

Algorithme III.7 – *Branch & Bound* pour la recherche du meilleur classement des avions

L'algorithme III.7 implémente cette forme de recherche du classement optimal par l'intermédiaire d'une fonction récursive Explorer : on suppose la suite d'avion $(a_i)_{1 \leq i \leq N}$ donnée dans l'ordre du classement « instable » et chaque variable u_i vaut initialement i . Les classements suivants sont obtenus par transposition élémentaire de deux variables (u_j, u_k) , avec $1 \leq j < k \leq N$.

Résultats

Pour évaluer l'efficacité de la résolution avec recherche d'un classement optimal, deux nouvelles simulations sont effectuées, l'une à Roissy, l'autre à Orly, avec les mêmes journées de trafic et les mêmes paramètres que précédemment. La durée maximale de résolution optimale d'un cluster est fixée à $\delta_s = 1$ minute.

Le tableau suivant juxtapose les résultats globaux des nouvelles simulations à ceux des simulations précédentes (dans lesquelles l'algorithme A* est remplacé par le *Branch & Bound*) :

Aéroport	Classement	Temps de calcul	Pénalité moyenne (sec.)	Cas non résolus
Roissy	fixe « instable »	22 min.	112	17
	optimal	6h15 min.	111	0
Orly	fixe « instable »	11 min.	61	10
	optimal	1h45 min.	41	0

Deux points importants peuvent être constatés :

- Il n'y a plus de cas d'échec de la résolution : un classement admissible est trouvé dans toutes les situations.
- La pénalité moyenne est significativement diminuée à Orly. Elle est équivalente à Roissy, mais ceci s'explique par l'absence de cas d'échec, qui implique que tous les mouvements sont comptabilisés. L'étude des trajectoires résultantes fait apparaître deux raisons principales de l'amélioration des résolutions :
 - Dans les aires de parkings, la pénalité est diminuée dans les situations où une arrivée doit rejoindre un parking voisin de celui d'un départ : si l'arrivée est prioritaire sur le départ (classement optimal), elle ne lui cause que très peu de retard ; dans la négative (classement « instable »), l'arrivée doit attendre pendant toute la procédure de push-back, ce qui est beaucoup plus pénalisant.
 - Les séquences de piste sont optimisées, notamment lorsque des départs peuvent être insérés dans la séquence en retardant les atterrissages : à chaque fois qu'une telle insertion est possible, chaque décollage de la file d'attente gagne au moins une minute (le temps de séparation élémentaire entre chaque mouvement).

Le classement des avions est donc une méthode efficace pour résoudre le problème lorsqu'il est régulièrement corrigé pour être adapté à chaque situation de trafic.

La figure 7.8 donne les temps de résolution des clusters en fonction de leur taille, avec un micro-processeur à 1.6 GHz. :

- Le temps *de première résolution* correspond au temps maximal au bout duquel l'algorithme a trouvé une première solution admissible : ce temps est inférieur à la seconde pour tous les clusters.
- Le temps *de résolution finale* correspond au temps maximal de la recherche d'un classement optimal (lorsqu'il est inférieur à $\delta_s = 1$ minute, l'arbre des classements a été exploré entièrement).

Ces résultats montrent qu'un classement optimal peut toujours être trouvé en moins d'une minute pour les clusters de moins de 9 avions et en moins de 15 secondes pour les clusters de moins de 7 avions : la durée relativement élevée des simulations avec recherche du classement optimal peut être diminuée en limitant davantage le temps de recherche maximal, bien que cela se fasse au détriment de la qualité des solutions trouvées.

La figure 7.9 donne la pénalité moyenne en fonction du nombre d'avions impliqués dans chaque situation, avec et sans recherche de classement optimal. La figure 7.10 fournit les temps de roulage correspondants. Pour les deux aéroports, l'amélioration des résolutions est significative dans les situations modérément chargées. Par contre, lorsque le nombre d'avions devient plus grand, les pénalités résultantes sont comparables. L'ordre finit même par s'inverser pour le nombre maximal d'avions (mais ceci est dû à l'absence de cas d'échec).

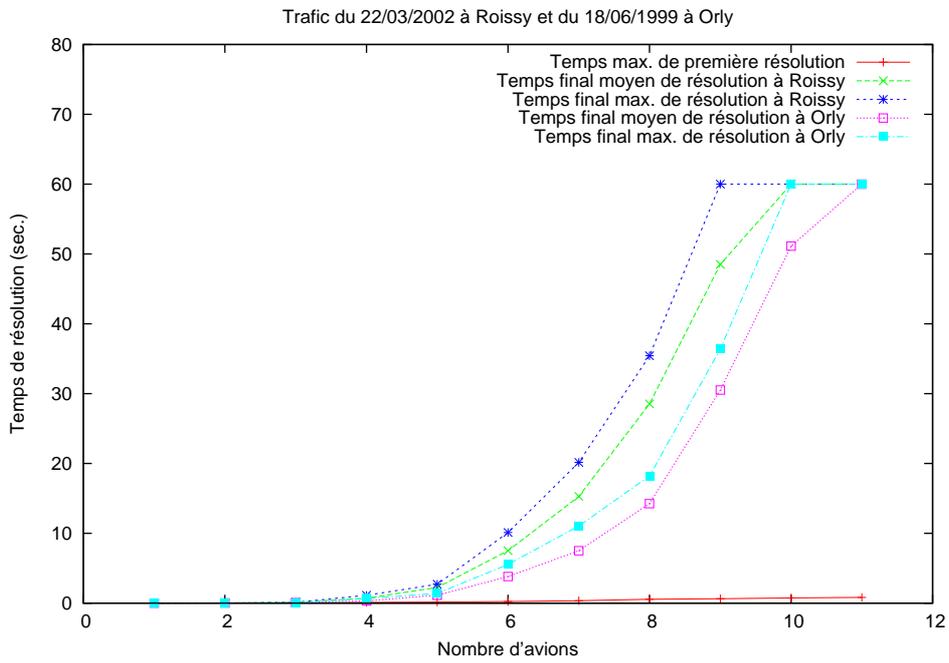


FIG. 7.8 – Temps de résolution par tailles de clusters

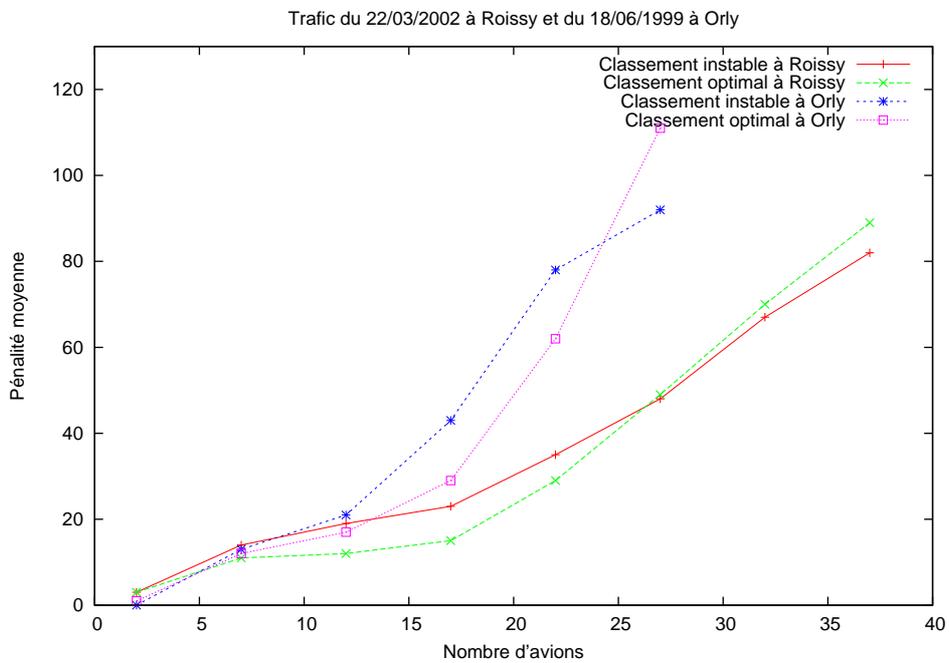


FIG. 7.9 – Pénalité en fonction du nombre d'avions avec variations du classement

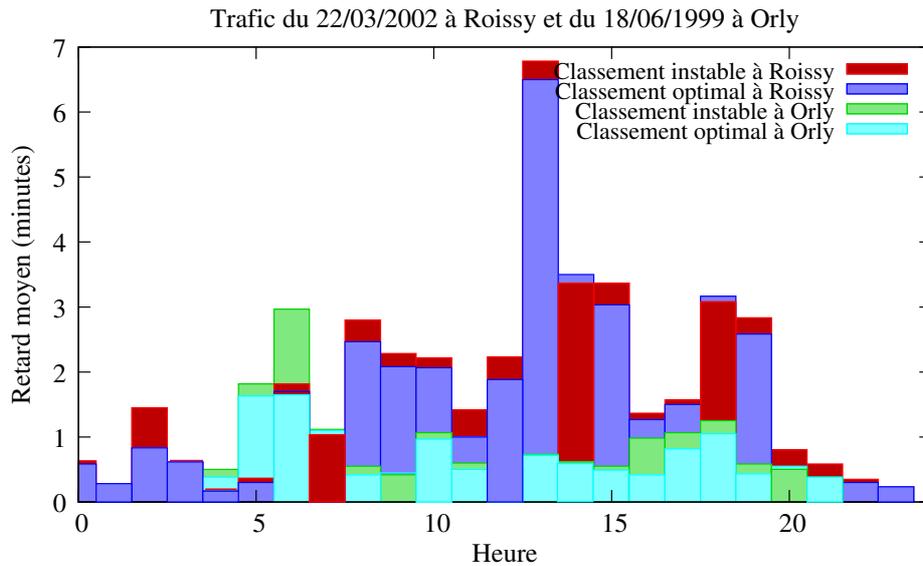


FIG. 7.10 – Retard moyen avec variations du classement

7.6 Conclusion

La méthode 1 contre n résoud les situations de trafic en classant les avions par ordre de priorité. Elle travaille donc sur un problème simplifié mais adapté aux méthodes de résolution déterministes par parcours de graphe. Elle revêt deux formes d'application intéressantes :

- Elle peut être appliquée au problème sans incertitude sur la vitesse de déplacement, avec un horizon de prédiction infini. Dans ce cas, elle permet une analyse très rapide des conditions d'écoulement du trafic au sol et des retards induits, en affectant une trajectoire admissible à chaque mouvement d'une journée entière de trafic.
- Elle peut également s'appliquer aux situations de trafic issues du modèle de simulation tabulé, avec horizon de prédiction limité et prise en compte des incertitudes sur la vitesse. Dans ce cas, plusieurs classements doivent être envisagés pour pouvoir résoudre efficacement toutes les situations de trafic. Le nombre de classements explorés est maximisé en remplaçant l'algorithme A^* initial par un *Branch & Bound*, qui s'avère en moyenne dix fois plus rapide. Les retards résultants se rapprochent de ceux obtenus sans incertitude, ce qui signifie que les solutions trouvées sont mieux optimisées : le classement des avions, lorsqu'il est adapté et corrigé en fonction de chaque situation, ne semble pas extrêmement pénalisant pour le trafic au sol. Ceci demande naturellement à être confirmé par comparaison avec d'autres méthodes de résolution.

Deux limites non négligeables de la méthode doivent cependant être soulignées :

- La définition du graphe exploré dépend fortement du critère global à optimiser : la méthode n'est donc pas généralisable à d'autres critères plus complexes.
- La nécessité de définir un ordre total sur les avions peut entraîner l'absence de solution au problème. Ceci peut notamment être constaté dans les situations particulières de face-à-face, où les deux avions doivent effectuer un détour pour s'éviter mutuellement.

Chapitre 8

Résolution par algorithmes génétiques

8.1 Introduction

La méthode déterministe développée au chapitre précédent permet de résoudre toutes les situations de trafic étudiées mais au prix d'une réduction considérable de l'espace de recherche. Plusieurs raisons incitent pourtant à étudier le problème dans sa forme la plus générale, telle qu'elle a été formulée au chapitre 5 :

- L'optimisation des situations peut en être significativement améliorée : les solutions trouvées correspondront à un optimum global, prenant en compte simultanément les intentions et les contraintes de tous les avions impliqués dans la situation.
- La méthode de résolution peut être rendue plus indépendante du critère à optimiser, qui est nécessairement évolutif (fonction de l'aéroport et de ses procédures d'exploitation spécifiques par exemple).

La complexité élevée des contraintes liées au problème de résolution de conflits le rend difficilement accessible à d'autres méthodes déterministes classiques, telles que la programmation par contraintes ou les méthodes d'intervalles : l'emploi de méthodes stochastiques telles que les algorithmes génétiques s'est en revanche déjà montrée efficace [Granger 02, Médioni 98] sur le même type de problème, relatif à l'évitement aérien. Dans ce chapitre, les mêmes concepts sont développés pour le trafic au sol, tel qu'il a été modélisé au chapitre 4 : les deux premières parties sont consacrées à la présentation générale des algorithmes génétiques et à leurs améliorations classiques. La dernière partie en fournit une implémentation adaptée aux situations de trafic au sol et une application à l'environnement de simulation.

8.2 Principes généraux

8.2.1 Présentation

Les algorithmes génétiques s'inspirent des lois de la sélection naturelle et de l'évolution génétique pour résoudre des problèmes d'optimisation fortement combinatoires : les points de l'espace de recherche sont considérés comme des individus d'une population et la fonction à optimiser représente leur adaptation. Lorsque la population se reproduit, seuls les individus les mieux adaptés subsistent : l'héritage génétique permet aux nouvelles générations d'être de plus en plus adaptées aux conditions de sélection. Au bout d'un certain nombre de générations, la population doit converger vers les régions où se trouvent les optima recherchés et ce, quelle que soit la fonction d'adaptation : le principal avan-

tage des algorithmes génétiques est qu'ils ne font aucune hypothèse particulière sur la fonction à optimiser.

Les origines de ces algorithmes remontent aux années 60, avec les travaux de John Holland sur les systèmes adaptatifs [Holland 62]. L'ouvrage de référence de David E. Goldberg [Goldberg 89] a fortement participé à leur essor : ils sont désormais très populaires et leurs domaines d'applications sont de plus en plus variés. Ils forment aujourd'hui une branche particulière des *Algorithmes Évolutionnaires*, qui regroupent les *Stratégies d'Évolution* [Rechenberg 65, Beyer 01], la *Programmation Évolutionnaire* [Fogel 66a] et les *Algorithmes Génétiques*, bien que les spécificités de chaque catégorie ne soient pas toujours explicites et relèvent le plus souvent de l'historique ou des problèmes étudiés.

D'une manière générale, les algorithmes génétiques sont utilisés pour rechercher les optima d'un critère d'optimisation, défini sur un espace de recherche \mathcal{E} . Leur mise en œuvre nécessite :

- **La définition d'une fonction d'adaptation** f à maximiser, définie sur l'espace de recherche \mathcal{E} et généralement à valeurs dans $[0; 1]$ ou \mathbb{R}^+ . Cette fonction peut également être appelée la *fonction d'évaluation* ou encore *fitness*. Elle se définit en fonction du critère du problème : les extrema recherchés du critère doivent correspondre aux maxima de f .
- **Un codage des données**, associant à chaque point de l'espace de recherche une structure de donnée particulière, appelée *génotype* ou *chromosome*, qui caractérisera chaque *individu* de la population. Inversement, on doit pouvoir associer à tout individu un point de \mathcal{E} : ce point, qui peut parfois correspondre à plusieurs individus différents, représente les *caractéristiques apparentes* de ces individus et sera ainsi appelé leur *phénotype*.
- **Une population initiale**, définie comme un ensemble d'individus, qui sera souche de toutes les futures générations.
- **Des opérateurs d'évolution** de la population, permettant l'exploration de l'espace de recherche \mathcal{E} .
- **Un processus de sélection** des individus les mieux adaptés, qui sera appliqué sur chaque nouvelle génération d'individus.

Description algorithmique

La figure 8.1 illustre les principales étapes d'un algorithme génétique :

1. La population initiale est créée, généralement de manière aléatoire.
2. Chaque itération correspond à une nouvelle génération de la population. Pour ce faire :
 - (a) **L'adaptation (ou fitness)** de chaque individu est calculée : ceci nécessite le *décodage* du génotype de l'individu, qui donne son *phénotype*.
 - (b) **La reproduction ou sélection** consiste à faire proliférer les meilleurs individus de la population, au détriment des individus les moins adaptés. Elle aboutit à une nouvelle population, uniquement formée d'individus sélectionnés dans la population courante.
 - (c) **La diversification** de la population remplace certains individus en altérant leur génotype. Cette étape est essentielle pour assurer l'exploration de l'espace de recherche. Les individus sélectionnés dans la population sont appelés les *parents* et les nouveaux individus les *enfants*. Deux types d'opérateurs d'évolution sont distingués : les mutations et les croisements, comme cela sera détaillé plus loin.
3. **Un critère d'arrêt** détermine la fin de l'algorithme génétique. Il peut être défini en fonction de nombreux indicateurs, comme le nombre de générations effectuées, l'adaptation du meilleur individu, ou encore l'évolutivité de la population.

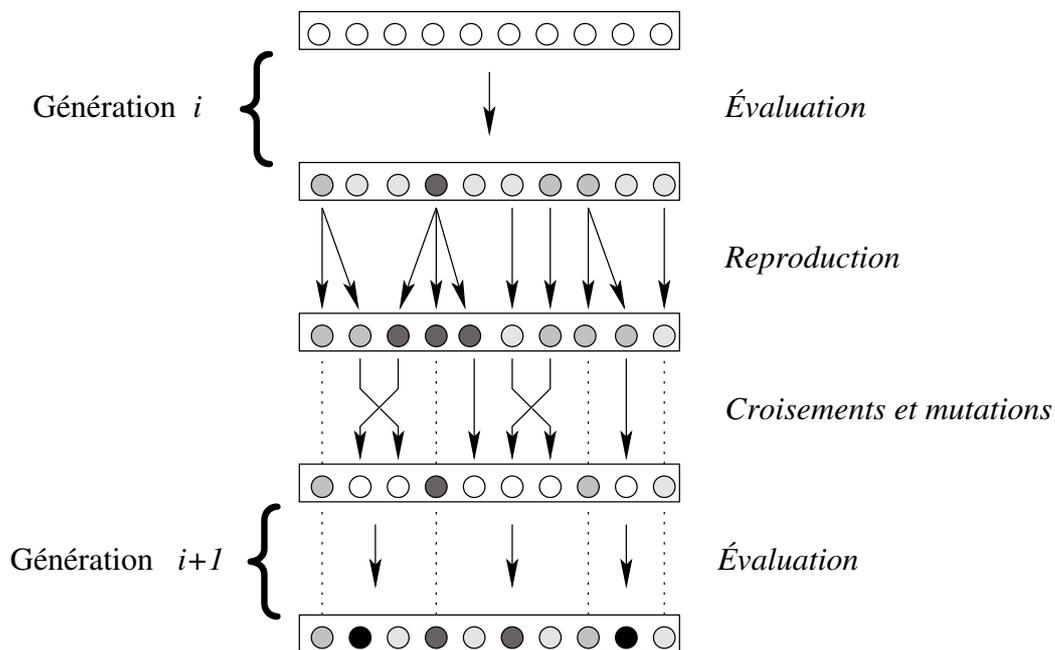


FIG. 8.1 – Étapes de l’algorithme génétique

Paramètres des algorithmes génétiques

La rapidité et la qualité de convergence d’un algorithme génétique sont influencées par de nombreux paramètres, souvent liés entre eux :

- Des paramètres numériques, dont les principaux sont :
 - **La taille de la population** (notée N_{pop}) : plus le nombre d’individus est grand, plus la probabilité de convergence est élevée, mais plus l’algorithme est lent pour calculer chaque génération. La complexité de l’algorithme génétique s’exprime généralement en $O(N_{pop})$.
 - **Le nombre de générations maximal** que doit effectuer l’algorithme, qui se définit le plus souvent en fonction de la taille de la population et du temps imparti pour résoudre le problème.
 - **Les taux de croisement et de mutation**, qui fixent la proportion de la population qui sera renouvelée à chaque génération. Ces taux ont généralement une influence importante sur la qualité de convergence de l’algorithme génétique, mais leur valeur optimale dépend de la nature des opérateurs employés et du problème à résoudre.
- Des paramètres qualitatifs, comme :
 - **La pression sélective** induite par la quantité d’individus retenus pour la reproduction. Sans cette pression, l’algorithme ne peut converger, mais si elle est trop élevée, la population risque de s’agglomérer trop rapidement autour des meilleurs individus.
 - **Le niveau de déterminisme** des opérateurs de croisement et de mutation. Lorsqu’il est trop élevé, la population ne peut qu’évoluer cycliquement entre divers optima locaux de la fonction et n’a que très peu de chance d’explorer certaines régions de l’espace de recherche. Dans le cas inverse, la convergence peut être fortement ralentie, au point de rendre la méthode inefficace.
 - **L’élitisme** de l’algorithme, qui consiste à forcer la conservation du ou des meilleurs individus trouvés, tant qu’ils ne sont pas dépassés par d’autres. Cette stratégie assure la croissance de la fonction d’adaptation du meilleur individu de la population, mais peut nuire à la diversité

de la population.

Le réglage de ces paramètres est souvent fastidieux : il nécessite beaucoup d'essais et dépend fortement du problème (de la fonction) à optimiser. Les résultats théoriques sur la convergence des algorithmes génétiques n'apportent que très peu d'éléments concluants sur ce point, qui reste de ce fait l'étape la plus difficile de leur mise en œuvre.

8.2.2 Définition de la fonction d'adaptation et gestion des contraintes

Dans le cadre des algorithmes génétiques, aucune hypothèse de régularité n'est faite sur la fonction à maximiser : elle peut ne pas être dérivable et peut même être fortement discontinue. Le calcul de la fonction d'adaptation peut ainsi être quelconque et dans certains cas, résulter de plusieurs itérations algorithmiques : dans le cas de la résolution de conflits par exemple, le calcul nécessite la simulation du trafic sur l'horizon de prédiction.

Une façon très simple de définir l'adaptation consiste donc à prendre le critère lui-même lorsqu'il doit être maximisé, ou son inverse pour un problème de minimisation (en prenant soin toutefois de traiter les cas particuliers des valeurs négatives ou nulles).

Cependant, la qualité et surtout la rapidité de convergence de l'algorithme génétique sont influencées par l'allure de la fonction d'adaptation, notamment par le nombre de ses optima locaux. Pour cette raison, dans beaucoup de problèmes, la relation entre le critère et l'adaptation n'est pas toujours directe :

Pour les problèmes avec contraintes, deux principales stratégies peuvent être employées :

- Lorsqu'il est possible d'assurer que tous les individus respectent les contraintes, l'adaptation ne représente que le critère [Michalewicz 91]. Ceci nécessite que la génération de la population initiale et que l'application des opérateurs de variations ne fournissent que des individus correspondant à une région admissible de l'espace de recherche. Cette option est généralement la plus efficace (car elle réduit directement l'espace de recherche) mais ne concerne que les problèmes avec des contraintes relativement simples (linéaires par exemple).
- Dans les autres cas (contraintes non linéaires), la fonction d'adaptation prend en compte les violations de contraintes en pénalisant les individus concernés. Un palier peut par exemple distinguer l'adaptation des individus ne respectant pas les contraintes : ces derniers ont une adaptation systématiquement inférieure au palier, et sont alors évalués sur le nombre de violations de contraintes. Cette méthode a l'avantage de conserver, dans les premières générations tout au moins, des individus situés dans le domaine non admissible : ceux-ci peuvent malgré tout générer des individus de bonne qualité car les optima sont souvent situés à la frontière du domaine admissible. Certaines techniques [Joines 94] utilisent un palier évolutif, fonction par exemple de l'adaptation du meilleur élément et du nombre de générations effectuées pour conserver plus longtemps des individus hors du domaine admissible.

Lorsque les irrégularités de la fonction objectif peuvent être lissées sans changer ses optima, la convergence de l'algorithme peut être accélérée. Cette situation peut notamment survenir dans les problèmes fortement contraints, pour lesquels certaines contraintes sont plus restrictives que d'autres : la fonction d'adaptation peut pénaliser d'avantage leurs violations, ce qui permet d'orienter progressivement (de générations en générations) la population vers le domaine admissible.

8.2.3 Définition de la population

Codage des données

Le codage des données est la première phase de l'élaboration d'un algorithme génétique :

- Cette phase détermine la structure de données qui sera utilisée pour coder le génotype des individus de la population. Cette structure dépend donc du problème à résoudre et sa contrainte principale est d'être en adéquation avec la taille de l'espace de recherche.
- Le codage des données conditionne toute l'implémentation de l'algorithme : la définition des opérateurs de croisement et de mutation et de la fonction d'évaluation des individus dépend nécessairement de la structure employée.

Les premiers algorithmes génétiques utilisaient systématiquement un codage binaire, qui reste encore aujourd'hui très populaire : dans ce cadre, le génotype associé à chaque élément de l'espace de recherche est une chaîne de booléens, destinée à être manipulée directement (sans structure apparente) par les opérateurs de croisement et de mutation. Certains problèmes se prêtent très bien à un codage binaire et les premiers résultats théoriques sur la convergence des algorithmes génétiques sont fondés sur ce type de codage.

Par la suite et pour d'autres problèmes plus structurés, le codage binaire s'est parfois révélé moins pratique ou moins efficace que des codages entiers ou réels. Dans ce cadre, le génotype se définit par une chaîne d'entiers ou de réels [Goldberg 91, Wright 91], ou encore plus généralement par une combinaison entre eux. Cette forme de codage revêt plusieurs avantages :

- Les opérations de codage et de décodage sont souvent simplifiées.
- Le génotype devient structuré et peut être décomposé en différentes parties identifiables, parfois appelées gènes. Cette structure peut être utilisée par des opérateurs de croisement et de mutation spécifiques, pour augmenter les chances d'obtenir des génotypes enfants meilleurs que leurs parents, lors du renouvellement de la population.
- Lorsque la structure du génotype est liée à la structure de l'espace de recherche, une faible modification du génotype correspond à un point voisin de l'espace de recherche : la finesse des opérateurs de variation peut être contrôlée.
- La gestion des contraintes du problème peut également être simplifiée, lorsque celles-ci ne concernent qu'un sous-ensemble réduit de variables du problème. L'exemple classique est celui de variables réelles dont le domaine est un intervalle fixé de \mathbb{R} . Le codage structuré permet d'assurer facilement le respect de ces contraintes par tous les individus de la population.

Génération initiale

La population initiale contient l'ensemble des génotypes qui serviront de souche à tous les individus générés par l'algorithme. Elle a donc une influence évidente sur la rapidité de convergence de l'algorithme, bien que cette influence ne soit pas toujours maîtrisable :

- Pour la plupart des problèmes, aucune information ne permet de présager de la position de l'optimum et la population initiale est générée aléatoirement, par un tirage uniforme dans l'espace de recherche. Dans ce cas, des traitements particuliers peuvent être définis pour assurer la diversité de la population (qui conditionne les chances d'explorer tout l'espace de recherche) : il est par exemple possible de scinder l'espace en plusieurs composantes pour y injecter au moins un individu par composante.
- Pour les problèmes fortement contraints dont le critère est relativement régulier, l'optimum recherché est souvent conditionné par les contraintes (il se trouve sur la frontière du domaine admissible) et la position de l'optimum du problème sans contrainte est parfois connue. Dans ce

type de problème, la génération initiale peut être composée d'individus générés aléatoirement dans un voisinage fixé de l'optimum sans contrainte : le non respect des contraintes forcera les individus à se diversifier dans les premières générations de l'algorithme, en explorant en premier la région où le critère est optimal.

- Lorsque des méthodes sont connues pour trouver différentes solutions non optimales mais respectant les contraintes du problème, ces solutions peuvent former autant de régions dans lesquels seront générés les individus de la population initiale.

8.2.4 Renouvellement de la population

Principe de sélection

Le rôle de la sélection est d'assurer la prolifération des meilleurs individus au détriment des plus mauvais. Au cours de cette étape, la probabilité de survie de chaque individu doit donc correspondre à son adaptation : une nouvelle population de N_{pop} individus est construite, contenant en plus grand nombre ceux qui sont les mieux adaptés. Trois techniques de sélection, décrites dans [Goldberg 89], sont très répandues :

- **Sélection par tournoi** : la nouvelle population est obtenue par N_{pop} combats entre des groupes d'individus choisis au hasard dans la population courante (selon une loi de probabilité uniforme). Lors de chaque combat, l'individu le mieux adapté l'emporte avec une probabilité $P \in]\frac{1}{2}; 1]$, où P est un paramètre de la méthode permettant le réglage de la pression sélective. Ce type de sélection a l'avantage d'être applicable quelles que soient les valeurs de la fonction d'adaptation (elles peuvent être négatives), mais la pratique montre qu'elle induit souvent une pression sélective trop faible (même avec $P = 1$).
- **Roulette wheel selection** : cette technique reproduit le principe du tirage aléatoire utilisé dans les roulettes de casino avec une structure linéaire. Chaque individu i est associé bijectivement à un segment incluant dans l'intervalle $[0; 1]$, dont la longueur l_i est proportionnelle à l'adaptation de l'individu :

Si $\sum f$ désigne la somme des adaptations de la population,

$$\text{Si } \sum f = 0, l_i = \frac{1}{N_{pop}} ; \quad \text{sinon } l_i = \frac{f_i}{\sum f}$$

La fonction d'adaptation est donc supposée à valeurs dans \mathbb{R}^+ . La nouvelle population est obtenue par N_{pop} tirages aléatoires (avec distribution uniforme de probabilité) de réels entre 0 et 1, chaque réel tiré désignant un individu sélectionné. Avec cette méthode, les individus les mieux évalués sont statistiquement sélectionnés plus souvent. Cependant, lorsque la taille de la population est réduite, le nombre de tirages est faible et l'espérance mathématique de sélection peut ne pas être obtenue : il peut en résulter un biais de sélection plus ou moins fort suivant la taille de la population.

- **Stochastic remainder without replacement** : cette technique a pour but de minimiser le biais de sélection pouvant exister avec la technique précédente. Un nombre minimal de représentants de chaque individu dans la future population est déterminé par avance en fonction de l'adaptation de l'individu, puis la population est complétée par tirages aléatoires :
 - Le nombre minimal de représentants d'un individu i est donné par $E(\tau_i)$, où E désigne la partie entière et τ_i l'adaptation de i rapporté à la moyenne des adaptations de tous les individus.

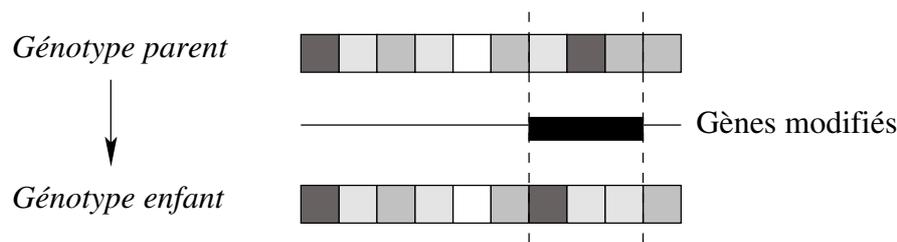


FIG. 8.2 – Opérateur de mutation

- La population est complétée par la méthode de *Roulette wheel selection*, où l'évaluation de chaque individu serait donnée par le *reste stochastique* $r_i = \tau_i - E(\tau_i)$.

Ces trois processus de sélection induisent une *pression sélective* particulière et sont très sensibles aux écarts entre les évaluations des individus. Lorsque la population n'est pas adaptée à la pression sélective, le comportement de l'algorithme génétique est biaisé :

- Soit il ne converge pas : les individus se répandent dans l'espace de recherche sans jamais progresser (pression sélective trop faible). La solution consiste généralement à affiner les opérateurs de croisement et de mutation, pour les rendre plus déterministes par exemple.
- Soit il converge trop rapidement vers un optimum local de la fonction objectif : un individu est largement mieux évalué que le reste de la population et induit une pression sélective trop forte. La population a rapidement tendance à devenir homogène et l'algorithme n'a que très peu de chance d'explorer convenablement l'espace de recherche. Ce cas est le plus fréquent et de nombreuses techniques comme le *scaling* et le *sharing*, décrites plus loin dans ce chapitre, permettent de faire varier artificiellement la pression sélective pour remédier au problème.

Opérateurs de variation

Le rôle des opérateurs de variation est d'enrichir la population avec de nouveaux individus (appelés les enfants) et de garantir ainsi que tout point de l'espace de recherche a une chance d'être exploré, quelque soit la population initiale (propriété d'ergodicité).

Deux formes d'opérateurs sont utilisées conjointement. Elles se distinguent par le nombre de parents impliqués :

La mutation consiste à créer un enfant en modifiant localement et aléatoirement le génotype d'un seul individu (figure 8.2). C'est donc l'opérateur le plus simple et il est suffisant pour garantir la propriété d'ergodicité sur le plan théorique : il est essentiel pour le fonctionnement de l'algorithme génétique et certaines implémentations n'utilisent que cette forme d'opérateur [Fogel 66b].

Le croisement (figure 8.3) crée plusieurs enfants à partir de plusieurs parents (généralement, deux enfants sont créés à partir de deux parents). Son rôle premier est d'être complémentaire à l'opérateur de mutation, en ne fournissant par exemple que des nouvelles combinaisons de gènes déjà existants. Lorsque l'opérateur de mutation est suffisamment aléatoire, les croisements utilisés peuvent être à l'inverse plus déterministes.

De nombreux exemples de mutations et de croisements *types* peuvent être trouvés dans la littérature [Bridges 91], en fonction de la nature du codage des données (binaire, entière, réelle ou autre), la seule véritable contrainte étant qu'ils ne nuisent pas à l'ergodicité de l'algorithme.

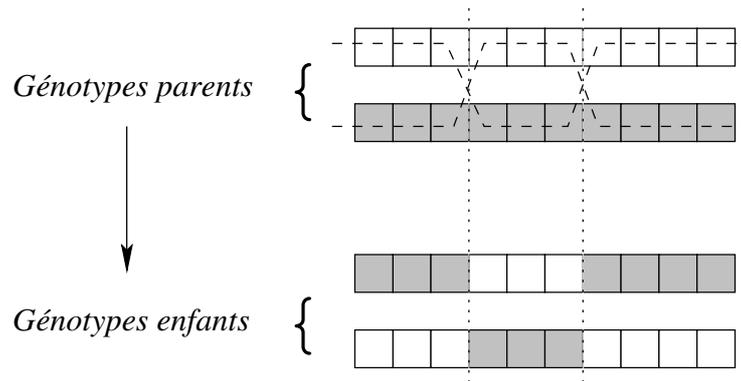


FIG. 8.3 – Opérateur de croisement

La principale difficulté relative à la définition des opérateurs est de doser l'amplitude des altérations de génotype : si elle est trop faible, la population risque de se cloisonner aux premiers optima locaux rencontrés. Si elle est trop forte, l'algorithme est incapable d'affiner correctement la recherche en fin de convergence. Deux classes particulières d'opérateurs permettent de réduire ce problème :

- **Les opérateurs adaptatifs** [Eiben 99, Yang 02] : l'amplitude des modifications est évolutive, fonction de différents indicateurs tels que le nombre de générations effectuées, l'adaptation moyenne de la population, ou encore la distance moyenne entre les individus (lorsque l'espace de recherche en possède une).
- **Les opérateurs auto-adaptatifs** [Schwefel 81] : l'amplitude maximale des modifications des opérateurs est codée dans une partie du génotype de chaque individu. Cette amplitude évolue donc au cours des croisements ou des mutations et subit le processus de sélection.

8.2.5 Convergence

Les premières études théoriques sur les algorithmes génétiques se sont intéressées à leur fonctionnement avec un codage binaire. Ces résultats sont connus sous le nom de la *théorie des schémas*. Ils sont largement développés dans [Goldberg 89] et résumés dans [Alliot 92]. Ils permettent de mieux comprendre l'effet du processus de sélection et de certains opérateurs de mutation et de croisements sur les *schémas* constituant le génotype des individus : il apparaît par exemple qu'avec les opérateurs envisagés, les schémas de petite taille (définis sur un nombre restreint de bits) sont favorisés et ont plus de chance d'être reproduits au cours des générations.

Des propriétés de convergence des algorithmes génétiques avec codage réel ont été développées par R. Cerf [Cerf 94] à partir des travaux de O. Catoni [Catoni 90] et de Trouvé [Trouvé 93]. L'algorithme génétique est considéré comme un processus Markovien, perturbé par les opérateurs génétiques [Freidlin 83]. Cette étude théorique permet notamment d'établir le comportement asymptotique des algorithmes génétiques, ainsi que les cycles qui régissent la dynamique du processus. Elle démontre par ailleurs que l'algorithme peut converger sans croisement.

Ces résultats théoriques ne rendent cependant pas encore compte du fonctionnement des algorithmes génétiques tels qu'ils sont utilisés en pratique, où leur convergence est fortement accélérée par des raffinements tels que le scaling et le sharing (décrits ci-après), ou l'utilisation d'opérateurs adaptatifs dont l'efficacité est augmentée au fil des générations à partir d'approximations quadratiques de la fonction d'adaptation [Auger 04]. De plus, les nombreux paramètres de ces algorithmes impliquent un réglage final qui nécessite souvent l'emploi de la méthode *essai-erreur*.

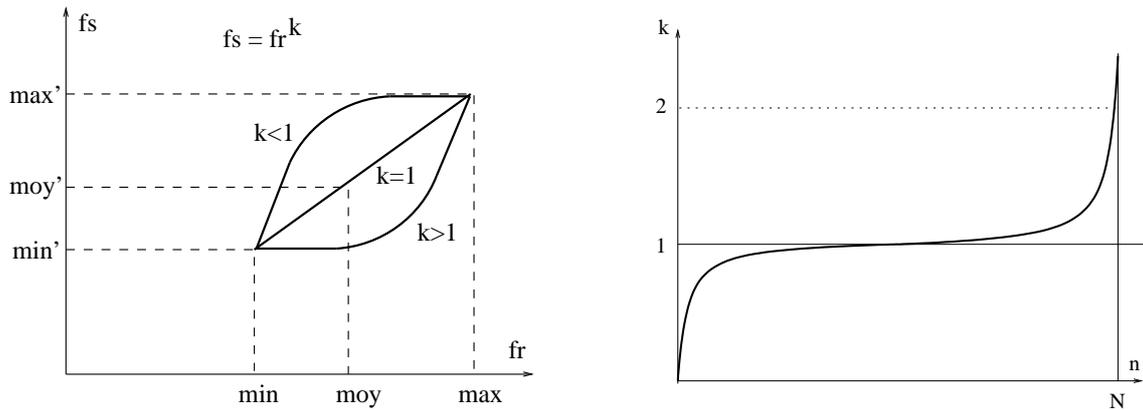


FIG. 8.4 – Fonction de scaling exponentiel

8.3 Améliorations et adaptations classiques

8.3.1 Scaling

Le *scaling* est une mise à l'échelle de la fonction d'adaptation, souvent indispensable au bon fonctionnement de l'algorithme génétique. Son but est d'augmenter ou de diminuer artificiellement les écarts d'évaluation entre les individus, pour ajuster la pression sélective en fonction de la génération courante. Le paramètre d'ajustement est l'espérance du nombre de copies du meilleur individu, en général fixé à 2. La fonction donnant l'adaptation d'un individu ne sera donc plus la fonction f à optimiser, mais son image f_s par une *fonction de scaling*, de \mathbb{R}^+ dans lui-même, généralement strictement croissante. Deux principales fonctions de scaling peuvent être utilisées :

Le scaling linéaire [Michalewicz 92], défini par une fonction de scaling affine :

$$f_s = af + b \quad \text{avec } a > 0$$

La position relative de a par rapport à 1 détermine si la pression sélective est affaiblie et favorise l'exploration de l'espace ($a < 1$) ou si elle est au contraire renforcée pour favoriser les modes dominants ($a > 1$). Si la technique de sélection nécessite que la fonction d'adaptation soit positive ou nulle, la constante b permet d'ajuster le scaling pour que f_s soit également à valeurs dans \mathbb{R}^+ . Il est naturellement envisageable de faire varier a et b au fil des générations.

Le scaling exponentiel [Michalewicz 92], où la fonction de scaling est donnée par :

$$f_s = f^k$$

- Pour k proche de 0, la pression sélective est fortement réduite : l'algorithme génétique se comporte comme une exploration aléatoire de l'espace de recherche.
- Pour k proche de 1, le scaling devient transparent.
- Pour $k > 1$, la pression sélective est renforcée et seuls les meilleurs individus parviennent à se maintenir.

Dans la pratique, k varie généralement de façon croissante au fil des générations, avec par exemple (figure 8.4) :

$$k = \left(\tan \left[\left(\frac{n}{N_{gen} + 1} \right) \frac{\pi}{2} \right] \right)^p \quad \text{avec } \begin{cases} n & : \text{ la génération courante} \\ N_{gen} & : \text{ le nombre maximal de générations} \\ p & : \text{ fixe l'intensité du scaling} \end{cases}$$

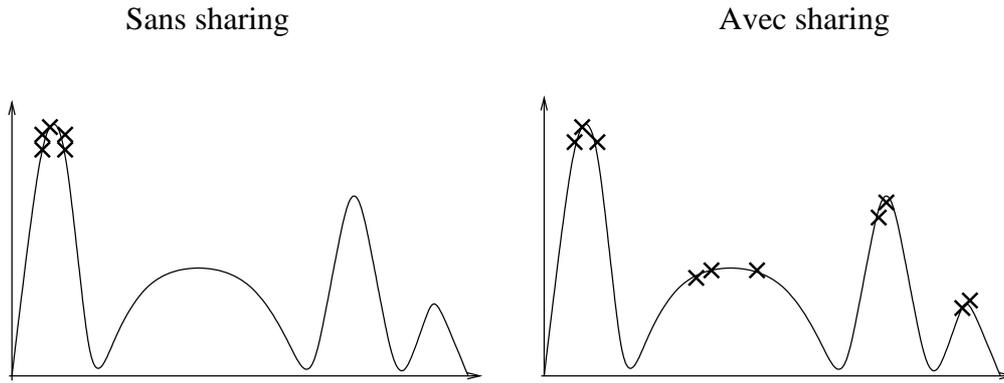


FIG. 8.5 – Objectif du sharing

Dans la pratique, la valeur $p = 0.1$ s'est souvent montrée efficace.

8.3.2 Sharing

Principe

Le *Sharing* relativise l'adaptation des individus par rapport à leur proximité (ou densité) dans l'espace de recherche. L'objectif est de forcer la population à se répartir dans différentes régions de l'espace de recherche : son utilisation est particulièrement recommandée avec des fonctions objectifs présentant de nombreux optima locaux, pour éviter que la population ne se regroupe que sur un seul d'entre eux (figure 8.5). Sa mise en œuvre nécessite la connaissance d'une distance sur l'espace de recherche ou entre les individus.

De la même façon que le *scaling*, le *sharing* modifie l'adaptation de chaque individu, en la pénalisant en fonction du taux d'aggrégation de la population dans un voisinage de l'individu. Pour cela, on suppose définie une distance d , représentative des différences entre individus. La nouvelle fonction d'adaptation f' d'un individu i est donnée par :

$$f'(i) = \frac{f(i)}{m'_i}; \quad m'_i = \sum_{j=1}^{N_{pop}} S(d(x_i, x_j))$$

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

- Le paramètre σ_{share} définit la zone d'influence des individus : seuls les individus dont la distance est inférieure à σ_{share} se pénalisent mutuellement. Sa valeur doit être déterminée en fonction du problème traité et de la distance définie sur l'espace de recherche. Il est souvent utile, pour ce réglage, de normaliser les distances (entre 0 et 1 par exemple).
- Le paramètre α fixe l'intensité du *sharing* : plus α est grand, plus les groupes d'individus agglomérés sont pénalisés (figure 8.6).

Dans la pratique, ce type de *sharing* donne de bons résultats mais sa complexité en $\mathcal{O}(N_{pop}^2)$ par rapport au nombre d'individus peut s'avérer pénalisante, comparée aux autres opérations de l'algorithme génétique qui s'effectuent généralement en $\mathcal{O}(N_{pop})$. C'est pourquoi le *sharing clusterisé* lui est souvent préféré.

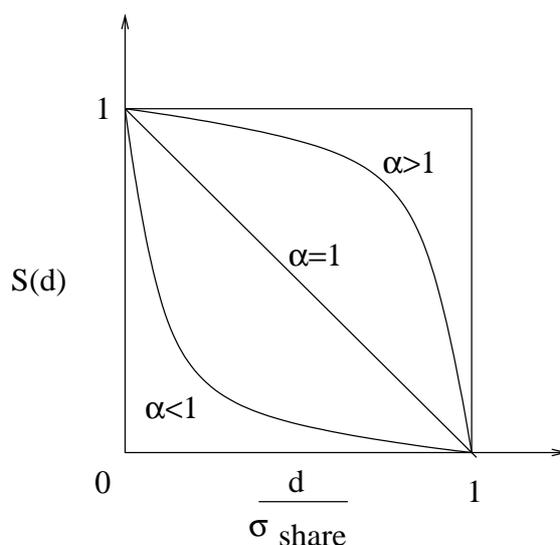


FIG. 8.6 – Allure de la fonction de sharing

Sharing clusterisé

Le *sharing clusterisé* [Yin 93] permet de réduire la complexité du sharing en répartissant les individus de la population par groupes de proximité, appelés *clusters*. Deux paramètres $d_{\min} < d_{\max}$ définissent la façon avec laquelle sont construits ces clusters :

- Initialement, chaque individu de la population est considéré comme le centre d'un cluster, dont il est l'unique élément.
- Si deux centres sont à une distance inférieure à d_{\min} , les deux clusters correspondants sont réunis dans un unique cluster, dont le centre est le milieu des deux centres initiaux.
- Si la distance d'un individu au centre du cluster le plus proche est inférieure à d_{\max} , l'individu est ajouté au cluster et le centre de ce dernier est recalculé, comme un barycentre entre l'individu et l'ancien centre du cluster.

Cette méthode peut être implémentée en $\mathcal{O}(N_{pop} \log N_{pop})$ opérations [Yin 93], en utilisant un algorithme de classification hiérarchique ascendante [Day 84] pour obtenir les clusters représentant les regroupements d'individus dans l'espace de recherche. Le sharing consiste alors à modifier l'adaptation de chaque individu comme suit :

$$f'(i) = \frac{f(i)}{m'_i}; \quad m'_i = n_i \left(1 - \left(\frac{d_i}{2d_{\max}} \right)^\alpha \right)$$

$$\text{Avec : } \begin{cases} n_i : \text{nombre d'individus dans le même cluster que } i \\ d_i : \text{distance entre } i \text{ et le centre de son cluster} \\ \alpha : \text{intensité du sharing} \end{cases}$$

Ce type de sharing s'avère aussi efficace que le sharing initial. La méthode nécessite cependant le calcul de barycentres entre les points de l'espace de recherche, ce qui peut sortir du cadre de certains problèmes. D'autre part, le dosage des paramètres d_{\min} et d_{\max} n'est pas toujours évident : leur valeur devrait être du même ordre que la distance minimale entre deux optima de la fonction, ce qui peut rarement être connu à priori. Le clustering adaptatif permet de s'abstraire de ce dosage.

Clustering adaptatif

Le *clustering adaptatif* est une technique de sharing clusterisé permettant de réguler automatiquement les paramètres d_{\min} et d_{\max} en fonction de l'adaptation du meilleur individu de chaque cluster [Alliot 96] : l'idée est d'augmenter le nombre de clusters (par diminution de d_{\min} et d_{\max}) lorsque beaucoup d'entre eux contiennent des individus *bien* adaptés et au contraire de forcer leur agrégation dans les autres cas. La technique se définit en fonction de trois nouveaux paramètres, dont les valeurs sont comprises entre 0 et 1, mais dont le réglage s'avère plus général que celui de (d_{\min} , d_{\max}) :

- Le *taux de sharing* τ_s est le seuil caractérisant les *bons* clusters, par rapport à l'adaptation du meilleur individu de la population (0.9 s'avère souvent efficace en pratique) ;
- Deux *seuils d'adaptation* $S_1 < S_2$ déterminent les cas où (d_{\min} , d_{\max}) doivent être adaptés (0.75 et 0.85 en pratique).

L'évolution de d_{\min} et d_{\max} est régi par une variable Δ , maintenue de générations en générations :

- Initialement, d_{\min} et d_{\max} ont une valeur nulle et Δ vaut 2.
- À chaque génération, les clusters sont formés à partir des valeurs courantes de d_{\min} et d_{\max} , selon la technique décrite dans la partie précédente. Les valeurs pour la génération suivante sont calculées comme suit :

1. Le taux de *bons* clusters est calculé :

$$S = \frac{1}{N_c} |\{j \in 1, \dots, N_c \mid \frac{f_j}{f_{\max}} \geq \tau_s\}|$$

Où f_j est l'adaptation du meilleur individu du cluster j et f_{\max} l'adaptation du meilleur individu de toute la population.

2. Δ est mis-à-jour :

$$\begin{cases} \Delta \leftarrow 0.95\Delta & \text{si } S < S_1 \text{ et } \Delta > 1 \\ \Delta \leftarrow 1.05\Delta & \text{si } S > S_2 \text{ et } \Delta < 100 \\ \Delta \text{ reste inchangé} & \text{dans les autres cas.} \end{cases}$$

3. La distance moyenne entre les individus et les centres de clusters fournit les nouvelles valeurs de d_{\min} et d_{\max} :

$$\begin{cases} d_{\max} \leftarrow \frac{d_{moy}}{\Delta} \\ d_{\min} \leftarrow \frac{d_{\max}}{3} \end{cases} \quad \text{avec} \quad d_{moy} = \frac{\sum_{i=1}^{N_{pop}} \sum_{j=1}^{N_c} d(x_i, C_j)}{N_{pop} N_c}$$

Où N_c est le nombre de clusters et N_{pop} le nombre d'individus.

En pratique, cette technique a été mise à l'épreuve et s'est déjà révélée efficace sur des problèmes de résolution de conflit [Granger 02].

8.3.3 Problèmes partiellement séparables

Principe

Dans [Durand 96b], des opérateurs de variation spécifiques sont définis et permettent d'accélérer significativement la rapidité de convergence des algorithmes génétiques, pour un certain type de problèmes d'optimisation souvent étudié, dits *partiellement séparables* [Griewank 82].

Ces problèmes se caractérisent par un critère à minimiser, fonction de n variables (x_1, x_2, \dots, x_n) , qui peut s'exprimer comme la somme de m fonctions positives, chacune d'elles ne dépendant que d'un sous-ensemble de variables :

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

- Si le problème était totalement séparable, aucune variable x_j ne serait impliquée dans la définition de plusieurs fonctions F_i . La minimisation de F reviendrait à minimiser séparément chaque fonction F_i .
- L'idée intuitive pour résoudre le problème partiellement séparable par algorithme génétique consiste à s'inspirer des valeurs des F_i pour orienter les opérateurs de variation : les variables (x_j) relatives aux fonctions F_i à forte valeur doivent avoir plus de chance d'être modifiées.

Le codage des données est dans ce cas supposé direct : le génotype des individus est composé des n variables (x_1, x_2, \dots, x_n) (qui peuvent être indifféremment entières, réelles, ou autres).

Fonctions d'adaptation partielles

Pour définir les opérateurs de variation, chaque individu de la population est associé à autant de *fonctions d'adaptation partielles* $(G_k)_{1 \leq k \leq n}$ qu'il y a de variables :

$$\forall k \in \{1, \dots, n\}, G_k(x_1, x_2, \dots, x_n) = \sum_{i \in S_k} \frac{F_i(x_{j_1}, x_{j_2}, \dots, x_{j_{n_i}})}{n_i}$$

$$\text{Avec } \begin{cases} S_k = \{i \in \{1, \dots, n\} \mid x_k \text{ est une variable de } F_i\} \\ n_i \text{ est le nombre de variables de } F_i \end{cases}$$

On a donc par construction :

$$F(x_1, x_2, \dots, x_n) = \sum_{k=1}^n G_k(x_1, x_2, \dots, x_n)$$

Intuitivement, $G_k(x_1, x_2, \dots, x_n)$ représente la participation de la variable x_k au critère.

Opérateurs de variation adaptés

Selon l'idée intuitive ayant guidé la construction des fonctions d'adaptation partielles, la probabilité qu'un opérateur de variation modifie la variable x_k doit augmenter lorsque $G_k(x_1, x_2, \dots, x_n)$ est élevé. La principale difficulté consiste à doser le niveau de déterminisme des opérateurs : le risque est d'orienter trop souvent l'élaboration des enfants, pour finalement obtenir une population qui oscille indéfiniment entre plusieurs schémas opposés ou qui converge trop vite vers un mauvais optimum local.

Pour cette raison, la définition des opérateurs est fonction d'un paramètre $\Delta > 0$, qui permet d'ajuster le niveau de déterminisme (plus il est faible, plus les opérateurs seront déterministes) :

La mutation d'un individu $X = (x_1, x_2, \dots, x_n)$ peut se définir comme suit :

$$\text{Soit } \begin{cases} M = \max_{k \in \{1, \dots, n\}} G_k(X) \\ S = \{k \in \{1, \dots, n\} \mid G_k(X) \geq M - \Delta\} \end{cases}$$

Un élément k de S est tiré au hasard et la variable x_k est modifiée aléatoirement.

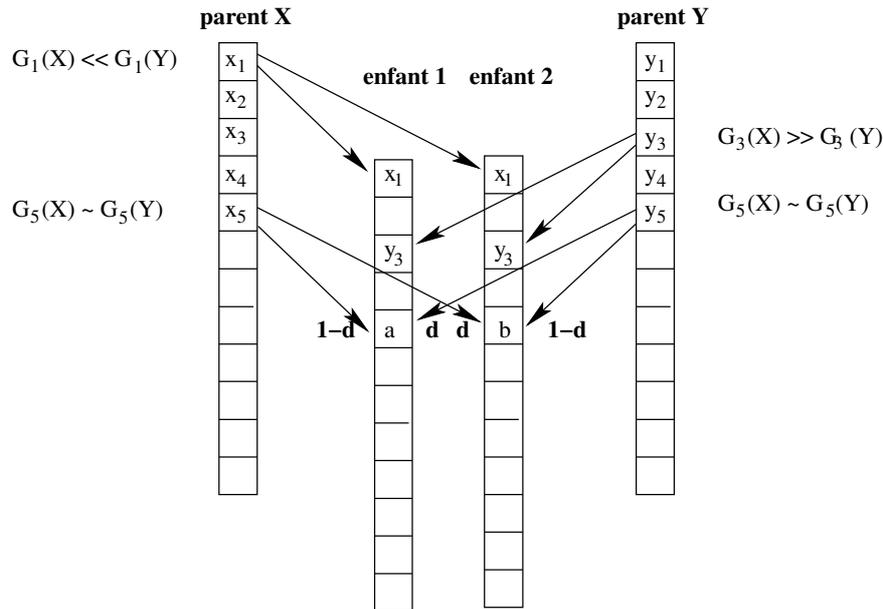


FIG. 8.7 – Croisement adapté

Le croisement de deux parents $X = (x_1, x_2, \dots, x_n)$ et $Y = (y_1, y_2, \dots, y_n)$ est illustré figure 8.7 :

$\forall k \in \{1, \dots, n\}$,

- Si $G_k(X) \geq G_k(Y) + \Delta$, alors les deux enfants héritent de y_k ;
- Si $G_k(Y) \geq G_k(X) + \Delta$, alors les deux enfants héritent de x_k
- Dans les autres cas, les variables x_k et y_k sont croisées selon des techniques classiques (croisement barycentrique pour des réels par exemple).

Extensions de la méthode

La notion de fonction d'adaptation partielle (*partial fitness*) est souvent employée dans le cadre des algorithmes génétiques, même pour des problèmes qui ne sont pas partiellement séparables au sens défini dans cette partie : dans ce cadre plus général, il peut s'agir de toute fonction heuristique G_k , mesurant « l'adaptation » d'une partie du génotype, selon des critères qui sont généralement dépendants du problème traité.

Les opérateurs de variation, tels que ceux définis ci-dessus, peuvent utiliser les fonctions d'adaptation partielles pour altérer plus souvent les parties du génotype qui semblent mal adaptées.

8.3.4 Optimisation locale en fin de convergence

Les algorithmes génétiques sont efficaces pour explorer des espaces de grande dimension et y trouver différentes régions intéressantes, contenant des points proches des optima recherchés. Cependant, l'aspect stochastique de la méthode d'exploration (qui est indispensable à son ergodicité) ne garantit pas que la meilleure solution trouvée correspond exactement à l'optimum de la région explorée : il est souvent bénéfique d'appliquer une méthode d'optimisation locale aux meilleurs individus trouvés. Deux approches peuvent être employées :

- L'optimisation locale peut être appliquée une fois l'algorithme terminé, sur le ou les meilleurs

éléments de la population finale. Il peut s'agir par exemple d'une méthode de descente, convergeant vers l'optimum local le plus proche de chaque individu.

- L'optimisation locale peut être appliquée lors des mutations ou des croisements, avec une probabilité à fixer (opérateur hybride). La méthode d'optimisation peut être déterministe (méthode de descente le long de la plus grande pente par exemple) ou stochastique : le recuit simulé [Toulouse 85, Ingber 92] est souvent utilisé dans ce contexte. Ce type d'opérateur hybride peut cependant avoir deux conséquences néfastes : il ralentit l'algorithme et peut orienter trop souvent la population vers de mauvais optima.

8.4 Application au trafic au sol

8.4.1 Méthodologie

Dans cette partie, un algorithme génétique est appliqué au problème formulé au chapitre 5. Les premières parties en décrivent l'implémentation générale, qui sera ensuite mise en pratique en deux étapes :

1. Dans un premier temps, l'algorithme sera *étalonné* (réglage des paramètres et sélection des options les plus efficaces) sur un échantillon de situations de trafic à forte densité, générées par simulation avec résolution par la méthode 1 contre n définie au chapitre précédent.
2. L'efficacité de l'algorithme génétique sera ensuite analysée, par applications successives des résolutions à chaque situation d'une journée de trafic.

8.4.2 Définition de la population

Codage des données

Le codage des données est défini directement en fonction des variables du problème, précisées dans la partie 5.2.1. Il s'agit donc d'un codage entier, dans lequel le génotype d'un individu est défini par $(1 + 2\Omega)N$ variables entières bornées, où N est le nombre d'avions impliqués :

$$\left\{ \begin{array}{ll} (\nu_i)_{1 \leq i \leq N} \in \{1, \dots, \eta\}^N & \text{les chemins suivis par les avions ;} \\ (p_{ik})_{1 \leq i \leq N; 1 \leq k \leq \Omega} & \text{les positions d'attente des avions ;} \\ (t_{ik})_{1 \leq i \leq N; 1 \leq k \leq \Omega} & \text{les heures de fin d'attente des avions.} \end{array} \right.$$

Pour restreindre au maximum l'espace de recherche, les contraintes élémentaires, relatives aux intervalles de définition des variables, seront supposées toujours vérifiées par les individus de la population. Si t_{\min}^i désigne l'heure relative du début du mouvement i et t_{\max}^i son heure relative maximale de fin sans attente (mesurée sur son chemin le plus long) :

$$\forall i \in \{1, \dots, N\}, \forall k \in \{1, \dots, \Omega\}, \left\{ \begin{array}{l} 0 \leq p_{ik} \leq t_{ik} \leq H_p \\ t_{\min}^i \leq p_{ik} \leq t_{\max}^i \end{array} \right.$$

Dans la suite, on désignera par (ν, p, t) les $(1 + 2\Omega)N$ variables du problème.

Génération initiale

La complexité de la résolution de conflits est due aux contraintes du problème : le critère à optimiser (cumul des retards des avions) est relativement régulier, mais l'optimum est conditionné par

les règles de séparation qui morcellent fortement le domaine admissible. Pour cette raison, la population initiale n'est pas obtenue par tirages uniformes sur l'espace de recherche, mais par une loi de distribution favorisant les faibles valeurs de la pénalité (sans contrainte) :

- Les faibles valeurs de ν_i ont plus de chance d'être choisies (les chemins sont indicés par temps de parcours croissants) ;
- Les valeurs de p_{ik} sont choisies par tirage uniforme sur leurs intervalles de définition ;
- Les valeurs de t_{ik} sont plus souvent proches de leur valeur minimale.

La population initiale contiendra donc le plus souvent des individus situés hors du domaine admissible mais faiblement pénalisés en délai et en détour : le processus de sélection et les opérateurs de variation devront orienter progressivement les individus vers le domaine admissible, en augmentant leur pénalité.

8.4.3 Sélection

Critère d'optimisation

Les individus de la population représentent des instanciations des variables qui ne respectent pas nécessairement les contraintes du problème. Le critère global $F(\nu, p, t)$ doit donc prendre en compte à la fois les violations de contraintes et la qualité de la solution lorsque celle-ci est admissible. Son calcul nécessite :

1. Le calcul des trajectoires des N avions ;
2. La détection des violations de contraintes, représentées par les valeurs $c_{ij}(t) \in \{0, 1\}$, pour $t \in \{0, \dots, H_p\}$ ($c_{ij}(H_p) = 1$ représente le non respect d'une contrainte relative à l'effet de l'horizon).

Deux définitions différentes du critère global, notées F_1 et F_2 , seront comparées : elles ne diffèrent que par la pénalité associée aux violations de contraintes, ce qui les rend totalement équivalentes sur le domaine admissible :

1. La première définition du critère pénalise chaque violation de contrainte de façon homogène :
 - Si le nombre de conflits est non nul,

$$F_1(\nu, p, t) = K \sum_{t=0}^{H_p} \sum_{i=1}^{N-1} \sum_{j=i+1}^N c_{ij}(t)$$

- Sinon,

$$F_1(\nu, p, t) = \alpha \mathcal{L}(\nu) + \mathcal{D}(p, t)$$

Où K caractérise les instanciations non admissibles des variables ($K \gg NH_p$). Les autres notations sont celles définies au chapitre 5 lors de la modélisation du problème :

- $\mathcal{L}(\nu) = \sum_{i=1}^N l_i(\nu_i)$ est la somme des retards dus au choix des chemins ;
 - $\mathcal{D}(p, t) = \sum_{i=1}^N t_{i\Omega} - p_{i\Omega}$ est l'attente cumulée des avions ;
 - α le coefficient de pondération entre attente et détour ;
2. La deuxième définition du critère a pour but de diminuer l'importance des optima locaux situés hors du domaine admissible, autour desquels la population risque de s'attarder sans jamais aboutir à une solution : il faut pour cela remarquer qu'il est généralement inutile de résoudre les derniers conflits d'un avion lorsque les premiers ne sont pas résolus (dans l'ordre chronologique pour l'avion). L'importance d'un conflit peut donc être relativisée par son heure, d'où la deuxième définition du critère :

- Si le nombre de conflits est non nul,

$$F_2(\nu, p, t) = K \sum_{t=0}^{H_p-1} \sum_{i=1}^{N-1} \sum_{j=i+1}^N \frac{H_p + 1 - t}{T} c_{ij}(t)$$

Où T est une constante pondérant l'importance relative des premiers conflits.

- Sinon, $F_2(\nu, p, t) = F_1(\nu, p, t) = \alpha \mathcal{L}(\nu) + \mathcal{D}(p, t)$

Fonction d'adaptation

La fonction d'adaptation f de l'algorithme génétique doit être maximisée : elle est définie de manière inversement proportionnelle au critère F (défini par F_1 ou F_2). Un palier fixe assure que les individus représentant des solutions admissibles soient toujours mieux évalués ($f \geq \frac{1}{2}$) que les autres ($f < \frac{1}{2}$) :

- Si $F(\nu, p, t) < K$, l'individu représente une solution admissible :

$$f(\nu, p, t) = \frac{1}{2} + \frac{K}{2K + F(\nu, p, t)}$$

- Sinon,

$$f(\nu, p, t) = \frac{K}{K + F(\nu, p, t)}$$

Scaling

Le scaling exponentiel décrit dans la partie 8.3.1 est utilisé pour moduler la pression sélective induite par la fonction d'adaptation. Ses paramètres sont réglés avec des valeurs classiques pour le problème de résolution de conflits [Durand 96a, Granger 02] :

$$f_s = f^{k(n)} \text{ avec } k(n) = \left(\tan \left[\left(\frac{n}{N_{gen} + 1} \right) \frac{\pi}{2} \right] \right)^{0.1}$$

Ce scaling est nécessaire pour éviter la disparition trop rapide des individus violant les contraintes, lorsque les premières solutions admissibles apparaissent dans la population.

8.4.4 Caractère partiellement séparable du problème

Matrice d'adaptation

Le critère global $F(\nu, p, t)$ est défini en fonction de la pénalité de chaque avion et du nombre de conflits de chaque paire d'avions. L'ensemble de ces informations, qui ne peuvent être connues que via la prédiction de trafic et la détection de conflits, peut être représenté par une matrice $N \times N$ symétrique, qui sera appelée la *matrice d'adaptation* (figure 8.8) :

$$\begin{cases} m_{ii} = \alpha l_i(\nu_i) + d_i \text{ avec } d_i = t_{i\Omega} - p_{i\Omega} \\ \text{Pour } i \neq j : m_{ij} = \sum_{t=0}^{H_p} P(t) c_{ij}(t) \end{cases}$$

- m_{ii} représente la pénalité de l'avion i et ne dépend que de ses variables $\nu_i, (p_{ik}, t_{ik})_{1 \leq k \leq \Omega}$;
- m_{ij} pour $i \neq j$ représente la pénalité due aux violations de contraintes de la paire (i, j) et ne dépend que des variables de i et j . $P(t)$ est un coefficient général de pondération, défini en accord avec le critère global utilisé (F_1 ou F_2).

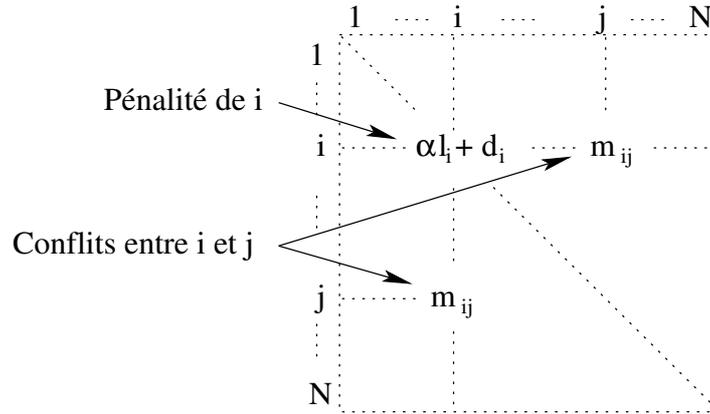


FIG. 8.8 – Matrice d’adaptation

L’utilité de cette matrice est double :

- Elle permet de décomposer le critère global F à minimiser en N fonctions d’adaptation partielles, pour guider les opérateurs de variation.
- Elle permet d’accélérer la phase d’évaluation de l’algorithme génétique (la plus pénalisante dans le cadre de la résolution de conflit) : lors des croisements et des mutations, les éléments de la matrice correspondant à des variables non modifiées peuvent être mémorisés : par la suite, il ne sera pas nécessaire de détecter les conflits entre les paires d’avions non modifiées, lors de l’évaluation du nouvel individu. Si les opérateurs de variation ne modifient que les variables d’un seul avion, la phase d’évaluation sera en $\mathcal{O}(N)$ au lieu de $\mathcal{O}(N^2)$.

Fonctions d’adaptation partielles

Le critère global F à minimiser peut s’exprimer en fonction de la matrice d’adaptation :

- Si la matrice d’adaptation est diagonale,

$$F(\nu, p, t) = \sum_{i=1}^N m_{ii}$$

- Sinon,

$$F(\nu, p, t) = K \sum_{i=1}^N \sum_{j < i} m_{ij}$$

Les adaptations partielles $(G_i)_{1 \leq i \leq N}$, telles que définies dans la partie 8.4.4 peuvent être extraites. Chacune d’elles est relative à l’ensemble des $1 + 2\Omega$ variables du $i^{\text{ème}}$ avion et représente la participation de ce dernier à la pénalité globale :

- Si la matrice d’adaptation est diagonale, $G_i(\nu, p, t) = m_{ii}$
- Sinon, $G_i(\nu, p, t) = \frac{K}{2} \sum_{j \neq i} m_{ij}$

Opérateurs de variation associés

Les opérateurs de variations adaptés aux problèmes partiellement séparables, présentés dans la partie 8.4.4 sont appliqués avec les adaptations partielles $(G_i)_{1 \leq i \leq N}$:

- **La mutation** d'un individu A modifie aléatoirement les variables d'un seul avion, présentant une des pires adaptations partielles.
- **Le croisement** est une adaptation de celui présenté dans la partie 8.4.4 : deux individus A et B donnent deux individus A' et B' , obtenus par la modification des variables d'un seul avion dans un des parents : A' (respectivement B') est une copie de A (resp. B), dans laquelle les variables d'un avion présentant une des pires adaptations partielles sont remplacées par celles du même avion dans B (resp. A).

Cette définition des opérateurs, qui ne modifie que les variables d'un seul avion d'un parent, permet de conserver une évaluation en $O(N)$, par héritage de la matrice d'adaptation du parent dont seule une colonne (ou une ligne) doit être recalculée. Aucun paramètre n'est utilisé pour contrôler le niveau de déterminisme des opérateurs, car ce dernier dépend directement des différences d'adaptations partielles :

- Dans les premières générations de l'algorithme, la plupart des individus sont hors du domaine admissible et présentent des adaptations partielles très différentes. Le niveau de déterminisme des opérateurs est alors relativement élevé : les trajectoires les moins conflictuelles auront de fortes chances d'être reproduites. Le rôle des opérateurs de variation, à ce stade de la convergence, est d'orienter la population vers le domaine admissible. Ceci pourrait aboutir à une convergence prématurée vers le premier optimum local admissible trouvé : la diversité de la population doit être assurée par application d'une méthode de sharing.
- Lorsque la population contient beaucoup d'individus situés dans le domaine admissible, les différences entre les adaptations partielles se resserrent et les opérateurs de variation deviennent de moins en moins déterministes.

8.4.5 Sharing

Le sharing est nécessaire pour éviter l'homogénéisation prématurée de la population autour des premiers individus représentant des solutions admissibles. La méthode de sharing clusterisé, avec clustering adaptatif décrite dans la partie 8.3.2 est donc implémentée : elle nécessite la définition d'une distance et de barycentres dans l'espace de recherche :

La distance entre deux individus A et B est définie en fonction des différences de choix de chemins et de durée d'attente :

$$d(A, B) = \frac{\sum_{i=1}^N (H_p \chi(\nu_i^A, \nu_i^B) + |(t_{i\Omega}^A - p_{i\Omega}^A) - (t_{i\Omega}^B - p_{i\Omega}^B)|)}{2NH_p}$$

$$\text{Avec } \begin{cases} \chi(\nu_i^A, \nu_i^B) = 0 & \text{si } \nu_i^A = \nu_i^B \\ \chi(\nu_i^A, \nu_i^B) = 1 & \text{si } \nu_i^A \neq \nu_i^B \end{cases}$$

Cette définition de la distance est normalisée : $0 \leq d(A, B) \leq 1$.

Le barycentre de deux individus A et B pondéré par les coefficients positifs α_A et α_B est un individu C défini par :

$$\forall i \in \{1, \dots, N\},$$

$$\begin{cases} \nu_i^C = \nu_i^A & \text{si } \alpha_A \geq \alpha_B \text{ et } \nu_i^C = \nu_i^B \text{ sinon ;} \\ \forall k \in \{1, \dots, \Omega\}, p_{ik}^C = \frac{\alpha_A p_{ik}^A + \alpha_B p_{ik}^B}{\alpha_A + \alpha_B} & \text{et } t_{ik}^C = \frac{\alpha_A t_{ik}^A + \alpha_B t_{ik}^B}{\alpha_A + \alpha_B} \end{cases}$$

Les variables résultantes (p_{ik}^C, t_{ik}^C) doivent éventuellement être corrigées pour satisfaire les contraintes élémentaires liées à leur domaine de définition.

8.4.6 Terminaison de l'algorithme

Critère d'arrêt

Le critère d'arrêt de l'algorithme est défini en fonction de deux paramètres N_{gen} et N_{opt} :

- N_{gen} est le nombre maximal absolu de générations : passé ce nombre de générations, l'algorithme est terminé même si aucune solution admissible n'a été trouvée.
- N_{opt} est un nombre maximal de générations *stationnaires*, destiné à interrompre l'algorithme lorsque la population ne semble plus progresser. Une génération sera dite stationnaire lorsque l'adaptation du meilleur élément est supérieure à $\frac{1}{2}$ et qu'elle est la même qu'à la génération précédente.

Optimisation locale en fin de convergence

Une fois terminé, l'algorithme génétique retourne la dernière génération :

- Si cette population contient des individus représentant des solutions admissibles, ces derniers peuvent être optimisés localement par une méthode déterministe, pour assurer la convergence précise vers le meilleur optimum local situé dans les régions explorées : pour chaque individu, la méthode 1 contre n définie au chapitre précédent est appliquée itérativement jusqu'à ce qu'elle ne permette plus d'améliorer la solution : à chaque itération, une trajectoire optimale est affectée à un avion en considérant que celles de tous les autres sont fixées.
- Si la meilleure adaptation est inférieure à $\frac{1}{2}$, la situation de trafic n'a pu être résolue : la méthode étant stochastique, il peut être bénéfique de recommencer la résolution (avec une initialisation différente du hasard).

8.4.7 Étalonnage de l'algorithme

Mise en situation

L'étalonnage d'un algorithme génétique est souvent la phase la plus délicate : le nombre de configurations possibles des paramètres (nombre de générations, taille de la population, taux de mutation et de croisement) est lui-même combinatoire. L'algorithme est de plus stochastique et il convient d'effectuer plusieurs résolutions du même problème dans la même configuration pour obtenir une bonne indication de sa performance.

Pour ce faire, des situations de trafic au sol à Roissy sont extraites des simulations 1 contre n du chapitre précédent et servent de référence : elles sont de densité de trafic moyenne à élevée et sont considérées dans les conditions suivantes :

- Vitesse maximale de roulage en ligne droite : 10 m/s ;
- Incertitude sur la vitesse : 20% ;
- Horizon de prédiction : 10 minutes ;
- Nombre maximal de positions d'attente par avion : 2 ;
- Nombre maximal de chemins envisagés par avion et par bretelle de piste : 12 ;

Chaque situation est résolue plusieurs fois (avec différentes initialisations du hasard), dans chaque configuration envisagée.

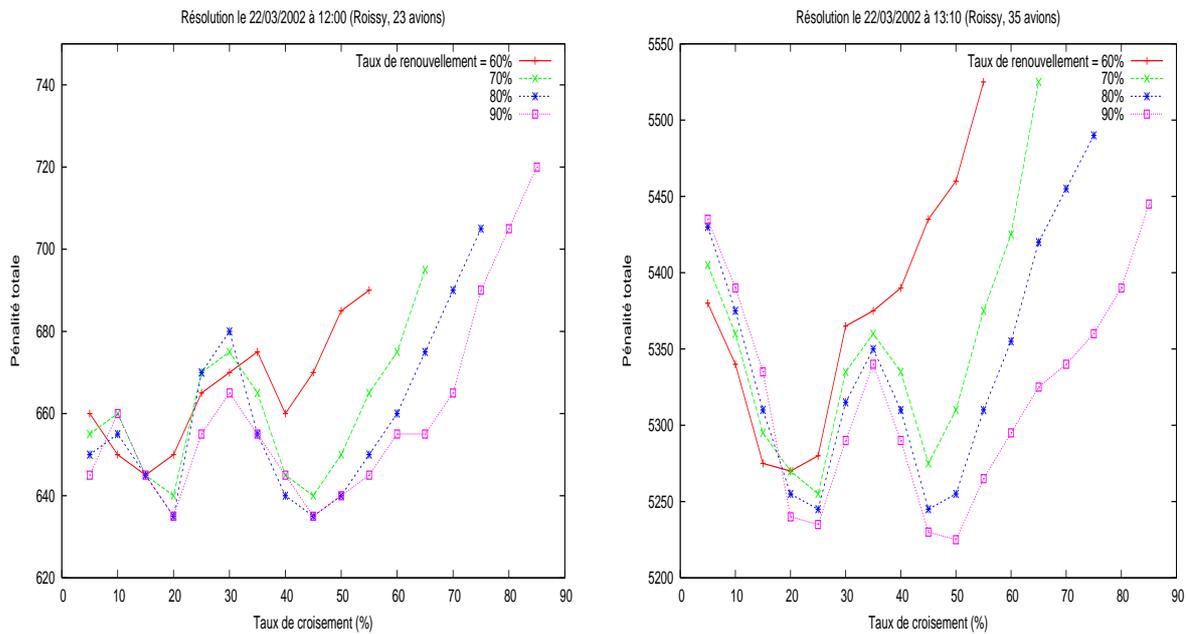


FIG. 8.9 – Influence du taux de croisement avec le critère homogène

Taux de croisement et de mutation

Le premier réglage concerne les taux de croisement et de mutation. Les valeurs envisagées pour ces taux sont comprises entre 10% et 85%, avec un pas de 5 et de telle sorte que leur somme, qui sera appelée le *taux de renouvellement*, varie de 60% à 90% avec un pas de 10 : ceci représente au total 56 configurations possibles de l'algorithme, soient 1120 résolutions de chaque situation (20 par configuration) avec chacun des deux critères envisagés. L'algorithme est appliqué avec un nombre maximal de $N_{gen} = 200$ générations, de $N_{opt} = 15$ générations stationnaires et une population de $N_{pop} = 200$ éléments.

Les figures 8.9 et 8.10 donnent les pénalités moyennes obtenues dans chaque configuration pour les deux critères envisagés et pour deux situations extrêmes parmi les situations étudiées (l'une correspond à la situation à 12h00, moyennement chargée et l'autre est la plus pénalisante, à 13h10) :

- Globalement, les configurations « déséquilibrées » (où l'un des taux est très faible) aboutissent à des pénalités significativement supérieures : ceci montre que les deux opérateurs sont complémentaires et participent tous deux à l'obtention d'un optima de bonne qualité.
- Les meilleurs optima sont obtenus avec un taux de renouvellement de 90% : cette valeur relativement élevée peut s'expliquer par le niveau de déterminisme important des opérateurs (adaptés au problème partiellement séparable), qui nécessitent que la population se diversifie le plus possible. Dans toutes les situations, les meilleures solutions sont obtenues avec un taux de croisement égal au taux de mutation, autour de 45%, bien que la configuration avec un taux de croisement plus faible (entre 25% et 30%) aboutisse également à des optima intéressants.
- Le premier critère d'optimisation (pénalités homogènes associées aux violations de contrainte) est moins sensible aux variations des taux de croisement et de mutation mais se révèle moins performant : il aboutit à des pénalités supérieures et à 12% de cas d'échecs dans la situation à 13h10 (contre 5% avec le critère pondéré). Ceci montre toute l'importance des optima locaux situés hors du domaine admissible : avec le premier critère, ils sont plus nombreux et plus

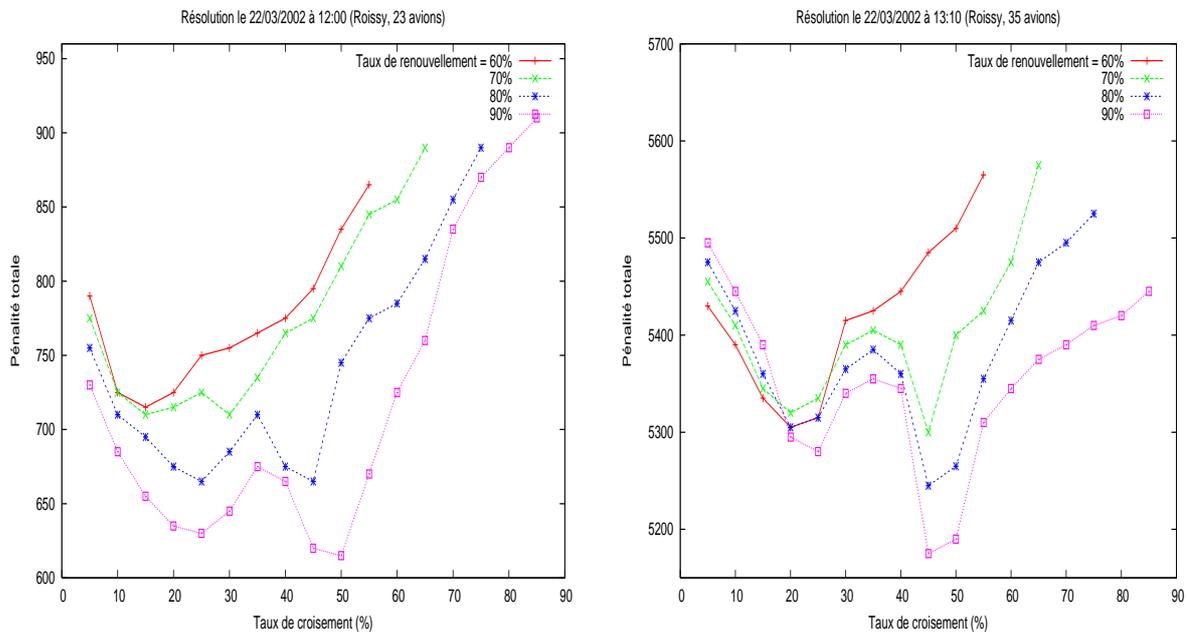


FIG. 8.10 – Influence du taux de croisement avec le critère pondéré

prononcés, ce qui piège plus souvent la population et ralentit la convergence de l'algorithme.

Taille de la population et nombre de générations

En règle générale, la convergence de l'algorithme est améliorée par augmentation du nombre d'individus ou du nombre de générations. Ces paramètres doivent pourtant être limités en fonction du temps imparti pour résoudre le problème. Si N_{pop} désigne le nombre d'individus et N_{gen} le nombre maximal de générations, le temps de résolution s'exprime globalement en $O(N_{pop}N_{gen})$, car l'opération la plus pénalisante est l'évaluation d'un individu. Il est possible de diminuer l'un des paramètres en augmentant l'autre : là encore, le réglage optimal des deux paramètres ne peut se faire que par essais successifs.

Dans cette optique, la vitesse de convergence de l'algorithme génétique est étudiée avec une population de 50 à 250 individus (par pas de 50), avec un nombre de générations maximal inversement proportionnel à la population, de 200 (pour 250 individus) à 1000 (pour 50 individus). L'évolution de la pénalité associée au meilleur individu est corrélée au nombre total d'individus explorés, donné par le produit du nombre de génération et de la taille de la population. Les autres paramètres de l'algorithme sont fixés avec les valeurs optimales obtenues dans la partie précédente (critère pondéré avec un taux de croisement et de mutation de 45%).

La figure 8.11 donne l'évolution moyenne de la pénalité du meilleur individu, sur 20 résolutions d'un cluster de 22 avions (un des clusters les plus importants de la simulation 1 contre n , extrait de la situation à 13h10 à Roissy). Ces résultats montrent que la convergence la plus rapide (en temps et non en nombre de générations) est obtenue avec une population de 150 individus :

- Lorsque la taille de la population est moindre, l'algorithme converge plus rapidement mais vers de moins bonnes solutions : la diversité de la population n'est pas assurée et l'exploration de l'espace de recherche est insuffisante.
- Au delà de 150 individus, l'algorithme trouve des optima équivalents mais avec plus d'individus

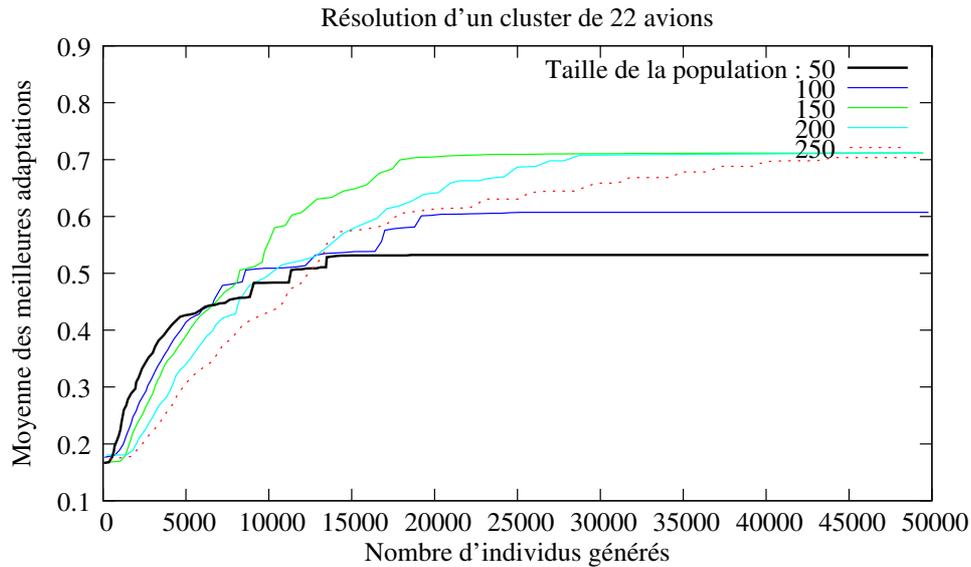


FIG. 8.11 – Influence de la taille de la population sur la convergence

explorés : le nombre de générations pour converger est moindre, mais le temps d'exécution réel de l'algorithme est plus long.

- Avec 150 individus, la solution finale est trouvée après exploration de 20000 individus (soit moins de 150 générations) : les futures applications de l'algorithme seront effectuées avec cette taille de population et un nombre maximal de 150 générations.

Efficacité du sharing

La figure 8.12 donne l'évolution moyenne de la pénalité du meilleur individu en fonction du nombre de générations, pour la résolution du cluster de 22 avions, dans quatre configurations : la première sans sharing, les trois autres avec différentes valeurs du *taux de sharing*, défini dans la partie 8.3.2 :

- Ces résultats confirment l'importance du sharing pour la résolution de conflits : avec sharing, la diversité de la population est maintenue artificiellement pour éviter qu'elle se fasse « piéger » dans un optimum local et la solution trouvée est meilleure (bien que le nombre de générations nécessaires pour y parvenir soit plus grand).
- La valeur optimale de 90% pour le taux de sharing est également confirmée. Ce taux influence la taille des clusters de sharing :
 - Pour une valeur plus faible du taux, la plupart des clusters sont considérés comme des groupes d'individus bien adaptés et ils sont moins souvent agrégés : beaucoup d'individus restent seuls dans leur cluster, ce qui nuit au concept même du sharing.
 - Pour une valeur supérieure, les clusters sont au contraire trop souvent agrégés et la population ne se diversifie donc plus suffisamment.

8.4.8 Application à la simulation

L'algorithme génétique, paramétré en fonction des résultats de la partie précédente, est appliqué dans l'environnement de simulation : la résolution de chaque situation est utilisée pour générer la

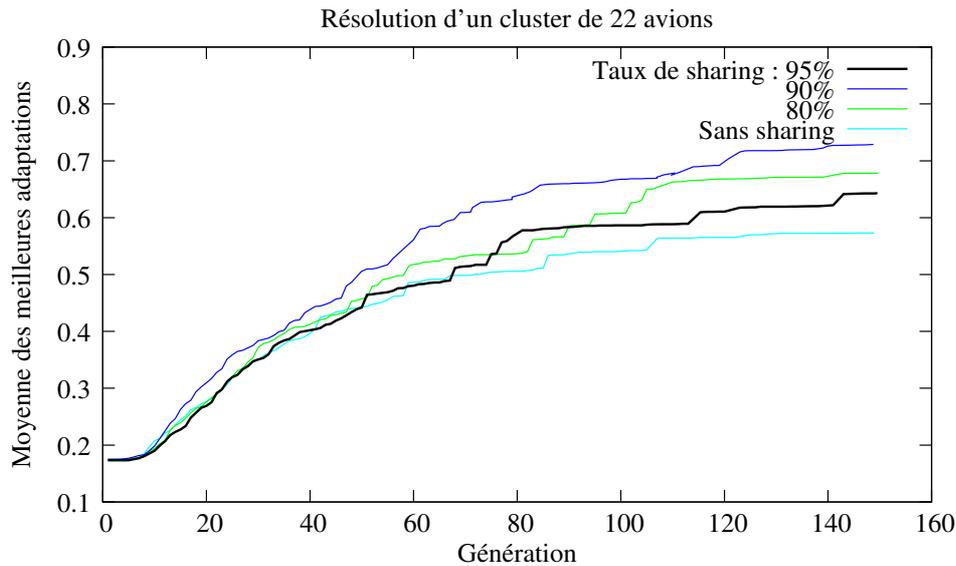


FIG. 8.12 – Influence du taux de sharing sur la convergence

situation suivante, tout au long d'une journée de trafic à Roissy et à Orly. Les résultats sont comparés à ceux obtenus avec la méthode de résolution 1 contre n :

Les paramètres de la simulation sont les mêmes que ceux utilisés précédemment :

- Trafic à Roissy : 22/03/2002 (1468 mouvements)
- Trafic à Orly : 18/06/99 (807 mouvements)
- Horizon de prédiction : 10 minutes
- Période de résolution : 2 minutes
- Nombre maximal de positions d'attente : 2
- Vitesse maximale : 10 m/s
- Incertitude sur la vitesse : 20%
- Sens préférentiels appliqués

Les paramètres de l'algorithme génétique sont les suivants :

- Taille de la population : 200
- Nombre maximal de générations : 100
- Taux de mutation : 45%
- Taux de croisement : 45%
- Taux de sharing : 90%
- Nombre de répétitions de l'AG en cas d'échec : 2

Les résultats globaux des simulations (effectuées avec le même processeur à 1.6GHz) sont juxtaposés à ceux obtenus par la méthode de résolution 1 contre n , appliquée avec recherche du classement optimal, développée au chapitre précédent :

Aéroport	Résolution	Temps de calcul	Pénalité moyenne (sec.)	Cas d'échec
Roissy	1 contre n	6h.15	111	Aucun
	Algorithme génétique	8h.30	100	
Orly	1 contre n	1h45	41	Aucun
	Algorithme génétique	2h15	37	

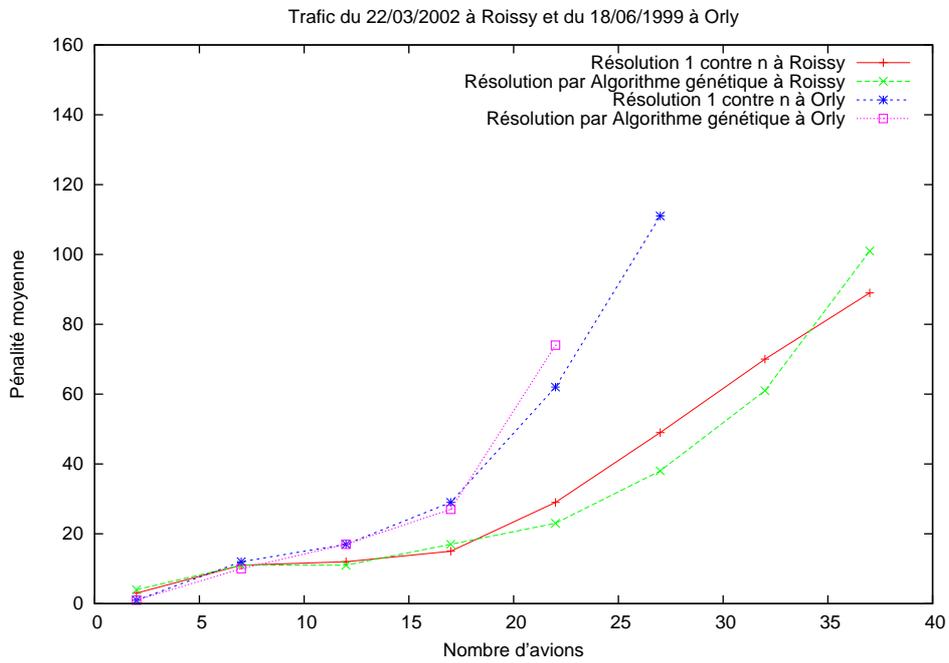


FIG. 8.13 – Pénalité moyenne en fonction du nombre d'avions pour l'AG

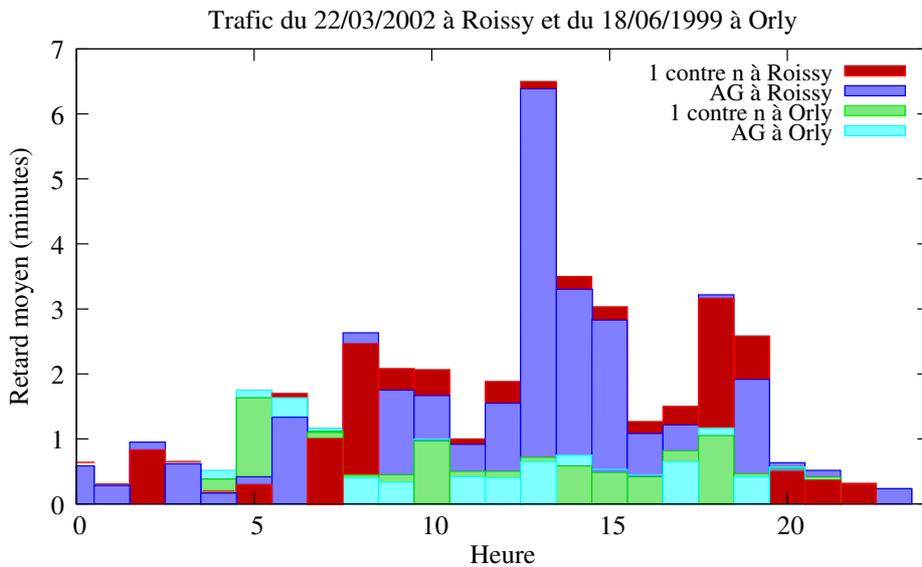


FIG. 8.14 – Retard moyen avec l'AG

Les figures 8.13 et 8.14 donnent les résultats plus détaillés de ces simulations :

- À Roissy, les solutions trouvées par l’algorithme génétique dans les situations moyennement chargées sont meilleures qu’avec la méthode 1 contre n . Par contre, les situations les plus chargées sont moins bien résolues : ces situations correspondent à des files d’attente importantes d’avions devant la piste et l’algorithme génétique ne trouve pas systématiquement l’optimum global (il atteint en quelque sorte sa limite). Dans ces situations, le classement des avions (opéré pour la méthode 1 contre n) s’avère beaucoup mieux adapté à la résolution du problème. L’effet est visible sur les retards résultants (figure 8.14), qui sont diminués de plus de 30 secondes dans beaucoup de périodes de la journée, mais qui restent élevés pendant les pics de trafic.
- À Orly, les deux méthodes semblent équivalentes : la pénalité est légèrement supérieure dans les périodes les plus chargées, mais il faut remarquer que le nombre maximum d’avions dans la simulation résolue par algorithme génétique est inférieur à celui de la simulation 1 contre n (24 contre 27 avions), ce qui signifie que le débit est amélioré et que la pénalité est répartie différemment. Les retards révèlent le même phénomène qu’à Roissy : l’algorithme génétique perd légèrement son efficacité pendant les pics de trafic, mais trouve de meilleures solutions le reste du temps.

8.5 Conclusion

Les concepts hérités de la résolution de conflits aériens par algorithmes génétiques peuvent être appliqués à la séparation des avions au roulage sur un aéroport. Le problème reste partiellement séparable et les opérateurs de variation adaptés, lorsqu’ils sont associés à une méthode de sharing, permettent une convergence satisfaisante de l’algorithme : toutes les situations de trafic au sol sont résolues, en un temps opérationnellement acceptable. Le problème est considéré dans sa forme la plus générale et la plupart des solutions sont meilleures que celles de la méthode déterministe, qui ne peut raisonnablement que traiter un problème simplifié.

De plus, l’utilisation des algorithmes génétiques permet de ne faire aucune hypothèse particulière sur le critère à optimiser, qui reste de ce fait totalement malléable. Il faut cependant remarquer que la méthode déterministe, qui elle, est définie expressément pour minimiser les temps de roulage, s’avère particulièrement adaptée aux situations les plus chargées : dans ces situations, caractérisées par des files d’attente conséquentes à plusieurs endroits de l’aéroport, le classement des avions se révèle un bon moyen pour obtenir rapidement une solution proche de l’optimum. À ce stade, l’idée de concilier les deux méthodes d’optimisation semble donc prometteuse et le prochain chapitre lui est consacré.

Chapitre 9

Résolution par des méthodes hybrides

9.1 Introduction

Les méthodes de résolution présentées dans les deux chapitres précédents possèdent des caractéristiques très différentes :

- La méthode 1 contre n trouve l'optimum d'un problème simplifié de façon déterministe : l'espace de recherche restreint est toujours parcouru de la même façon et la méthode aboutit systématiquement à la même solution. L'exploration est définie en fonction de certaines propriétés particulières du critère à optimiser, qui est donc fortement ancré dans l'implémentation de la méthode.
- L'algorithme génétique est une méthode stochastique qui s'applique sur le problème dans sa forme la plus générale : aucune hypothèse n'est faite sur le critère et tous les points de l'espace de recherche ont une probabilité non nulle d'être explorés, bien que l'obtention de l'optimum global ne soit pas garanti. Plusieurs solutions admissibles sont généralement trouvées et elles peuvent être différentes d'une exécution à l'autre.

Il est naturel de vouloir concilier les avantages des deux méthodes, soit pour accélérer la convergence de l'algorithme génétique, soit pour élargir le champ d'application de la méthode 1 contre n .

Dans les deux cas, la démarche correspond à l'élaboration d'une méthode *hybride* et de nombreuses techniques [Lobjois 98] peuvent être utilisées : les principales d'entre elles sont décrites dans la première partie de ce chapitre et deux applications au problème de résolution de conflit sont fournies dans les parties suivantes.

9.2 Définition de méthodes hybrides

9.2.1 Recherche exacte et recherche approchée

De nombreux problèmes du monde réel se placent dans le cadre très général des problèmes de satisfaction de contrainte (CSP), où un critère défini par une fonction de coût doit être minimisé dans un domaine admissible des variables. Lorsque la complexité de ces problèmes est élevée (problèmes NP-complets notamment), deux grandes familles de méthodes de résolution se distinguent pour les résoudre :

Les méthodes exactes procèdent à une exploration complète et généralement déterministe de l'espace de recherche. Ces méthodes utilisent des algorithmes de type *Branch & Bound* pour instancier progressivement les variables du problème [Papadimitriou 82, Horst 95]. Des mécanismes

classiques de *backtrack* permettent d'abandonner une branche de l'arbre de recherche à chaque fois que l'instanciation partielle des variables viole certaines contraintes ou qu'une preuve permet d'établir que les solutions contenues dans la branche ne sont pas optimales. Par contre, toutes les autres branches sont systématiquement développées : ces méthodes deviennent inappropriées lorsque la taille du problème augmente alors que le temps imparti pour le résoudre est restreint. Dans ce cas, la méthode peut être adaptée, par simplification du problème ou par restriction de l'espace de recherche, mais l'optimum global initialement recherché ne sera plus forcément obtenu.

Les méthodes approchées se basent sur une exploration locale (méthodes de descente [Selman 98]) et aléatoire (algorithmes génétiques, recuit simulé, algorithmes de colonies de fourmis) de l'espace de recherche. Elles ne garantissent donc pas l'obtention d'une solution optimale mais permettent, dans de nombreuses applications, de trouver des solutions intéressantes en un temps raisonnable à des problèmes fortement combinatoires : elles n'ont naturellement d'intérêt que lorsqu'aucune méthode exacte ne permet de résoudre le problème dans le temps imparti.

Ces deux types de méthodes relèvent de deux approches opposées, mais les mécanismes qu'elles utilisent peuvent être combinés pour constituer des méthodes *hybrides*, dans lesquelles les deux concepts coopèrent : *rather than competing, can we cooperate* [Freuder 95]. Dans ce contexte, deux principales techniques d'hybridation peuvent logiquement être envisagées, selon le type de la méthode qui sera *hybridée* :

- La première consiste à utiliser des méthodes approchées pour orienter ou participer à l'exploration de l'espace de recherche d'une méthode exacte : selon les cas, la méthode hybride résultante conserve ou perd sa complétude initiale.
- Inversement, une méthode approchée peut être enrichie par des mécanismes issus des méthodes exactes. La méthode résultante reste forcément une méthode approchée, mais peut en être largement améliorée.

9.2.2 Hybridation de méthodes exactes

Ajout de mécanismes approchés

Une méthode exacte se définit en fonction d'heuristiques et de mécanismes d'élagage et de retour arrière (*backtrack*) dans l'arbre de recherche. Des méthodes approchées peuvent facilement être introduites à ces deux niveaux :

- Les heuristiques permettent d'orienter l'exploration : elles déterminent généralement quelles variables doivent être instanciées dans les prochaines étapes de l'algorithme et avec quelles valeurs. Dans ce cas, la qualité de l'heuristique influence directement la rapidité de l'algorithme mais ne conditionne pas l'obtention d'un optimum global : les méthodes approchées peuvent donc être utilisées pour fournir ces heuristiques sans nuire à la complétude de la méthode.
- Les mécanismes d'élagage de l'arbre reposent sur la vérification des contraintes (vérifications de cohérence des instanciations partielles) et sur l'utilisation de bornes inférieures et supérieures délimitant le coût des solutions devant être explorées. De nombreuses techniques, plus ou moins liées au problème d'optimisation, peuvent être utilisées pour définir ces bornes : la connaissance d'une solution admissible en est l'exemple le plus classique, car elle fournit directement une borne supérieure du coût des solutions optimales [Gondran 85]. Là aussi, une méthode approchée peut apporter une borne supérieure intéressante, en fournissant rapidement une solution sous-optimale au problème initial [Wallace 96] ou même à chaque sous-problème relatif à une instanciation partielle des variables [Mazure 96].

- Les mécanismes de *backtrack* définissent l'endroit où l'exploration doit reprendre une fois qu'une branche de l'arbre a été abandonnée : là encore, une méthode approchée peut être utilisée pour choisir la nouvelle branche à explorer de façon non déterministe [Prestwich 01].

Partage de l'exploration

Une méthode approchée peut également être utilisée pour participer à l'exploration de l'arbre de recherche. Ce partage de l'exploration peut se définir de différentes façons :

- La méthode approchée peut prendre le relais à partir d'une certaine profondeur de la recherche : dans ce cas, la profondeur de transition peut être dynamique et dépendre par exemple du temps restant pour trouver une solution, ou d'une estimation de la qualité et de la taille de la branche à explorer.
- La méthode approchée peut être utilisée pour définir les sous-espaces que doit explorer la méthode exacte. La taille relative des sous-espaces par rapport à l'espace initial peut servir de paramètre à l'hybridation : lorsqu'elle varie dans $]0, 1]$, la méthode passe progressivement d'une méthode approchée à une méthode exacte [Barnier 97].

Cette forme d'hybridation transforme le plus souvent la méthode exacte en une méthode approchée, car l'obtention d'une solution optimale n'est plus garantie.

9.2.3 Hybridation de méthodes approchées

Une méthode approchée se définit en fonction :

1. d'un état initial, représentant généralement une ou plusieurs instanciations des variables ;
2. d'un opérateur de transition, permettant de passer de l'état courant à l'état suivant, choisi généralement aléatoirement dans un voisinage du premier.
3. d'un critère d'arrêt.

L'hybridation peut être effectuée à tous les niveaux de l'algorithme, bien que dans la plupart des cas, elle ne concerne que la phase de transition. Elle ne sera envisagée ici que sur les deux premiers points.

Élaboration de l'état initial

L'état initial influence fortement la vitesse de convergence des méthodes approchées et peut parfois conditionner l'ensemble des solutions susceptibles d'être trouvées (pour les méthodes de descente particulièrement). L'emploi de mécanismes déterministes pour améliorer l'état initial est un procédé courant :

- Des mécanismes déterministes peuvent fournir des instanciations admissibles à partir desquelles l'exploration de la méthode approchée peut débiter : ceci se révèle très utile pour des problèmes fortement contraints, où il est difficile d'atteindre aléatoirement le domaine admissible. Ce genre de technique peut notamment s'appliquer à la population initiale d'un algorithme génétique, pour que les individus respectent les contraintes du problème dès la première génération.
- Une méthode exacte peut être appliquée sur une partie réduite de l'espace de recherche, pour en extraire rapidement un optimum sur ce sous-espace. Chaque optimum local ainsi obtenu peut former un point de départ intéressant pour une méthode d'exploration aléatoire. Ceci permet par exemple de former un individu particulier de la population initiale d'un algorithme génétique, ou encore d'initialiser la répartition des phéromones dans un algorithme de colonies de fourmis.

Exploration du voisinage

Des techniques issues des méthodes exactes peuvent être appliquées dans une méthode approchée, lors de la transition de l'état courant à l'état suivant :

- Des mécanismes classiques de renforcement de cohérence (tels que le *forward checking* ou l'*arc-cohérence*) utilisés dans les méthodes exactes permettent de restreindre le choix de l'état suivant vers les seuls sous-espaces voisins qui sont susceptibles de contenir l'optimum global [Bowen 96].
- Une méthode exacte peut permettre d'explorer complètement un voisinage de l'état courant, pour générer un nouvel état « optimal » dans ce voisinage. Dans ce contexte, il faut néanmoins veiller à ne pas rendre la méthode approchée trop déterministe, car elle convergerait alors systématiquement vers le même optimum local et perdrait donc son intérêt : un paramètre fixe généralement la participation de la méthode exacte par rapport aux transitions classiques de la méthode.

9.3 Greffe d'un opérateur déterministe sur l'algorithme génétique

9.3.1 Principe

Une première façon d'allier la méthode 1 contre n à la résolution par algorithme génétique consiste à hybrider ce dernier au niveau de l'opérateur de mutation : la résolution 1 contre n , appliquée sur un avion particulier et en considérant que les trajectoires des autres avions sont fixées, permet de trouver rapidement la solution optimale pour cet avion. Cette opération consiste à chercher un optimum local dans un voisinage restreint d'un individu de la population.

La méthode de résolution résultante conserve la forme générale d'un algorithme génétique, mais qui s'aide d'un opérateur déterministe pour optimiser localement le génotype de certains individus : cette approche s'intègre donc dans le cadre général des algorithmes *mémétiques*, tels que décrits dans [Radcliffe 94]. L'application de l'opérateur déterministe ne doit pas être systématique, sous peine de perdre l'ergodicité de l'algorithme : le degré d'hybridation sera contrôlé par le taux d'application de cet opérateur. Ce taux sera un paramètre supplémentaire et l'algorithme génétique devra donc subir un nouvel étalonnage.

9.3.2 Implémentation

Cette forme d'hybridation est relativement simple à réaliser : l'algorithme génétique n'est modifié qu'au niveau de l'opérateur de mutation et la méthode 1 contre n pour l'optimisation d'un avion est déjà implémentée. Le nouvel opérateur de mutation d'un individu A de la population se définit toujours en fonction des adaptations partielles des variables de chaque avion et du taux d'application ($\tau_{opt} \in [0, 1]$) de l'opérateur déterministe :

- Un avion possédant une des pires adaptations partielles est sélectionné.
- Un nombre ρ est tiré aléatoirement dans $[0, 1]$, avec une loi de distribution uniforme :
 - Si $a < \tau_{opt}$, la méthode 1 contre n est appliquée à l'avion sélectionné, sur le chemin décrit dans l'individu A et avec les trajectoires des autres avions décrites par A . La trajectoire obtenue fournit les attentes imposées à l'avion dans le nouvel individu (le chemin de l'avion n'est pas modifié).
 - Sinon, l'opérateur de mutation est inchangé : les variables de l'avion sélectionné sont modifiées aléatoirement, avec un tirage uniforme dans leur domaine de définition.

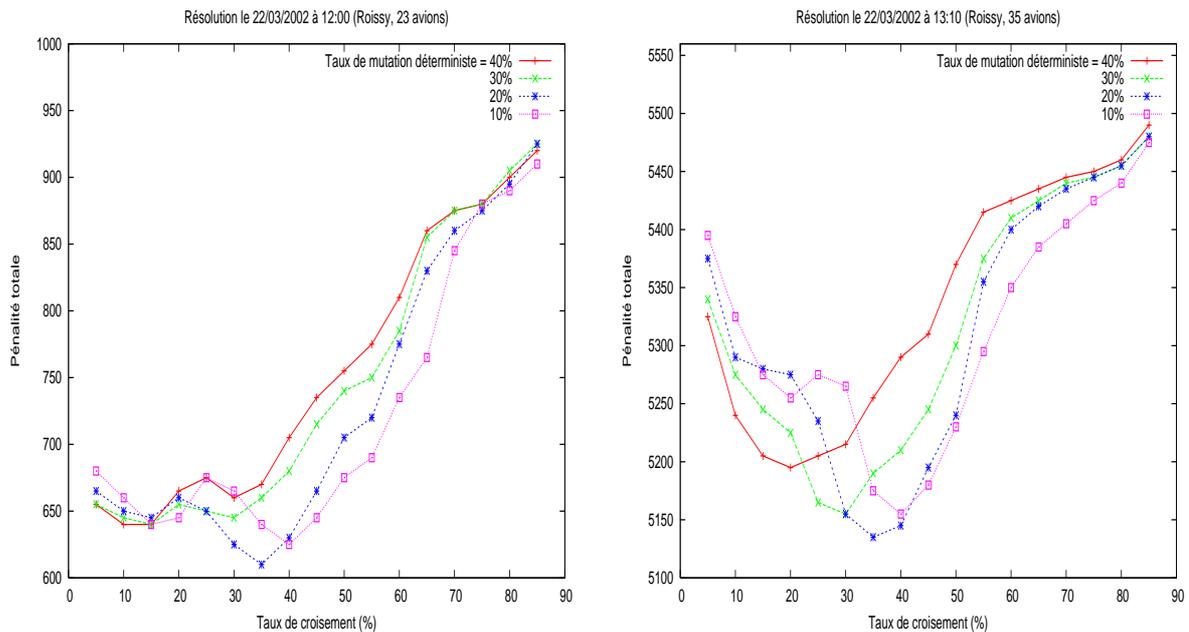


FIG. 9.1 – Influence des taux de croisement et de mutation déterministe

9.3.3 Application

Étalonnage

Comme au chapitre précédent, l’algorithme génétique doit être étalonné (sur des situations de trafic préétablies) avant d’être appliqué à la simulation. Le réglage concerne ici le taux d’application τ_{opt} de la mutation déterministe. Cependant, ce taux dénature l’opérateur de mutation initial : les taux de croisements et de mutations obtenus précédemment ne sont donc plus forcément adaptés et leur valeur optimale peut dépendre de τ_{opt} . Étant donné que le nouvel opérateur de mutation est plus déterministe que le précédent, on peut considérer que le taux de renouvellement de la population (90%) ne doit pas diminuer : cette valeur (qui correspond déjà à une valeur maximale envisagée) sera donc conservée. Les autres paramètres de l’algorithme génétique sont conservés (150 générations maximales de 150 individus et taux de sharing à 90%).

La figure 9.1 donne ainsi la moyenne des pénalités obtenues pour différentes valeurs de τ_{opt} et du taux de croisement, à taux de renouvellement constant de 90% (chaque point correspond à une moyenne sur 20 résolutions de la situation) :

- Dans les deux situations, la valeur optimale du taux de mutation déterministe se situe aux environs de $\tau_{opt} = 0.2$, pour un taux de croisement de 0.35 (soit un taux de mutation de 0.55) : le taux de croisement optimal est donc diminué, ce qui peut s’expliquer par l’augmentation d’efficacité de l’opérateur de mutation.
- Les valeurs supérieures du taux de mutation déterministe (quelque soit le taux de croisement) dégradent visiblement la convergence de l’algorithme génétique : le niveau de déterminisme de l’opérateur de mutation devient trop élevé dans ces configurations et la population se fait piéger trop rapidement dans un optimum local.

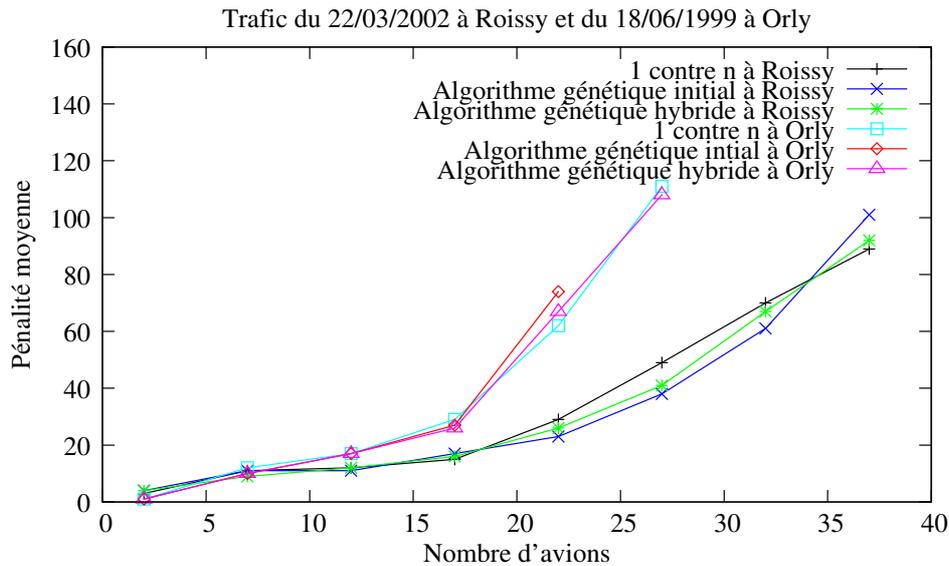


FIG. 9.2 – Pénalité moyenne avec mutation déterministe

Simulations

L'algorithme génétique hybridé est appliqué à la simulation, avec les nouveaux taux de croisement et de mutation issus des résultats de la partie précédente et avec les mêmes paramètres de simulation qu'au chapitre précédent (toujours avec un processeur à 1,6 GHz). Les résultats sont comparés à ceux obtenus avec la méthode 1 contre n seule et avec l'algorithme génétique initial (non hybridé) :

Aéroport	Résolution	Temps de calcul	Pénalité moyenne (sec.)	Cas d'échec
Roissy	1 contre n	6h.15	111	Aucun
	AG initial	8h.30	100	
	AG hybride	7h.30	105	
Orly	1 contre n	1h45	41	Aucun
	AG initial	2h15	37	
	AG hybride	2h00	36	

La figure 9.2 donne la corrélation entre le nombre d'avions impliqués dans les situations et la pénalité moyenne résultante, comparée aux mêmes courbes obtenues avec la méthode 1 contre n seule et avec l'algorithme génétique initial. On constate que cette forme d'hybridation de l'algorithme génétique aboutit à des solutions de pénalités intermédiaires à celles des méthodes initiales :

- Dans les périodes les plus chargées, la méthode hybride fournit de meilleures solutions que l'algorithme génétique initial, mais ces solutions restent plus pénalisantes que celles obtenues avec la méthode 1 contre n seule : dans ces situations, l'opérateur déterministe améliore l'algorithme génétique mais cette amélioration n'est pas suffisante pour trouver de meilleures solutions que la méthode déterministe.
- Dans les périodes moyennement chargées, la méthode hybride trouve de moins bonnes solutions que l'algorithme génétique initial : dans ces situations, l'opérateur déterministe nuit à l'algorithme génétique en orientant trop souvent la population vers un optimum local.

Globalement, cette forme d'hybridation est donc peu satisfaisante, dans la mesure où elle n'apporte pas de meilleures solutions que celles qui seraient obtenues en appliquant parallèlement les

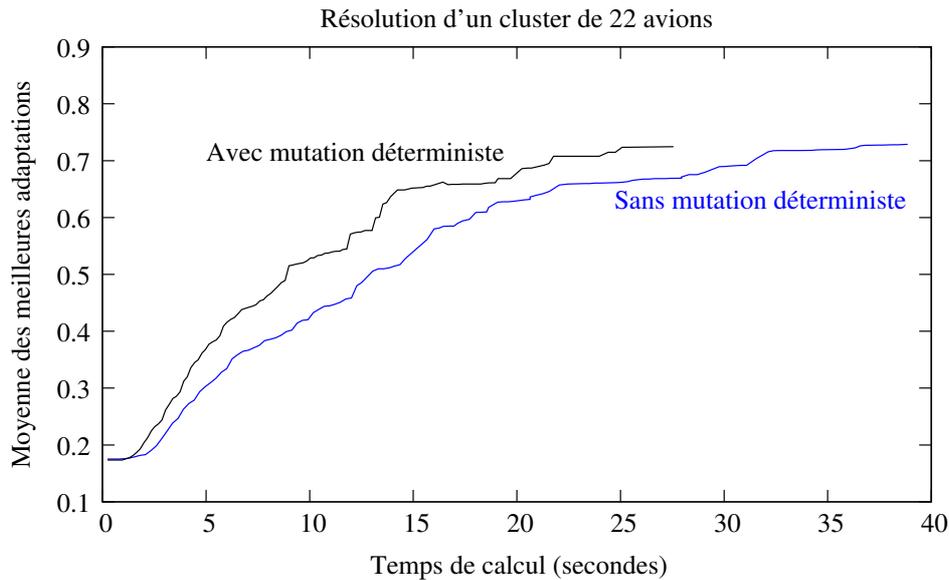


FIG. 9.3 – Vitesse de convergence avec l'opérateur déterministe

deux méthodes initiales (et en choisissant la meilleure d'entre elles). La seule véritable amélioration apportée par l'hybridation concerne la vitesse de convergence, qui est significativement augmentée, comme cela est illustré figure 9.3.

9.4 Algorithme génétique pour la résolution 1 contre n

9.4.1 Objectifs

Une seconde technique d'hybridation consiste à concevoir un nouvel algorithme génétique, qui définit les sous-espaces sur lesquels la méthode de résolution 1 contre n peut être appliquée : dans ce contexte, l'algorithme recherche la meilleure association d'un classement des avions avec une affectation de chemins.

Le problème n'est donc plus envisagé dans sa forme la plus générale, mais le domaine d'application de la méthode 1 contre n est toutefois étendu, puisque le chemin affecté à un avion ne correspond pas nécessairement au chemin optimal qu'il pourrait suivre en ne considérant que son rang dans le classement : les avions les plus prioritaires peuvent être contraints d'effectuer des détours, ce qui signifie que le classement est tempéré par l'affectation de chemins.

9.4.2 Définition de la population

Codage des données

Dans cette nouvelle modélisation du problème, la trajectoire d'un avion est décrite par le chemin qu'il doit suivre et le niveau de priorité qui lui est attribué par rapport aux autres avions. Un individu de la population sera donc décrit par $2N$ variables $(\nu_i, \pi_i)_{1 \leq i \leq N}$ pour un problème N avions :

- $\nu_i \in \{1, \dots, \eta_i\}$ est l'indice du chemin attribué à l'avion i ;
- $\pi_i \in \{1, \dots, N\}$ est son ordre de priorité.

Les variables $(\pi_i)_{1 \leq i \leq N}$ sont supposées définir une relation d'ordre totale sur les N avions et seront donc contraintes de représenter une permutation :

$$\forall (i, j) \in \{1, \dots, N\}^2, i \neq j \implies \pi_i \neq \pi_j$$

Dans la suite, ν et π désigneront respectivement l'ensemble des variables (ν_i) et (π_i) .

Génération initiale

Les pistes étant les ressources les plus pénalisantes de l'aéroport, il est naturel de chercher un classement des avions qui soit *proche* de celui donné par les heures prévues à la piste (c'est l'utilisation de ce classement qui permet à la méthode 1 contre n d'être relativement efficace dans les périodes les plus chargées). Il semble également préférable de commencer par examiner les chemins les plus directs pour chaque avion : en conséquence, la génération initiale de l'algorithme génétique sera formée à partir des heures prévues à la piste et des chemins les plus courts. Chaque individu est créé comme suit :

- Un bruit gaussien d'amplitude maximale H_p est appliqué aux heures prévues à la piste et les avions sont classés en fonction des heures obtenues : les variables $(\pi_i)_{1 \leq i \leq N}$ représentent ce classement.
- Les variables $(\nu_i)_{1 \leq i \leq N}$ sont choisies aléatoirement et de manière non homogène : les valeurs les plus faibles sont le plus souvent tirées, comme pour l'algorithme génétique initial.

9.4.3 Évaluation des individus

Critère d'optimisation

Comme précédemment, un individu de la population ne représente pas forcément une solution admissible : le classement et les chemins affectés peuvent aboutir à une situation sans solution pour un avion. Le critère doit donc prendre en compte les violations de contraintes, qui ne seront plus représentées directement par des conflits entre avions, mais par des cas d'échec de la résolution d'un avion. Dans ce cadre, l'évaluation nécessite :

1. **Le classement des avions**, en fonction de leurs niveaux de priorités $(\pi_i)_{1 \leq i \leq N}$;
2. **Une application de la méthode 1 contre n** sur ce classement et restreinte à un seul chemin par avion (ces chemins sont donnés par les variables $(\nu_i)_{1 \leq i \leq N}$). Cette application s'effectue par itérations de l'algorithme de *Branch & Bound* BB_chemin (algorithme III.6). Lorsqu'aucune solution ne peut être trouvée pour un avion d'indice i , le dernier avion conflictuel rencontré lors de l'exploration des possibilités est identifié comme responsable de l'échec : si j désigne son indice, on notera $c_{ij} = c_{ji} = 1$.

Le critère F à minimiser s'exprime alors comme suit :

- Si tous les avions ont pu être résolus, le critère correspond à la pénalité cumulée des avions :

$$F(\nu, \pi) = \mathcal{D}(\nu, \pi) + \alpha \mathcal{L}(\nu)$$

- Sinon,

$$F(\nu, \pi) = K \sum_{i < j} c_{ij}$$

Où K caractérise les instanciations non admissibles des variables ($K \gg NH_p$) et les autres notations sont celles définies au chapitre 5 :

- $\mathcal{D}(\nu, \pi) = \sum_{i=1}^N d(\nu_i, \pi_i)$ est l'attente cumulée des avions.
- $\mathcal{L}(\nu) = \sum_{i=1}^N l_i(\nu_i)$ est le cumul des temps de roulage supplémentaires dus aux choix des chemins.

Complexité de l'évaluation

L'évaluation de chaque individu est obtenue par l'application de la méthode 1 contre n restreinte à un seul chemin par avion. Le graphe des possibilités pour chaque avion possède donc $\mathcal{O}(\delta_v H_p^3)$ nœuds (cf. partie 7.2.3). L'opération la plus pénalisante à chaque nœud exploré est la détection de conflits avec les autres avions déjà résolus, soient $\mathcal{O}(N\delta_v H_p)$ opérations (cf. partie 4.3.4) :

L'évaluation d'un individu nécessite $\mathcal{O}(\delta_v^2 N^2 H_p^4)$ opérations.

La phase d'évaluation de l'algorithme génétique sera donc beaucoup plus coûteuse qu'au chapitre précédent.

Fonction d'adaptation

La fonction d'adaptation f est définie à valeurs dans $[0, 1]$ de manière inversement proportionnelle au critère, avec un palier fixe à $\frac{1}{2}$ distinguant les individus admissibles des autres :

- Si $F(\nu, \pi) < K$, l'individu représente une solution admissible :

$$f(\nu, \pi) = \frac{1}{2} + \frac{K}{2K + F(\nu, \pi)} \geq \frac{1}{2}$$

- Sinon,

$$f(\nu, \pi) = \frac{K}{K + F(\nu, \pi)} < \frac{1}{2}$$

9.4.4 Caractère partiellement séparable du problème

Fonctions d'adaptation partielles

Le critère à minimiser reste partiellement séparable. Les N fonctions d'adaptation partielles $(G_i)_{1 \leq i \leq N}$ correspondent à la participation des variables (ν_i, π_i) du $i^{\text{ème}}$ avion au critère F :

$\forall i \in \{1, \dots, N\}$:

- Si (ν, π) représente une solution admissible ($\forall j \neq i, c_{ij} = 0$) :

$$G_i(\nu, \pi) = d(\nu_i, \pi_i) + \alpha l(\nu_i)$$

- Sinon,

$$G_i(\nu, \pi) = \frac{K}{2} \sum_{j \neq i} c_{ij}$$

Les individus non admissibles sont caractérisés par l'existence de fonctions d'adaptations partielles $G_i(\nu, \pi) > \frac{K}{2}$. Dans ce cas :

- soit l'avion i n'a pu être résolu,
- soit il gêne la résolution d'autres avions moins prioritaires.

Opérateurs de variation adaptés

Les opérateurs de variation modifient plus souvent les variables ayant de mauvaises adaptations partielles. Les variables (π_i) définissant le classement des avions sont toujours modifiées par paires, ce qui revient à n'effectuer que des transpositions élémentaires sur le classement d'un parent :

La mutation d'un individu A modifie les variables de deux avions i et j ayant de mauvaises adaptations partielles. Le nouvel individu A' est défini par l'inversion des niveaux de priorité des deux avions et par le choix de deux nouveaux chemins :

- $\pi'_i = \pi_j$ et $\pi'_j = \pi_i$;
- ν'_i et ν'_j sont choisis aléatoirement dans leur domaine de définition.

Le croisement de deux individus A et B donne deux individus A' et B' , où A' (respectivement B') est une copie de A (resp B) dans laquelle les variables d'un avion i ayant une mauvaise adaptation partielle prennent les valeurs apportées par l'autre parent. Lorsque le niveau de priorité hérité π_i^B par l'avion i dans A' est modifié ($\pi_i^B \neq \pi_i^A$), l'avion ayant le niveau de priorité π_i^B dans A est également modifié : il prend l'ancienne priorité de i à savoir π_i^A (il en est de même réciproquement pour B').

Ces opérateurs possèdent les mêmes caractéristiques que ceux définis au chapitre précédent :

- Ils ne dépendent d'aucun paramètre.
- Leur niveau de déterminisme ne dépend que des différences entre les fonctions d'adaptation partielles d'un individu : plus elles sont grandes, plus l'opérateur est déterministe (car le choix des variables modifiées est moins aléatoire). Lorsque la population s'oriente vers le domaine admissible, les adaptations partielles se resserent et les opérateurs deviennent de plus en plus aléatoires.

9.4.5 Sharing

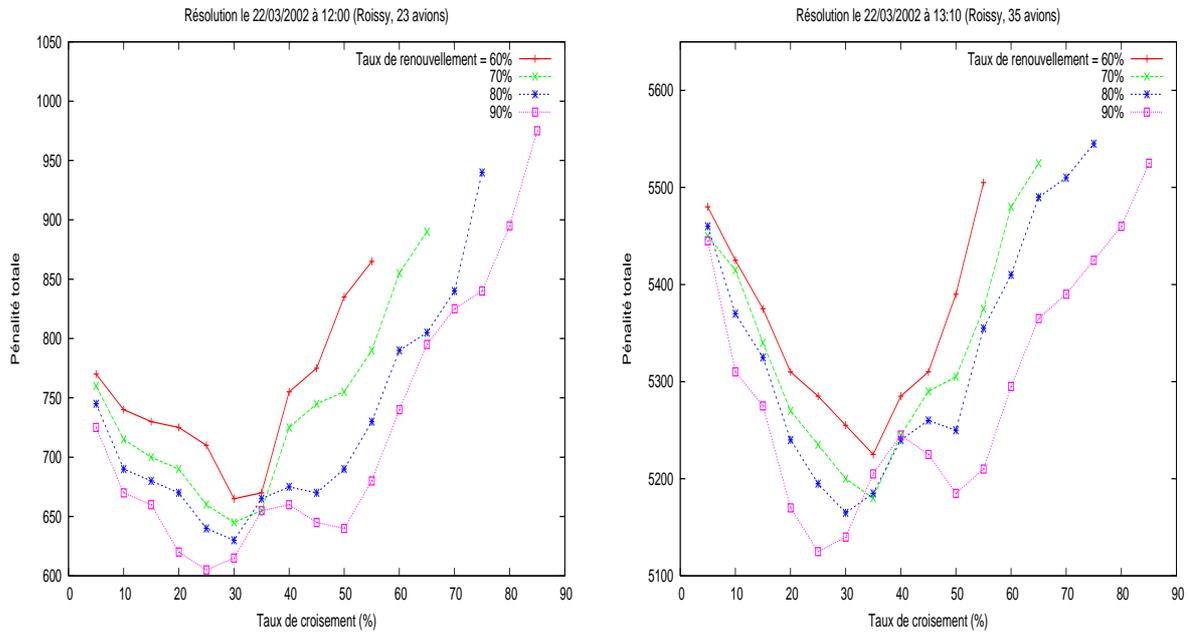
Le déterminisme important des opérateurs de variation lors des premières générations de l'algorithme doit être associé à une technique de sharing pour assurer le maintien de la diversité de la population. Comme précédemment, le sharing clusterisé est utilisé. Il nécessite la définition d'une distance d et du barycentre pondéré entre deux individus A et B :

- La distance n'est calculée qu'en fonction des différences de classement :

$$d(A, B) = \frac{\sum_{i=1}^N |\pi_i^A - \pi_i^B|}{N^2}$$

Cette définition correspond à une simplification de la notion de distance (la distance entre deux individus différents peut être nulle) mais permet de ne considérer que le classement des avions, qui est déterminant sur la solution apportée.

- Le barycentre de deux éléments A et B pondéré par les coefficients α_A et α_B est un individu C défini à partir des barycentres entiers (π_i) entre les niveaux de priorité (π_i^A) et (π_i^B) des avions. Les niveaux de priorités résultants (π_i^C) sont obtenus après réajustement des variables (π_i) , pour qu'elles représentent une permutation des N avions. La définition des distances ne faisant pas intervenir les chemins, les variables (ν_i^C) du nouvel individu formé n'ont aucune influence sur l'algorithme.

FIG. 9.4 – Influence du taux de croisement de l'AG 1 contre n

9.4.6 Étalonnage

Taux de croisement et de mutation

L'algorithme génétique hybride avec résolution 1 contre n subit le même étalonnage que celui défini au chapitre précédent : la figure 9.4 donne les moyennes des pénalités obtenues sur 20 résolutions des deux situations à Roissy, dans différentes configurations du taux de croisement et de mutation (la résolution de ces situations fait intervenir plusieurs clusters de différentes tailles), avec un nombre maximal de 100 générations de 100 individus :

- Les pénalités minimales sont obtenues avec un taux de renouvellement de 90% et un taux de croisement de 25% (soit un taux de mutation de 65%). Ces taux confirment l'importance de la diversification de la population, en rapport avec les opérateurs de variation adaptés au caractère partiellement séparable du problème.
- Les deux opérateurs sont utiles à la convergence (les valeurs extrêmes du taux de croisement aboutissent à de moins bonnes solutions), bien que l'opérateur de mutation semble là encore le plus important.

Taille de la population et nombre de générations

La figure 9.5 donne l'évolution de la moyenne des meilleures adaptations en fonction du nombre d'individus évalués (donné par le produit du nombre de générations et de la taille de la population). Ces moyennes sont obtenues sur 20 résolutions d'un cluster de 22 avions extrait de la situation à 13h10 à Roissy, pour différentes tailles de population :

- On constate que l'algorithme génétique hybride trouve très rapidement des solutions admissibles, caractérisées par des individus dont l'adaptation dépasse $\frac{1}{2}$: de tels individus apparaissent dans la population dès les premières générations.

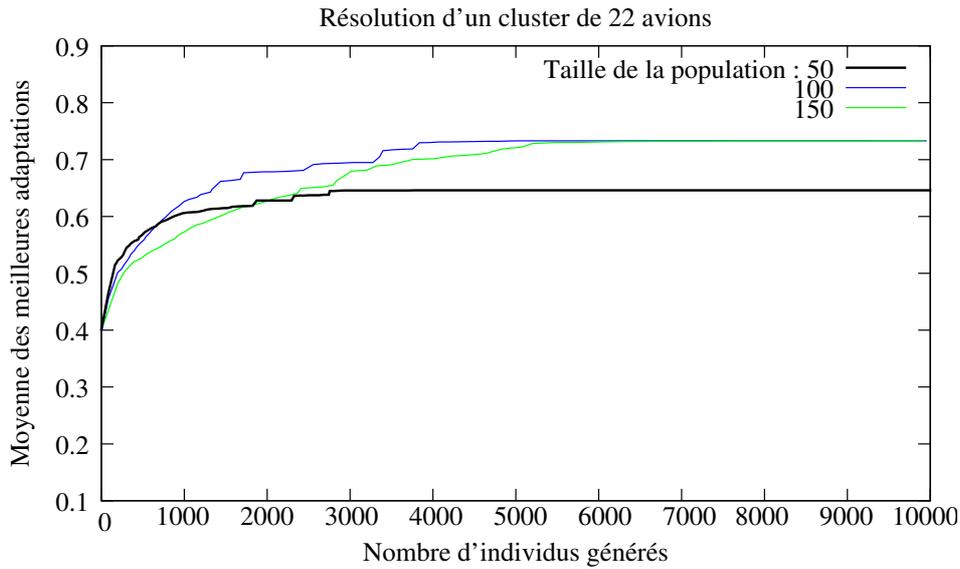


FIG. 9.5 – Influence de la taille de la population sur la convergence

- La convergence la plus rapide (en temps de calcul) est obtenue avec une population de 100 individus et nécessite environ 50 générations : par la suite, l'algorithme génétique sera appliqué avec cette taille de population et avec un nombre maximum de 100 générations.

9.4.7 Simulations

Le tableau suivant donne les résultats globaux de la méthode hybride ainsi définie, appelée *AG 1 contre n* , appliquée dans les mêmes conditions qu'au chapitre précédent (toujours avec un processeur à 1,6 GHz) :

Aéroport	Résolution	Temps de calcul	Pénalité moyenne (sec.)	Cas d'échec
Roissy	1 contre n	6h.15	111	Aucun
	AG initial	8h.30	100	
	AG 1 contre n	18h.00	76	
Orly	1 contre n	1h45	41	Aucun
	AG initial	2h15	37	
	AG 1 contre n	6h00	35	

Deux points importants peuvent être remarqués :

- Les temps de calcul de la simulation sont fortement augmentés avec l'algorithme génétique hybride (ceci est dû à la complexité de l'évaluation des individus). Le temps de résolution des situations les plus chargées devient presque prohibitif pour une application opérationnelle à Roissy, car il dépasse le pas de rafraîchissement ($\Delta = 2$ minutes) du système. Ce temps de résolution pourra cependant être facilement réduit (sans dégrader la résolution) en diminuant l'horizon de prédiction, comme cela sera étudié dans la partie 10.2.
- La pénalité moyenne est diminuée de manière non négligeable à Roissy : les solutions trouvées sont de meilleure qualité. A Orly, la pénalité résultante reste équivalente avec les trois méthodes de résolution.

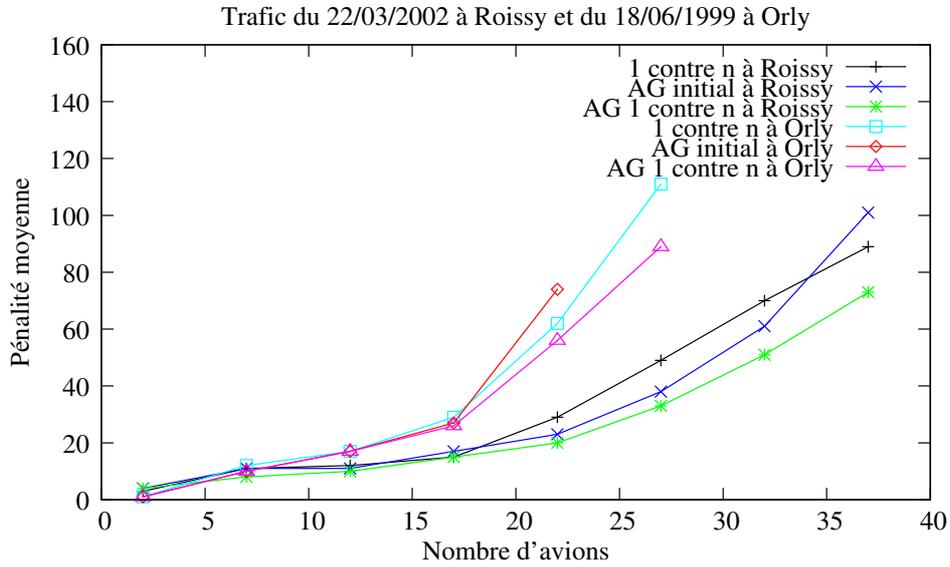


FIG. 9.6 – Pénalité moyenne en fonction du nombre d'avions pour l'AG 1 contre n

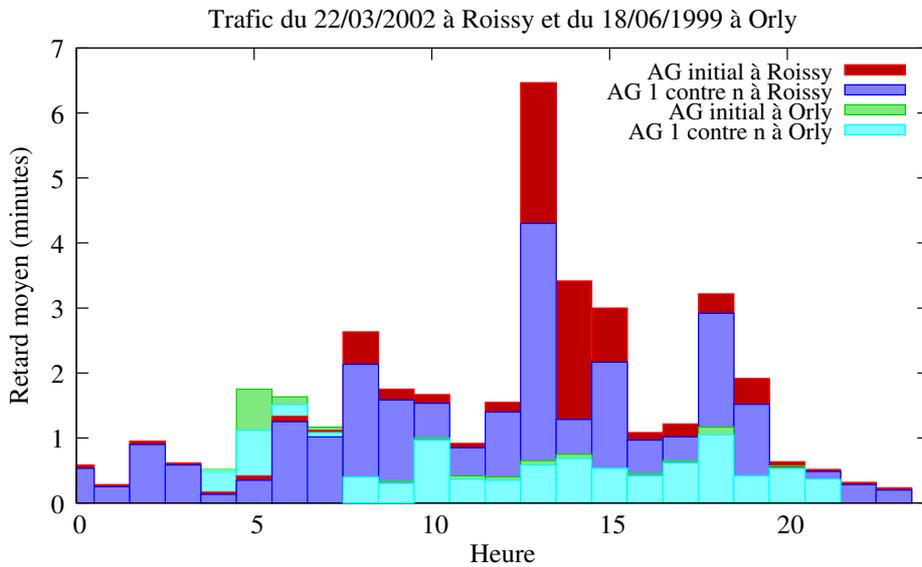


FIG. 9.7 – Retard moyen avec l'AG 1 contre n

Les figures 9.6 et 9.7 donnent les résultats détaillés des simulations :

- Les résolutions sont considérablement améliorées dans les périodes les plus chargées, où les pénalités moyennes obtenues par l’algorithme génétique hybride sont nettement inférieures à celles de la résolution 1 contre n .
- L’effet peut être constaté sur les retards moyens, qui sont diminués de plus deux minutes dans les périodes les plus chargées.

9.5 Conclusion

Dans ce chapitre, deux méthodes hybrides utilisent la même méthode de résolution déterministe 1 contre n au sein d’un algorithme génétique.

La première technique permet d’accélérer la convergence de l’algorithme génétique initial et en améliore les solutions pour les situations les plus chargées. Elle se révèle pourtant inefficace dans la mesure où les résultats obtenus ne sont pas meilleurs que si les deux méthodes de résolution étaient utilisées séparément et que la meilleure solution obtenue était sélectionnée.

Les résultats de la seconde hybridation sont beaucoup plus prometteurs : la pénalité est significativement diminuée par rapport aux deux méthodes séparées, même et surtout dans les périodes les plus chargées. Deux raisons permettent d’expliquer l’amélioration des résolutions :

- L’espace de recherche est une restriction de celui exploré par l’algorithme génétique initial, mais également un élargissement de celui considéré pour la méthode 1 contre n : l’affectation des chemins n’est pas fonction du classement des avions, ce qui permet de tempérer son effet.
- L’algorithme génétique explore les possibilités de classements des avions d’une façon suffisamment aléatoire pour trouver rapidement des combinaisons que la méthode systématique ne pourrait trouver qu’au bout d’un temps prohibitif.

Dans la configuration de simulation étudiée (horizon de prédiction relativement grand par rapport aux incertitudes sur les vitesses), le temps de résolution de l’algorithme génétique avec classement est par contre beaucoup plus coûteux que celui des deux méthodes initiales (certaines résolutions peuvent dépasser la pas de rafraîchissement du système initialement prévu). Les résultats obtenus établissent cependant un phénomène important pour le trafic au roulage sur un aéroport : les situations de trafic se prêtent particulièrement bien à la résolution par ordre de priorité, à condition que ces priorités soient suffisamment flexibles d’une résolution à la suivante.

Quatrième partie

Applications

Chapitre 10

Mesure de différents facteurs

10.1 Introduction

Le simulateur de trafic aéroportuaire, muni des méthodes de résolution définies dans les chapitres précédents, permet d'étudier divers aspects opérationnels de la gestion du trafic au sol :

- Le premier point concerne l'adaptation de l'horizon de prédiction aux incertitudes sur la vitesse de déplacement des avions. Lorsque ces incertitudes augmentent, l'horizon de prédiction doit être réduit, pour éviter que certaines trajectoires encore imprécises entraînent des attentes prématurées ou des détours s'avérant par la suite inutiles : la simulation d'une journée de trafic avec différentes tailles de l'horizon de prédiction permet de quantifier ces effets, qui restent difficilement mesurables autrement.
- Un deuxième point important est relatif aux sens préférentiels définis sur les voies de circulation de l'aéroport : s'il est évident que cette répartition logique des mouvements sur des axes prédéfinis permet de simplifier considérablement les situations de trafic, il n'en est pas moins prévisible que le cheminement de certains avions en soit augmenté et que des points bloquants indésirables puissent apparaître. La simulation permet d'en mesurer les effets, en envisageant plusieurs scénarii, dans lesquels les sens préférentiels sont plus ou moins marqués.
- L'impact des incertitudes sur la qualité des résolutions doit également être étudié : les projets de guidage automatique des avions au sol permettent d'envisager à plus ou moins long terme des systèmes où ces incertitudes seraient fortement diminuées. La nature et le coût des systèmes envisagés dépendent des gains attendus pour les compagnies et le système de contrôle : là encore, le gain relatif aux temps de roulage peut être estimé par simulation, avec différentes échelles d'incertitudes.

L'ensemble des paramètres énumérés ci-dessus demanderait à être complété, en fonction d'autres interrogations que peuvent se poser les opérationnels. Il faut cependant remarquer que le nombre de configurations possibles dans lesquelles une simulation peut s'effectuer devient rapidement combinatoire : dans ce chapitre, l'effet de chaque paramètre sera étudié séparément. Dans la mesure où les simulations effectuées à Orly aboutissent aux mêmes conclusions que celles effectuées à Roissy, seuls les résultats obtenus sur Roissy sont présentés dans ce chapitre.

10.2 Horizon de prédiction

10.2.1 Simulations

L'influence de l'horizon de prédiction est étudiée par simulations d'une même journée de trafic (le 22/03/2002) à Roissy, avec différentes valeurs de H_p (ce paramètre est défini dans la partie 4.2.1). Chaque méthode de résolution développée peut être plus ou moins sensible à la variation de ce paramètre et il n'est pas impossible, a priori, que l'efficacité relative de ces méthodes soit inversée dans certaines configurations : chaque valeur de H_p envisagée donnera lieu à trois simulations, relatives aux trois méthodes de résolutions développées :

- **La méthode 1** contre n avec recherche de classement optimal, définie au chapitre 7 ;
- **L'algorithme génétique** défini au chapitre 8, qui sera noté *AG* ;
- **La méthode hybride** définie au chapitre 9, qui sera notée *AG 1 contre n*.

Les simulations effectuées précédemment (lors de la mise en œuvre des méthodes de résolution) s'appliquent à une taille de l'horizon de prédiction ($H_p = 10$ minutes) qui peut être considérée comme maximale, par rapport aux incertitudes sur les vitesses de déplacement des avions ($\delta_v = 20\%$) et aux vitesses maximales de roulage ($V_{\max} = 10$ m/s) : au delà, la taille des zones d'incertitude des avions pourraient atteindre des valeurs qui ne seraient pas raisonnablement adaptées à l'optimisation des trajectoires :

$$l_{\max} = 600 \times 20\% \times 10 = 1200 \text{ mètres}$$

Dans ce contexte, les simulations se feront à taux d'incertitude fixe, pour les quatre valeurs de H_p suivantes : 4, 5, 7 et 10 minutes. Seul le nombre de positions d'attente par avion, envisagées à chaque résolution, suivra les variations de l'horizon de prédiction. Les autres paramètres seront fixés :

Paramètres		Valeur			
Horizon de prédiction (minutes)	H_p	4	5	7	10
Horizon étendu (minutes)	H_e	2			
Pas de rafraîchissement (minutes)	Δ	2			
Taux d'incertitude sur les vitesses	δ_v	20%			
Nombre de positions d'attente	Ω	1	1	2	2
Sens préférentiels		Oui			
Contraintes relatives à l'effet de l'horizon		Oui			

10.2.2 Méthode de mesure

Les variations de l'horizon de prédiction modifient les ordres de grandeurs du nombre d'avions et de la pénalité cumulée de chaque situation. Pour comparer de manière équitable l'efficacité des méthodes de résolution dans chaque configuration, les mesures doivent donc porter sur les trajectoires finales des avions dans la simulation et non sur les solutions trouvées à chaque étape. Dans ce cadre, les mesures en sortie de la simulation sont :

- Le *retard* assigné à chaque mouvement, cumulant (sans pondération) les attentes imposées à l'avion et le temps de roulage supplémentaire dû au choix du chemin.
- La corrélation entre le nombre de vols actifs dans chaque période de dix minutes de la journée et la moyenne de leurs *retards*, qui représente la sensibilité de la méthode de résolution à la densité de trafic.

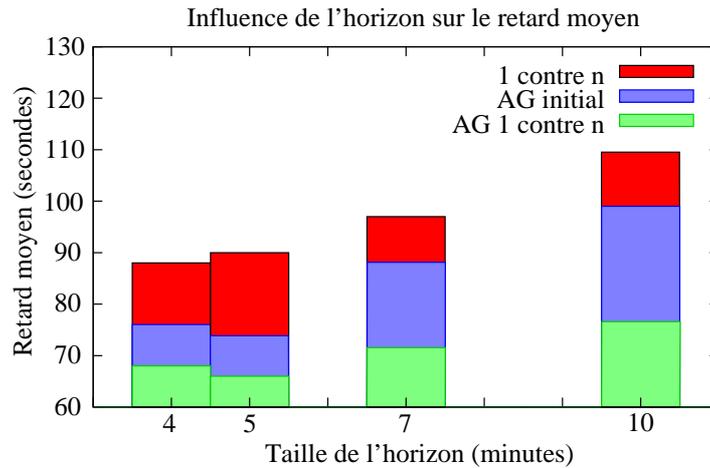


FIG. 10.1 – Influence de l'horizon sur la moyenne des retards

10.2.3 Résultats

Le retard moyen engendré par les 12 simulations est donné par la figure 10.1 :

- Aucun cas d'échec de la résolution n'est à signaler, même avec l'horizon de prédiction le plus réduit : ceci montre que les contraintes relatives à l'effet de l'horizon permettent d'anticiper correctement les principales situations à éviter à chaque résolution, pour que les résolutions suivantes restent possibles.
- D'une manière générale, la taille de l'horizon de prédiction influe de manière non négligeable sur les retards imputés aux mouvements. Paradoxalement, ces retards diminuent lorsque la taille de l'horizon se réduit.

Les figures 10.2, 10.3 et 10.4 montrent plus précisément la sensibilité de chaque méthode de résolution à la densité de trafic et à l'horizon de prédiction. La valeur $H_p = 5$ minutes semble optimale pour toutes les méthodes : le retard cumulé sur la journée est sensiblement le même qu'avec l'horizon de 4 minutes, mais il est réparti différemment, car les périodes les plus chargées créent moins de retard. On peut remarquer qu'avec la méthode *AG 1 contre n*, les mêmes effets peuvent être observés mais sont atténués : la résolution est moins sensible à la taille de l'horizon. Deux principales raisons permettent d'expliquer la diminution du retard avec la réduction de l'horizon :

1. La complexité du problème à résoudre est fortement diminuée avec l'horizon réduit : le domaine des variables de décision est restreint et les groupes d'avions en conflits dans chaque situation sont à la fois moins nombreux et moins gros, comme le montre la figure 10.5. Sur ce point, la complexité *pire cas* exprimée dans la partie 5.3.2 (qui permet déjà de quantifier grossièrement la simplification du problème) pourrait être affinée par une mesure de complexité relative à la probabilité de conflits, qui révélerait peut-être l'existence d'un horizon limite (au delà duquel la complexité augmenterait plus fortement) en fonction de paramètres tels que la taille de l'aéroport, la vitesse des avions et le nombre de pistes : des études ultérieures pourront ainsi porter sur les *transitions de phases* de la complexité *stochastique* [Hogg 96b, Hogg 96a], qui permettraient certainement d'appréhender plus rigoureusement ce phénomène.
2. L'horizon de prédiction de 10 minutes peut également biaiser le critère d'optimisation, suite aux incertitudes importantes sur les vitesses de déplacement : dans ce cas, des manœuvres préventives correspondraient trop souvent à l'optimum, alors qu'elles s'avèrent inutiles par la

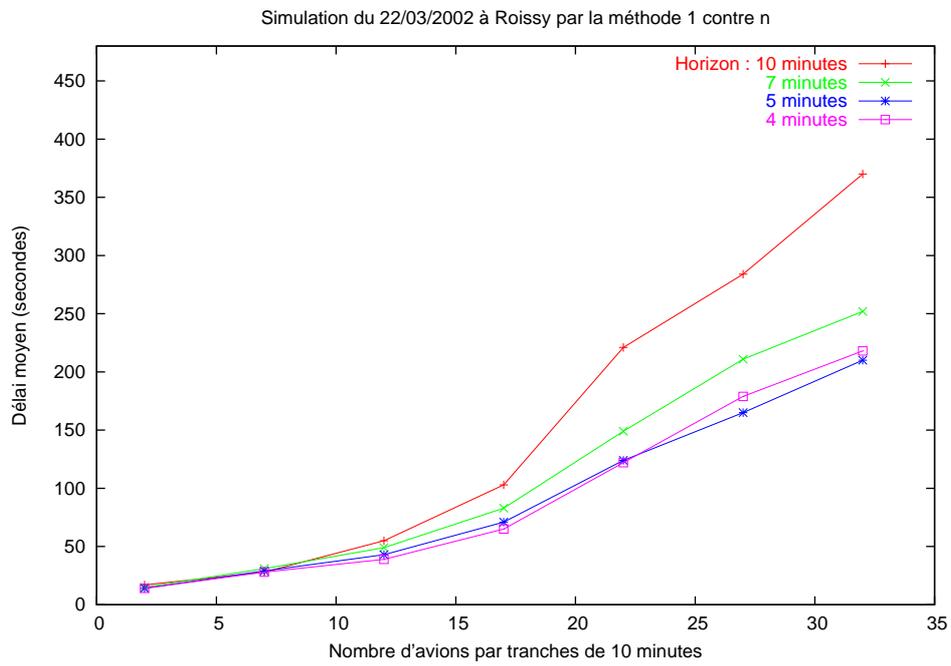
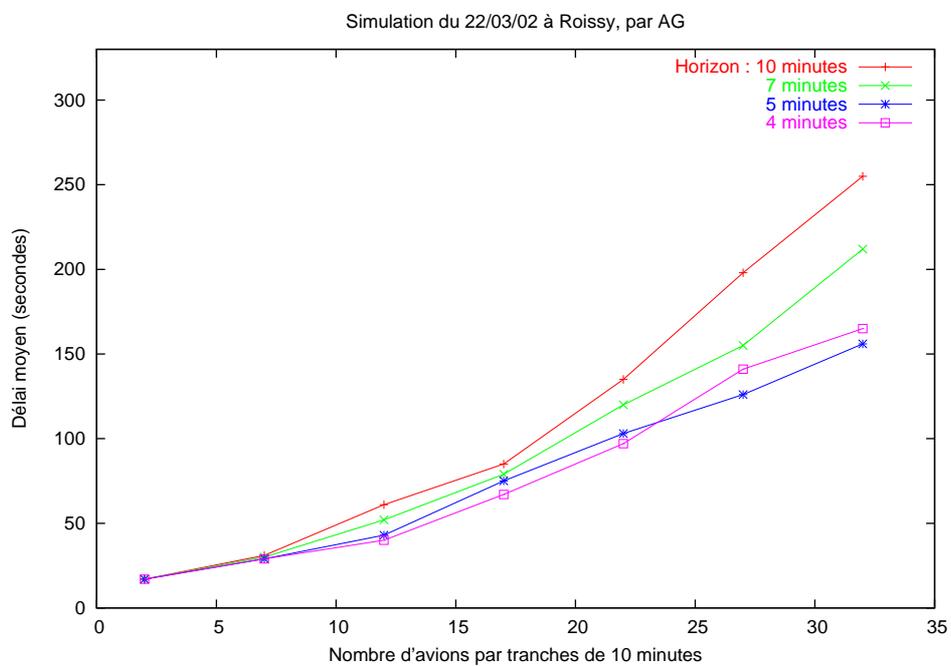
FIG. 10.2 – Influence de l'horizon pour la méthode 1 contre n 

FIG. 10.3 – Influence de l'horizon pour l'AG

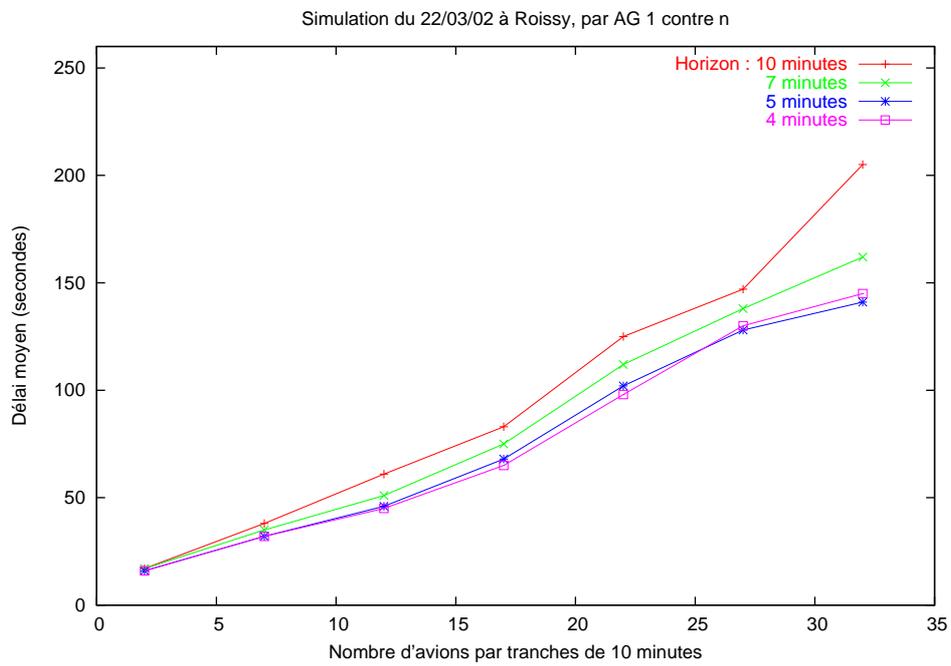


FIG. 10.4 – Influence de l’horizon pour l’AG 1 contre n

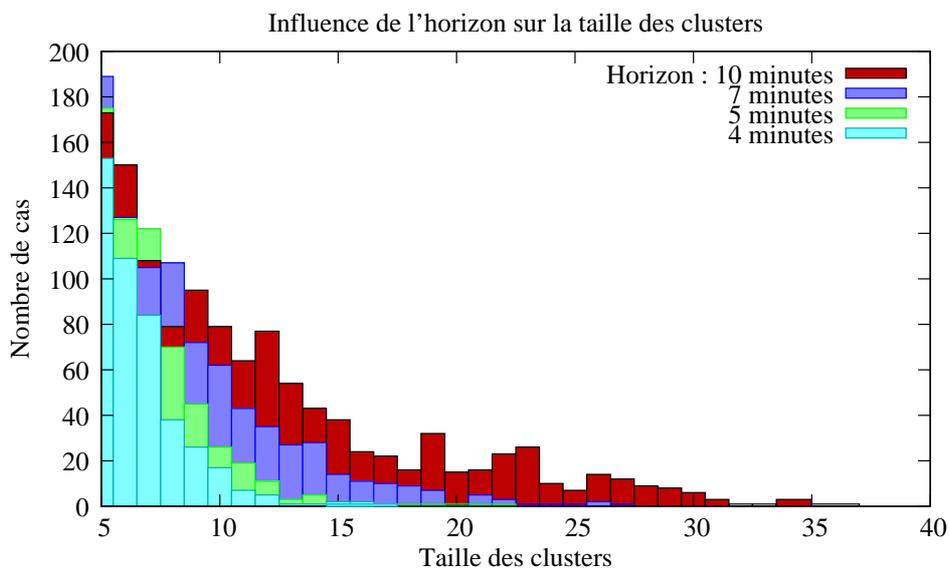


FIG. 10.5 – Influence de l’horizon sur la taille des clusters

suite, une fois que les trajectoires des avions deviennent plus précises. Aux vues des résultats uniformes des trois méthodes de résolution, cette explication semble pertinente : cela signifie que les solutions obtenues avec l'horizon de 10 minutes pourraient être améliorées si le critère pénalisait davantage le retard immédiat des avions. Cette technique nécessiterait cependant une pondération des retards subjective, risquant d'aboutir aux mêmes résultats qu'avec l'horizon réduit, mais en un temps de calcul beaucoup plus contraignant.

En résumé, ces résultats montrent qu'un horizon de prédiction de 5 minutes est suffisant (et même optimal) pour résoudre précisément les situations de trafic au sol, lorsque l'effet de l'horizon est neutralisé par des contraintes appropriées.

10.3 Sens préférentiels de circulation

10.3.1 Simulations

Pour mesurer l'impact des sens préférentiels de circulation sur les temps de roulage, trois configurations de Roissy sont envisagées :

- **Avec sens préférentiels** : les pondérations associées à chaque portion de taxiway (données directement dans la description de l'aéroport) sont appliquées intégralement ;
- **Sans sens préférentiels** : aucune pondération n'est prise en compte. Les chemins proposés aux mouvements ne sont sélectionnés que sur leur temps de parcours estimé ;
- **Avec sens préférentiels restreints** : seuls les principaux sens uniques, le long des grands axes de l'aéroport sont considérés. Pour cela, les pondérations ne sont appliquées que lorsqu'elles sont relatives à des lignes droites dépassant une longueur fixée $l_0 = 500$ mètres.

Là encore, les trois méthodes de résolution sont utilisées et comparées dans chaque configuration, avec un horizon de prédiction de 5 minutes et les mêmes paramètres que dans la partie précédente. Les mesures en sortie de chaque simulation sont :

- le *temps de parcours minimal* de chaque avion, défini par le temps de roulage qu'il aurait sans attente, en suivant le chemin le plus rapide qui lui est proposé (à la vitesse nominale modélisée dans la partie 3.2). Le temps de parcours minimal ne dépend donc pas de la résolution des situations, mais diffère dans chaque configuration.
- le *retard* de chaque avion, mesuré sur sa trajectoire finale, cumulant les attentes et l'éventuel détour imposés à l'avion. Le cumul du temps de parcours minimal et du retard donne le *temps de roulage effectif*, qui permet également de comparer entre elles les trois configurations.

10.3.2 Influence sur les chemins

La figure 10.6 donne les distributions de la réduction du temps de parcours minimal dans chaque configuration envisagée :

- La diminution des temps de parcours minimaux est significative dans les deux nouvelles configurations (sens préférentiels négligés ou restreints) : les chemins proposés aux avions permettent d'économiser directement entre 15 et 20 secondes de roulage en moyenne.
- La configuration intermédiaire (sens préférentiels restreints) aboutit à des temps de parcours minimaux plus proches de ceux de la configuration sans aucun sens préférentiel : elle semble donc avantageuse, dans la mesure où le trafic reste réparti par sens de circulation, sans que les cheminements en soient visiblement affectés.

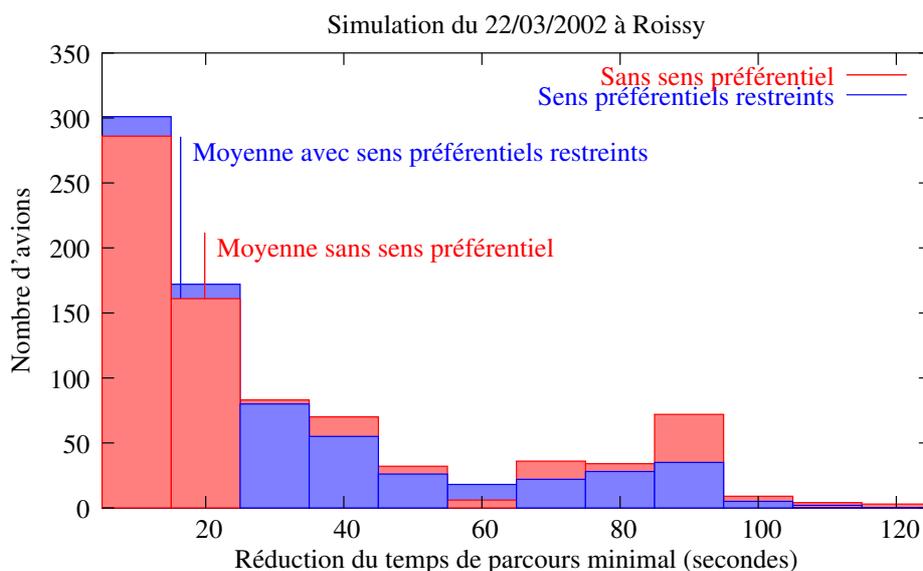


FIG. 10.6 – Influence des sens préférentiels sur les chemins

10.3.3 Influence sur les temps de roulage

Les résultats globaux des 9 simulations effectuées sont présentés sur la figure 10.7 :

- Lorsque les sens préférentiels définis par les opérationnels ne sont pas appliqués, des cas d'échec de la méthode 1 contre n apparaissent : l'analyse des situations non résolues révèle qu'il s'agit de situations où trois avions sont impliqués autour de la même intersection, le premier allant vers le second, le second vers le troisième et le troisième vers le premier (figure 10.8). Une solution existe mais nécessite que chaque avion effectue un détour ou une attente, ce qui ne peut pas être trouvé par la méthode 1 contre n , puisque l'avion le plus prioritaire ne prend pas en compte les deux autres. Cette disposition relative des avions correspond à une généralisation de la notion de *face à face* à plus de deux avions et pourrait être évitée par des contraintes supplémentaires sur les positions des avions à la fin de l'horizon.
- Dans la configuration sans aucun sens préférentiel, le retard est augmenté mais le temps de roulage effectif est diminué. L'augmentation des retards, due à la répartition moins stricte des avions, est donc compensée par la réduction des temps de parcours minimaux. L'amélioration n'est cependant pas suffisante pour rendre la configuration réellement avantageuse d'un point de vue opérationnel, car les temps de roulage deviennent plus variables et les situations beaucoup plus complexes pour un opérateur humain (car présentant de nombreux risques de face-à-face).
- La configuration avec sens préférentiels restreints aboutit à des temps de roulage effectifs du même ordre que lorsque tous les sens préférentiels sont appliqués : la répartition du trafic est stricte sur les grands axes, mais les sens d'entrée et de sortie des zones de stationnement ne sont pas définis, ce qui génère de nombreuses situations problématiques et cause finalement des retards plus importants.
- Dans chaque configuration, les trois méthodes de résolution conservent la même efficacité relative : l'algorithme génétique hybride (AG 1 contre n) fournit toujours les meilleures solutions.

La figure 10.9 montre la sensibilité de la meilleure méthode de résolution (AG 1 contre n) à la densité du trafic, dans les trois configurations étudiées : on remarque que le retard est diminué dans les périodes moyennement chargées (moins de 25 avions actifs par tranche de 10 minutes) lorsque les

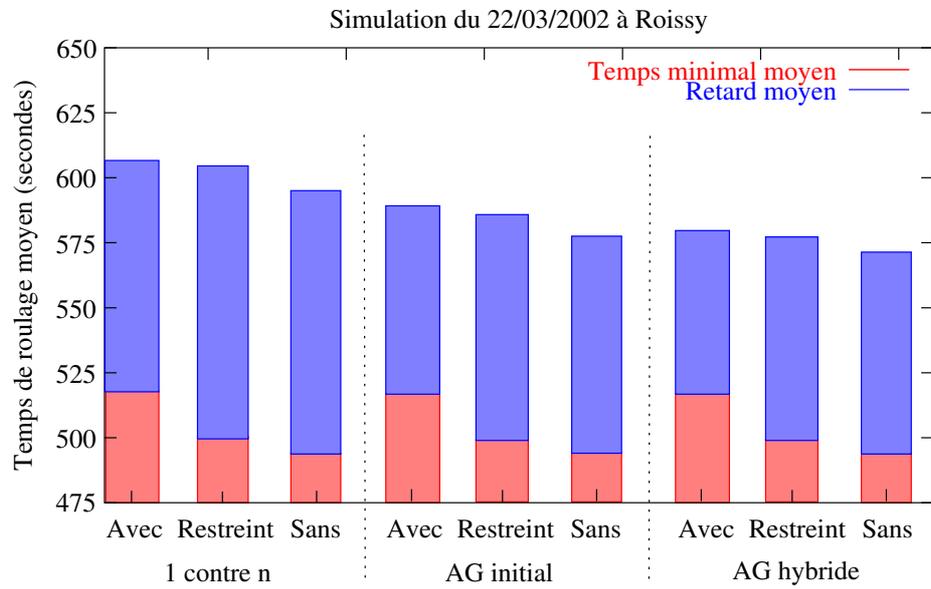


FIG. 10.7 – Influence des sens préférentiels sur les temps de roulage

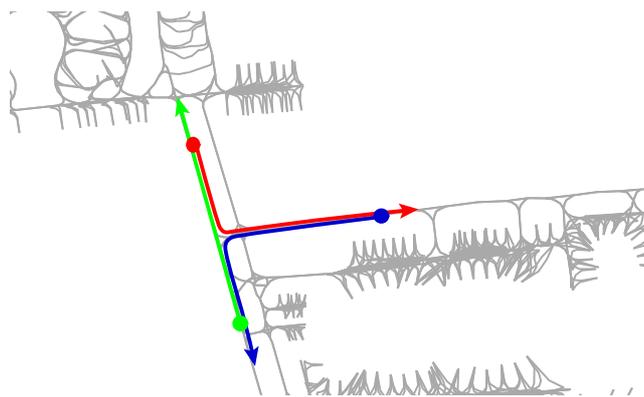


FIG. 10.8 – Exemple de face-à-face à trois

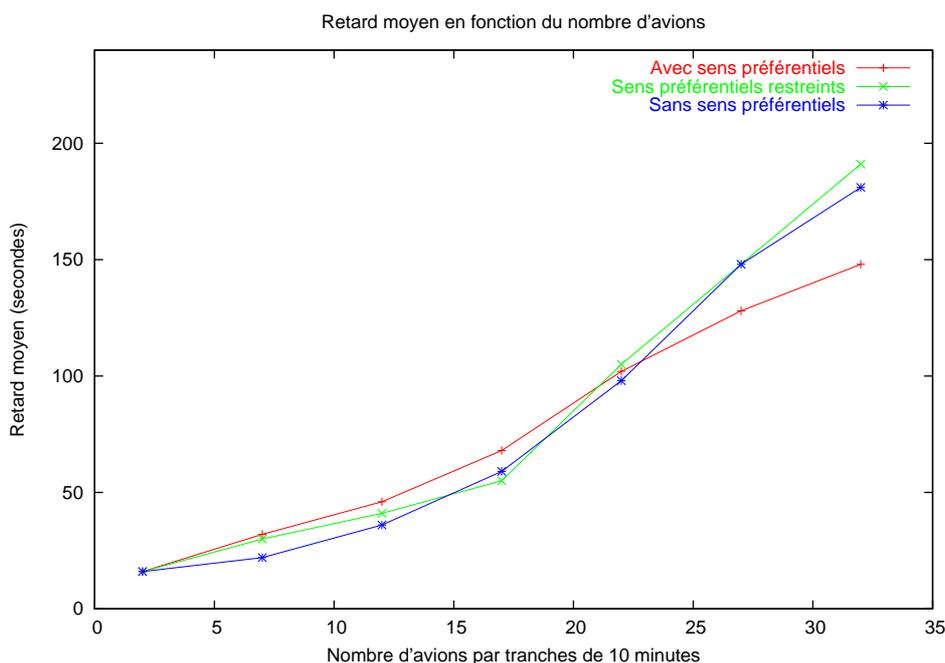


FIG. 10.9 – Influence des sens préférentiels pour l'AG 1 contre n

sens préférentiels sont restreints ou négligés. L'écoulement du trafic peut donc être significativement amélioré en dehors des périodes de pointe, si les sens préférentiels de circulation sont annulés (ce qui corrobore les pratiques opérationnelles).

10.4 Incertitudes sur la vitesse

10.4.1 Problèmes posés

L'étude des vitesses réelles des avions à Roissy et à Orly (partie 3.2) montre que le taux d'incertitude actuel sur les vitesses de roulage se rapproche plutôt de 50% que de 20% (en ligne droite tout au moins) : dans ces conditions, les ordres de contrôle doivent absolument être donnés le plus tard possible, pour rester flexibles et ne pas dégrader excessivement l'optimisation des trajectoires.

Ces considérations, appliquées à la simulation ou à tout système de maintien des séparations au roulage, impliquent que les résolutions doivent être améliorées lorsque le pas de rafraîchissement du système diminue, surtout si l'incertitude est importante : les simulations seront donc effectuées avec différentes valeurs du paramètre Δ .

De plus, pour que les variations du taux d'incertitude soient étudiées équitablement, les vitesses moyennes ne doivent pas être modifiées : l'intervalle des vitesses possibles étant défini par rapport à la vitesse maximale en ligne droite (V_{\max}), cette vitesse sera augmentée avec l'incertitude.

Enfin, pour que l'effet des incertitudes ne soit pas sous-estimé, le passage à la situation suivante se fera avec brouillage des positions (comme cela a été décrit dans la partie 4.4.2).

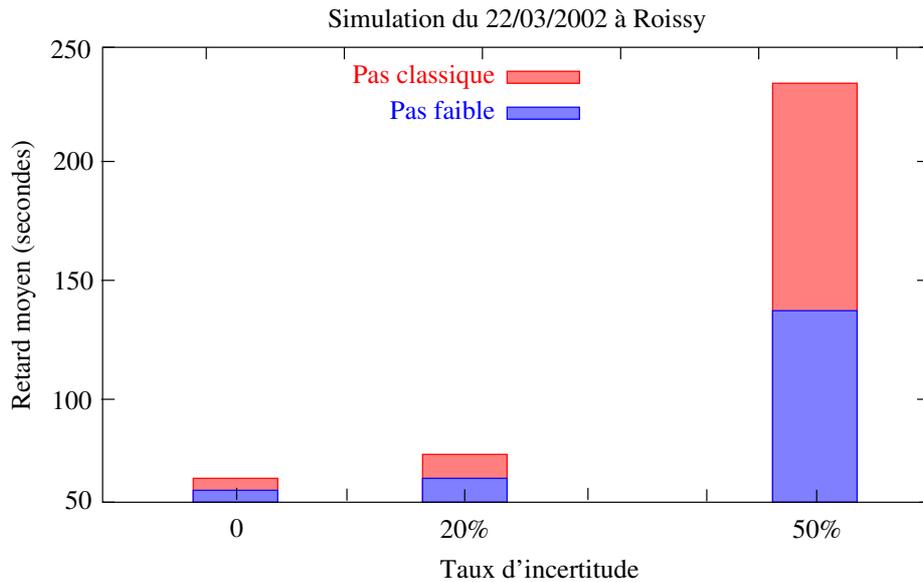


FIG. 10.10 – Influence des incertitudes sur le retard

10.4.2 Simulations

Les simulations sont effectuées à partir du même échantillon de trafic que précédemment, en utilisant la meilleure méthode de résolution (AG hybride 1 contre n) avec les paramètres les plus adaptés issus des résultats précédents :

- **Horizon de prédiction** : $H_p = 5$ minutes ;
- **Horizon étendu** : $H_e = 2$ minutes ;
- **Sens uniques** : appliqués ;
- **Nombre de positions d'attente** : $\Omega = 1$.

Six configurations sont envisagées :

Scénario	δ_v	Δ (minutes)	V_{\max} (m/s)
Incertitude élevée avec pas faible	50 %	1	12
Incertitude élevée avec pas classique	50 %	2	12
Incertitude moyenne avec pas faible	20 %	1	10
Incertitude moyenne avec pas classique	20 %	2	10
Nulle avec pas faible	0 %	1	9
Nulle avec pas classique	0 %	2	9

10.4.3 Résultats

Les résultats globaux des six simulations sont présentés sur la figure 10.10 :

- La diminution du pas de rafraîchissement Δ n'a aucun effet sans incertitude, mais devient de plus en plus bénéfique lorsque le taux d'incertitude augmente : lorsque ce dernier atteint 50%, le retard moyen sur l'ensemble de la journée est divisé par deux par simple réduction de Δ à 1 minute. Ces résultats confirment l'importance de la flexibilité du système de contrôle pour la gestion optimisée des incertitudes.

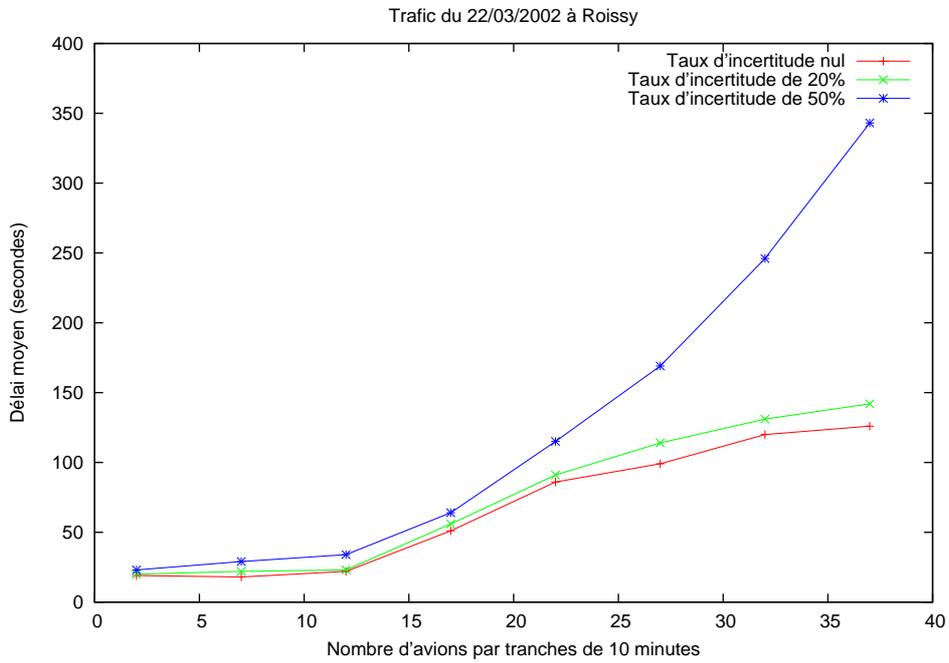


FIG. 10.11 – Influence des incertitudes pour l'AG 1 contre n

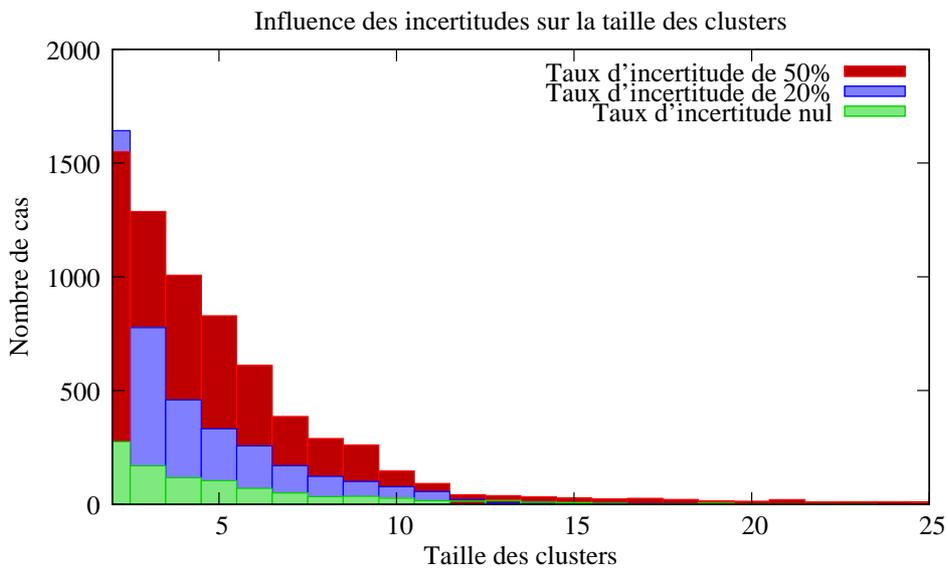


FIG. 10.12 – Influence des incertitudes sur la taille des clusters

- Dans tous les cas, le retard moyen augmente fortement avec l’incertitude : même dans la configuration la plus flexible ($\Delta = 1$ minute) et avec un horizon relativement réduit ($H_p = 5$ minutes), le retard est doublé lorsque le taux d’incertitude passe de 20% à 50%. L’obligation de rouler à des vitesses prédéfinies, même réduites, permettrait donc de diminuer considérablement le retard des avions au sol.

La sensibilité de la méthode de résolution à la densité de trafic et aux incertitudes est représentée sur la figure 10.11, pour les configurations avec pas faible ($\Delta = 1$ minute) : il apparaît que la dégradation du retard moyen est concentrée sur les périodes les plus chargées et très peu visible sur les périodes moyennement chargées (moins de 20 avions par tranche de 10 minutes). Un taux d’incertitude élevé n’est donc pas pénalisant tant que l’aéroport n’atteint pas un certain seuil de saturation.

Enfin, la figure 10.12 montre la croissance considérable de la taille et du nombre de clusters provoquée par l’augmentation des incertitudes (ces mesures sont extraites des résolutions avec le pas faible de 1 minute) : l’augmentation du nombre total de clusters résolus (alors que l’horizon et le pas de rafraîchissement sont fixes) signifie que la résolution itérative par clusters séparés (décrite dans la partie 5.4.3) devient caduque, car ces clusters finissent le plus souvent par être réunis en un seul cluster de grande taille.

10.5 Conclusion

Les simulations d’une journée de trafic aéroportuaire permettent de dégager les principales caractéristiques de la gestion des mouvements au sol :

- L’incertitude sur les vitesses de roulage, même si elle était réduite à un taux de 20%, implique que les trajectoires des avions au sol ne peuvent être véritablement optimisées qu’avec un horizon de prédiction de l’ordre de 5 minutes. Cette limite sur la prévisibilité du trafic au sol justifie la définition de procédures opérationnelles strictes visant à éviter les situations de blocage relatif des avions. Pour un système automatique d’optimisation des trajectoires, cette limite rend obligatoire la considération de contraintes spécifiques assurant la pérenité de l’écoulement du trafic.
- Différentes configurations des sens préférentiels de circulation peuvent être envisagées : leur mise en œuvre induit généralement un rallongement non négligeable des chemins proposés aux avions, mais s’avère bénéfique dans les périodes les plus chargées, dans la mesure où elle rend plus prévisibles les temps de roulage effectifs et assure une répartition plus organisée (donc plus sûre) du trafic. L’application des sens préférentiels s’avère cependant pénalisante en dessous d’un certain seuil d’activité de l’aéroport : des études ultérieures pourront s’intéresser à une définition plus flexible de ces sens préférentiels.
- L’influence des incertitudes liées aux vitesses de roulage sur les retards imposés aux mouvements est considérable : tout système permettant de réduire ces incertitudes peut potentiellement diminuer de moitié les retards au sol, tout en réduisant les risques de collision entre avions. Les projets de guidage automatique des avions au sol sont, dans ce contexte, extrêmement prometteurs.

Les simulations effectuées dévoilent également une grande sensibilité des temps de roulage aux méthodes de résolutions employées et aux différents paramètres du modèle : le trafic au sol n’obéit pas seulement à des règles simples de file d’attente et peut être largement amélioré par l’optimisation judicieuse des trajectoires. Ces résultats montrent toute l’attention qui doit être portée sur le développement d’outils d’aide à la décision, dont l’importance pour la gestion du trafic aérien n’est plus à prouver.

Chapitre 11

Respect des créneaux de décollage

11.1 Introduction

Dans le contexte de la régulation européenne du trafic aérien, des créneaux de décollage sont attribués à certains vols, lorsque ces derniers prévoient de traverser des espaces aériens congestionnés. Un créneau est défini par une heure de décollage imposée à l'avion, retardée par rapport à l'heure initialement demandée par la compagnie. La tolérance officiellement admise pour le décollage effectif de l'avion est l'intervalle de temps d'un quart d'heure, commençant 5 minutes avant le créneau, et se terminant 10 minutes après.

Ces contraintes posent souvent de nombreux problèmes opérationnels pour le contrôle au sol, car les temps de roulage sont extrêmement variables et pendant les périodes les plus chargées, les bretelles d'accès à la piste sont encombrées par les files d'attente de départs. Lorsque d'autres perturbations opérationnelles viennent s'ajouter à ces difficultés, le problème peut devenir trop complexe et les créneaux sont parfois manqués.

A titre d'exemple, la figure 11.1 donne la proportion des départs soumis à régulation qui décollent avant, pendant et après leur créneau. Ces données, issues directement des traces radar des avions à Roissy, montrent la proportion des avions (entre 20% et 30%) qui se voient attribué le décollage trop tôt ou trop tard. Cette proportion, qui pourrait paraître relativement acceptable, est en réalité extrêmement dommageable pour la régulation du trafic (ATFM : *Air Traffic Flow Management*) car les seuils de capacités des espaces aériens sont malgré tout dépassés alors que certains avions sont considérablement retardés.

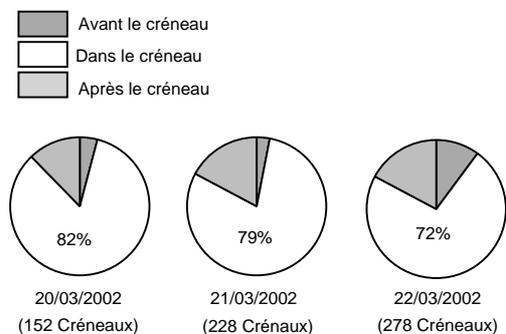


FIG. 11.1 – Respect actuel des créneaux à Roissy

Dans ce chapitre, l'objectif est donc d'évaluer la capacité d'un simulateur d'aéroport à fournir une prédiction des temps de roulage suffisamment précise pour prendre en compte les contraintes des créneaux de décollage dans l'optimisation des situations de trafic au sol.

11.2 Anticipation des temps de roulage

11.2.1 Problèmes posés

Appliquées à la minimisation des retards, les simulations effectuées au chapitre précédent montrent que l'horizon de prédiction le plus adapté aux incertitudes sur les vitesses de roulage est fortement réduit (de l'ordre de 5 minutes). Il est par ailleurs difficilement concevable d'étendre la prédiction au delà de 10 minutes : avec une incertitude de l'ordre de 20%, sur des vitesses de roulage pouvant atteindre 10 m/s, les positions des avions ne seraient connues qu'à 1200 mètres près.

En contre partie, un horizon limité à 5 minutes n'est clairement pas suffisant pour prévoir avec précision les temps de roulage des avions, surtout dans les périodes chargées, où ces derniers dépassent généralement 15 minutes : la gestion des créneaux de décollage pose donc un problème d'échelle, car elle nécessite un niveau de prédiction qui n'est pas directement compatible avec celui de la résolution des situations.

Dans ce cadre, le premier problème est d'évaluer aussi précisément que possible les futurs temps de roulage des départs à chaque étape de la simulation, pour en déduire ensuite l'heure à laquelle chaque mouvement régulé doit effectivement se diriger vers la piste. Ceci ne peut se faire sans considérer les futurs plans de vol, planifiés dans un horizon de prédiction bien supérieur à l'horizon H_p utilisé pour la résolution. Ce nouveau paramètre sera appelé *l'horizon d'anticipation* H_a et ne sera utilisé que pour l'estimation des temps de roulage.

Dans les parties suivantes, deux techniques sont développées et comparées pour réaliser cette anticipation. La précision de chaque technique sera mesurée par l'erreur entre les estimations et les temps de roulage effectifs, issus de la simulation classique avec résolution de conflits sur l'horizon de prédiction initial H_p .

11.2.2 Estimation par la séquence de piste

Principe

Une première technique d'anticipation des temps de roulage consiste à étudier un problème simplifié par rapport à la résolution des situations au sol, en ne considérant que les ressources les plus pénalisantes de l'aéroport, à savoir ses pistes : dans ce cadre, seuls les retards engendrés par les séparations en temps au niveau des séquences de piste sont pris en compte et l'utilisation de la piste est supposée maximale.

Chaque piste sera donc associée à une unique file d'attente de mouvements (dont certains correspondent à des futurs plans de vol non encore actifs dans la situation courante, car ne débutant que plus tard dans l'horizon d'anticipation). L'application exacte des temps de séparation imposés permet de déduire séquentiellement une heure de décollage ou d'atterrissage pour chaque mouvement.

Réalisation

La pertinence de cette technique d'estimation dépend en grande partie de la justesse de la file d'attente envisagée pour chaque piste : si l'ordre prévu des avions est trop différent de l'ordre qui sera

effectivement appliqué, les estimations des heures de décollage seront totalement erronées. Dans cette optique, deux types de contraintes permettent d'affiner l'élaboration des files d'attente :

- La situation courante d'un mouvement au départ, associée au temps minimal qui le sépare de la piste, donne une borne inférieure de son heure de décollage possible. Pour les départs correspondant à des futurs plans de vol non encore activés, cette borne peut être obtenue de la même façon, en augmentant l'heure de début du mouvement du temps de parcours minimal (calculé à partir du chemin le plus court).
- Les heures d'atterrissage des futures arrivées sont fortement contraintes car elles ne peuvent qu'être très légèrement retardées et ne peuvent pas changer d'ordre.

Ces contraintes permettent d'estimer les heures de décollage et d'atterrissage sur une piste comme la solution optimale d'un problème de satisfaction de contrainte relativement simple :

- Soient :
 - $(h_i)_{1 \leq i \leq n_A}$ les heures minimales d'atterrissage des n_A futures arrivées ;
 - $(h_i)_{n_A < i \leq n_A + n_D}$ les heures minimales de décollage des n_D départs prévus ;
 - (δ_{ij}) le temps de séparation minimal de la séquence $(i \rightarrow j)$ sur la piste.
- Les variables du problème sont les heures effectives $(t_i)_{1 \leq i \leq n_A + n_D}$ des décollages ou atterrissages ;
- Les contraintes sont données par :

$$\left\{ \begin{array}{ll} \forall i \in \{1, \dots, n_A\}, & h_i \leq t_i < h_i + \lambda H_p \\ \forall i \in \{n_A + 1, \dots, n_A + n_D\}, & h_i \leq t_i \\ \forall (i, j) \in \{1, \dots, n_A\}^2, & h_i < h_j \implies t_i < t_j \\ \\ \forall i \neq j \in \{1, \dots, n_D + n_A\}, & \left\{ \begin{array}{l} t_i \leq t_j \implies \delta_{ij} \leq t_j - t_i \\ t_j \leq t_i \implies \delta_{ji} \leq t_i - t_j \end{array} \right. \end{array} \right.$$

- Le critère à minimiser s'écrit :

$$f(t_i) = \sum_{i=1}^{n_A + n_D} t_i - h_i$$

Ce problème se résout par instanciations successives des variables, en supposant à chaque nouvelle instanciation que l'avion concerné est le suivant dans la séquence de piste. L'ordre d'instanciation initial (qui définit la stratégie d'exploration) est naturellement choisi en suivant l'ordre donné par les heures minimales (h_i) . Lorsque les contraintes rendent l'instanciation courante impossible, l'exploration est abandonnée et l'ordre d'instanciation est corrigé selon l'ordre lexicographique, comme cela a déjà été effectué lors de l'élaboration de la méthode 1 contre n avec recherche de classement initial dans la partie 7.5.3.

Résultats

La précision de cette technique d'anticipation est évaluée par simulation à Roissy : l'estimation des temps de roulage des départs est mémorisée à chaque étape de la simulation, puis comparée aux temps de roulage effectifs de la simulation. L'échantillon de trafic considéré est celui du 22/03/2002 et les paramètres de la simulation sont les suivants :

- Méthode de résolution : AG 1 contre n
- Période de résolution : $\Delta = 2$ minutes
- Horizon de prédiction : $H_p = 5$ minutes
- Horizon d'anticipation : $H_a = 30$ minutes

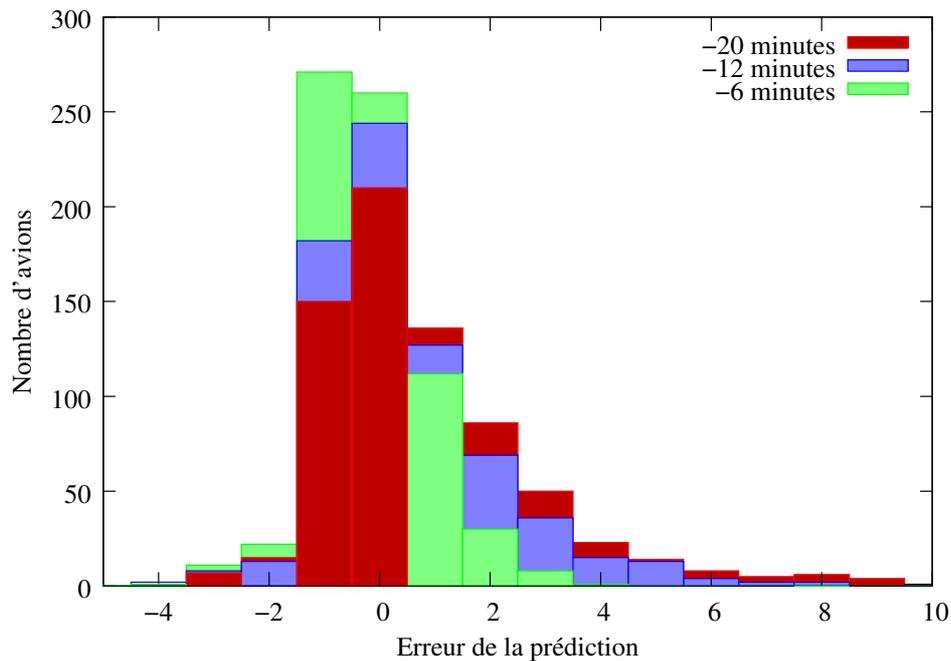


FIG. 11.2 – Anticipation des temps de roulage par la séquence de piste

- Incertitudes sur la vitesse : $\delta_v = 20\%$
- Passage à la situation suivante : avec brouillage

La figure 11.2 donne la distribution des erreurs des estimations calculées 6, 12 et 20 minutes avant : l'intervalle des écarts est restreint à $[-4; +10]$ minutes, mais seules 30% des estimations faites 20 minutes en avance ont moins de 30 secondes d'erreur.

11.2.3 Estimation par simulation

Principe

Une seconde technique d'anticipation des temps de roulage consiste à effectuer une résolution anticipée et simplifiée des futures trajectoires des avions, sur l'horizon d'anticipation H_a . Ceci est rendu possible en négligeant l'incertitude sur les vitesses lors de cette résolution simplifiée.

Réalisation

A chaque étape t_0 de la simulation (toutes les Δ minutes), tout plan de vol commençant avant $t_0 + H_a$ est activé : un ensemble de chemins lui est attribué et sa trajectoire directe (sans attente) est insérée dans la prédiction de trafic. La méthode de résolution 1 contre n sans incertitude est appliquée sur l'intégralité de la période anticipée $[t_0; t_0 + H_a]$ (cette méthode de résolution est ici choisie pour sa rapidité) : les temps de roulage des avions en mouvement dans l'horizon de prédiction initial $[t_0; t_0 + H_p]$ sont alors extraits directement de la résolution.

Bien évidemment, le passage à l'étape suivante de la simulation (Δ minutes plus tard) s'effectue après résolution par la méthode initiale (AG 1 contre n) sur l'horizon de prédiction réduit $[t_0; t_0 + H_p]$ et en considérant l'incertitude nominale sur la vitesse de roulage. De plus, l'application de cette

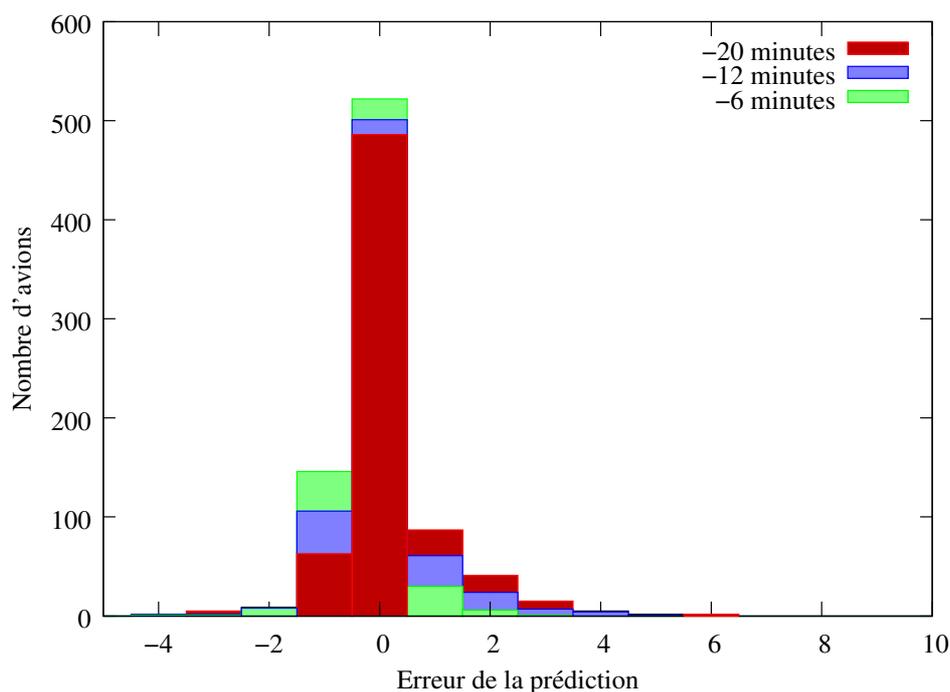


FIG. 11.3 – Évaluation des temps de roulage par simulation

résolution se fait avec brouillage : les nouvelles positions des avions sont choisies aléatoirement dans leur zone d'incertitude, comme cela a été décrit dans la partie 4.4.2.

Résultats

La simulation est effectuée avec les mêmes plans de vol et les mêmes paramètres que dans la partie précédente.

La figure 11.3 donne les différentes distributions des erreurs de chaque estimation. On peut observer que l'intervalle des écarts est réduit à $[-4; +6]$ minutes et que 71% des estimations faites 20 minutes en avance sont exactes à 30 secondes près :

- Le premier point important est que l'intervalle des erreurs des deux techniques d'anticipation des temps de roulage est inclus dans l'intervalle de tolérance officiel pour les créneaux de décollage ($[-5; +10]$ minutes). Cela signifie qu'il semble possible de prévoir les temps de roulage 20 minutes en avance, avec une précision compatible avec celle demandée par la régulation européenne.

Ce résultat est intéressant car la méthode de résolution appliquée dans la simulation n'est en rien corrélée aux estimations effectuées (le critère global à minimiser reste la somme des pénalités et non l'erreur des estimations).

- La seconde observation est que la deuxième méthode d'évaluation donne des estimations plus précises : comme beaucoup d'études le démontrent ([Idris 98] par exemple), les temps de roulage ne sont pas simplement fonction des files d'attente au décollage. Ils sont aussi largement influencés par la répartition du trafic près des aérogares et aux intersections les plus usitées de l'aéroport.

En conséquence, la méthode d'estimation des temps de roulage par simulation sera retenue pour la

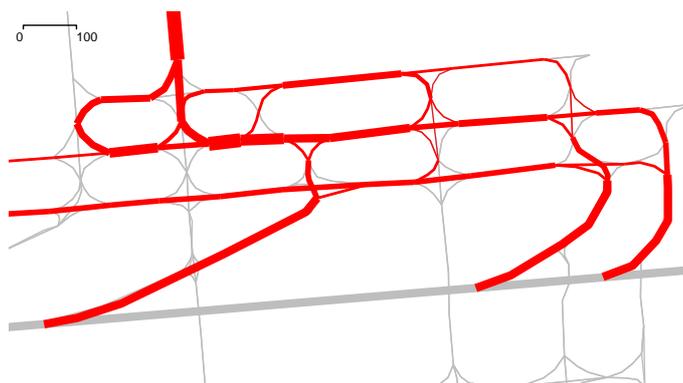


FIG. 11.4 – Aires d'attente

partie suivante. Les estimations seront cette fois-ci corrélées à la méthode de résolution pour prendre en compte les contraintes des créneaux de décollage imposés par la régulation européenne.

11.3 Prise en compte des créneaux de décollage

11.3.1 Approches envisagées

Concepts opérationnels

Deux principales difficultés relatives au respect des créneaux de décollage doivent être soulignées :

- Le créneau est une heure de décollage retardée (parfois plus d'une heure après le départ demandé par la compagnie). Il faut donc trouver des positions et des durées d'attente appropriées pour l'avion concerné, sans que cela ne pénalise démesurément le reste du trafic. Pour l'équipage et les passagers, la position d'attente la plus intéressante se situe naturellement au parking, mais une telle occupation du parking n'est pas toujours possible. En réalité, l'avion doit souvent rester « prêt au départ » car le créneau peut être finalement avancé, donc les passagers sont rarement débarqués.
- Pendant les pointes de trafic, les départs régulés doivent être positionnés stratégiquement dans la file d'attente pour la piste : si l'avion atteint la piste trop tôt, il doit attendre au seuil de piste, ce qui en bloque l'accès et retarde tous les autres vols. Si au contraire il l'atteint trop tard, le créneau est manqué et il doit théoriquement en attendre un nouveau. Les deux cas se produisent fréquemment et sont à l'origine de nombreux dysfonctionnements du contrôle aérien et de son système de régulation du trafic.

Deux principales techniques sont employées par les contrôleurs pour gérer ces problèmes :

- Les aéroports comme Roissy se sont équipés d'aires d'attente spécifiques près des pistes (figure 11.4). Ces aires permettent de modifier dynamiquement la séquence de piste de façon à l'adapter aux contraintes des secteurs d'approche, mais peuvent aussi servir à stationner les vols régulés pour améliorer le respect des créneaux de décollage.
- Des procédures opérationnelles spécifiques peuvent réserver certains accès à la piste aux vols régulés, dans les périodes où ces vols sont nombreux. Ces procédures ne peuvent cependant s'appliquer qu'aux aéroports présentant une infrastructure adéquate.

Application à la simulation

L'objectif est de résoudre les problèmes précédemment décrits avec des méthodes d'optimisation :

- Les aires d'attente font partie intégrante de la description de l'aéroport et sont donc directement prises en compte dans l'ensemble des chemins proposés aux avions (figure 11.4). Les méthodes de résolution peuvent donc aboutir à des ordres d'attente dans ces aires lorsque cela contribue à optimiser le critère global.
- Des procédures spécifiques de répartition du trafic sur les bretelles d'accès à la piste peuvent être envisagées lors de l'affectation de chemins (elles se définissent comme des restrictions sur les chemins à chercher).

Le problème du respect des créneaux de décollage sera alors intégré à la simulation du trafic comme suit :

1. L'estimation des temps de roulage définie dans la partie précédente permet de calculer une *attente initiale* pour chaque avion régulé.
2. Le critère d'optimisation doit être corrigé pour les vols régulés : la pénalité attribuée à ces vols sera définie en fonction de l'heure du créneau, lorsqu'elle est située avant la fin de l'horizon de prédiction ou en fonction de *l'attente initiale* lorsque le créneau est trop éloigné dans le temps pour intervenir directement. La modification du critère implique nécessairement une adaptation de la méthode de résolution déterministe 1 contre n , car les coûts du graphe à explorer en sont modifiés. Aucune adaptation particulière n'est en revanche nécessaire pour la résolution par algorithme génétique présenté au chapitre 8. La version hybride du chapitre 9 héritera directement des modifications opérées sur la méthode déterministe.

11.3.2 Modification du critère d'optimisation

Définition de l'attente initiale

Il s'agit d'évaluer, à chaque étape de la simulation et pour chaque départ i soumis à un créneau de décollage T_i , le temps τ_i que doit attendre l'avion avant de chercher à rejoindre la piste, dans la situation courante. On considérera que cette attente doit avoir lieu dès que possible (au parking en particulier, lorsque celui-ci est disponible) et qu'elle sera bornée par l'horizon de prédiction.

Une approximation de cette attente est logiquement donnée par :

$$\tau_i = \max(0, \min(H_p, T_i - t - r_i))$$

Où $\left\{ \begin{array}{l} t \text{ est l'heure courante de la simulation} \\ \text{et } r_i \text{ le temps de roulage restant de l'avion } i, \text{ estimé dans la partie précédente.} \end{array} \right.$

Critère d'optimisation

Pour un avion i soumis à un créneau de décollage T_i , l'objectif est modifié, puisque l'intérêt de l'avion peut être d'attendre. Lorsque le créneau est situé avant la fin de l'horizon de prédiction, la pénalité f_i de l'avion peut s'exprimer directement par l'écart au créneau. Dans la négative, la pénalité dépend de l'attente initiale qui a été calculée pour l'avion :

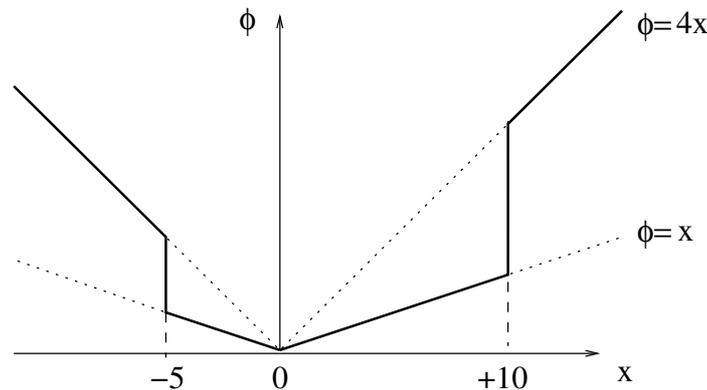


FIG. 11.5 – Critère pour un départ régulé

- Si le créneau est antérieur à la fin de l'horizon ($T_i \leq H_p$), la pénalité est fonction de l'écart entre le créneau et l'heure de décollage t_i résultante de sa trajectoire courante :

$$f_i = \phi(t_i - T_i)$$

Où la fonction ϕ est définie par intervalles, en fonction des seuils inférieurs (-5 minutes) et supérieurs ($+10$ minutes) de la tolérance relative au respect des créneaux (figure 11.5).

- Lorsque le créneau est plus éloigné ($T_i > H_p$), la pénalité mesure à la fois le détour l_i et l'attente d_i imposés à l'avion :

$$f_i = \beta|d_i - \tau_i| + \alpha l_i$$

Où τ_i est l'attente initiale calculée pour l'avion et β un coefficient pondérant l'importance accordée au respect des créneaux par rapport au retard des autres avions (en pratique, $\beta = 2$).

11.3.3 Adaptation de la méthode 1 contre n

Le graphe des possibilités d'un avion (défini dans la partie 7.4.3) doit être redéfini lorsque l'avion concerné est soumis à un créneau de décollage : pour correspondre au critère à minimiser, les coûts des arcs représentant des attentes et des arcs représentant l'avancée de l'avion sont inversés, tant que le retard de l'avion est inférieur à l'attente initiale qui lui a été calculée. Dans ce cas, le coût d'une attente devient nul, tandis que le coût de l'avancée de l'avion devient unitaire.

L'algorithme de *Branch and Bound* conserve la même stratégie d'exploration *meilleur en premier*, ce qui signifie que les nœuds ne seront plus développés dans le même ordre :

- Tant que l'attente imposée à l'avion est inférieure à l'attente initiale, l'algorithme développe en priorité les nœuds du graphe représentant une attente ;
- Une fois que l'attente initiale est atteinte, la progression de l'avion sur son chemin devient de nouveau la priorité.

Le classement initial des avions est également modifié : les départs régulés sont classés en fonction de leur créneau (alors que les autres avions restent classés par heure minimale prévue à la piste). En règle générale, les avions régulés sont ainsi rendus moins prioritaires tant que leur créneau est éloigné dans le futur, puis deviennent de plus en plus prioritaires au fur et à mesure que leur créneau se rapproche.

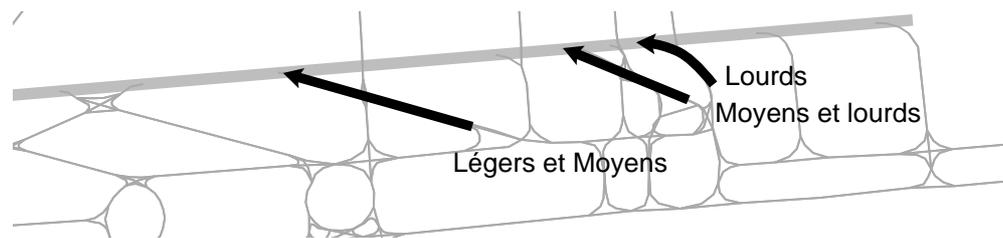


FIG. 11.6 – Stratégie d'affectation de bretelles par catégories

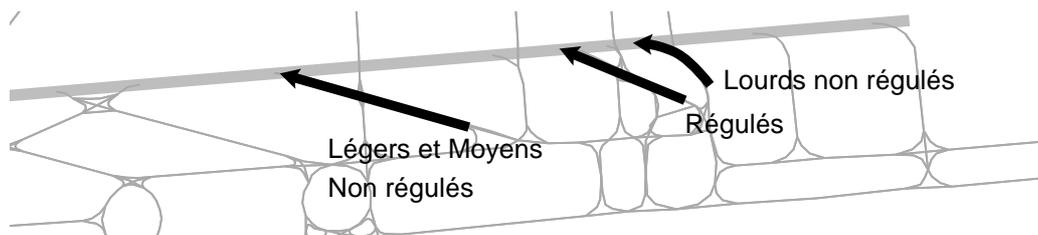


FIG. 11.7 – Stratégie d'affectation de bretelles par créneaux

11.3.4 Affectation de bretelles d'entrée de piste

Opérationnellement, le trafic est souvent réparti sur les bretelles d'entrée de piste en fonction de la catégorie de turbulence de sillage des avions : ceci permet d'optimiser la séquence de piste en laissant possible le choix de la catégorie du prochain décollage. Il semble par ailleurs avantageux de réserver certaines bretelles d'accès aux départs soumis à régulation, pour pouvoir leur imposer une attente sans gêner le reste du trafic.

En conséquence, deux stratégies d'affectation de bretelles d'accès sont comparées par simulation :

- Sans réservation de bretelle : les bretelles ne sont attribuées que par catégorie de turbulence de sillage (figure 11.6) ;
- Avec réservation de bretelle : pour chaque piste, une des bretelles est réservée aux vols ayant un créneau et les autres sont affectées en fonction des catégories de turbulences de sillage (figure 11.7).

11.3.5 Simulations

Les trois méthodes de résolution sont combinées aux deux stratégies d'affectation, ce qui donne lieu à six simulations de la même journée de trafic à Roissy (22/03/2002) avec prise en compte des créneaux de décollage dans le critère d'optimisation. Cette journée est sélectionnée sur le nombre relativement important de vols régulés : les données disponibles décrivent 278 créneaux pour 695 départs. Les paramètres de ces simulations découlent des mesures effectuées au chapitre précédent :

- Horizon de prédiction : $H_p = 5$ minutes
- Horizon d'anticipation : $H_a = 30$ minutes
- Période de résolution : $\Delta = 2$ minutes
- Incertitudes sur la vitesse : $\delta_s = 20\%$
- Sens uniques : appliqués
- Passage à la situation suivante : avec brouillage des positions

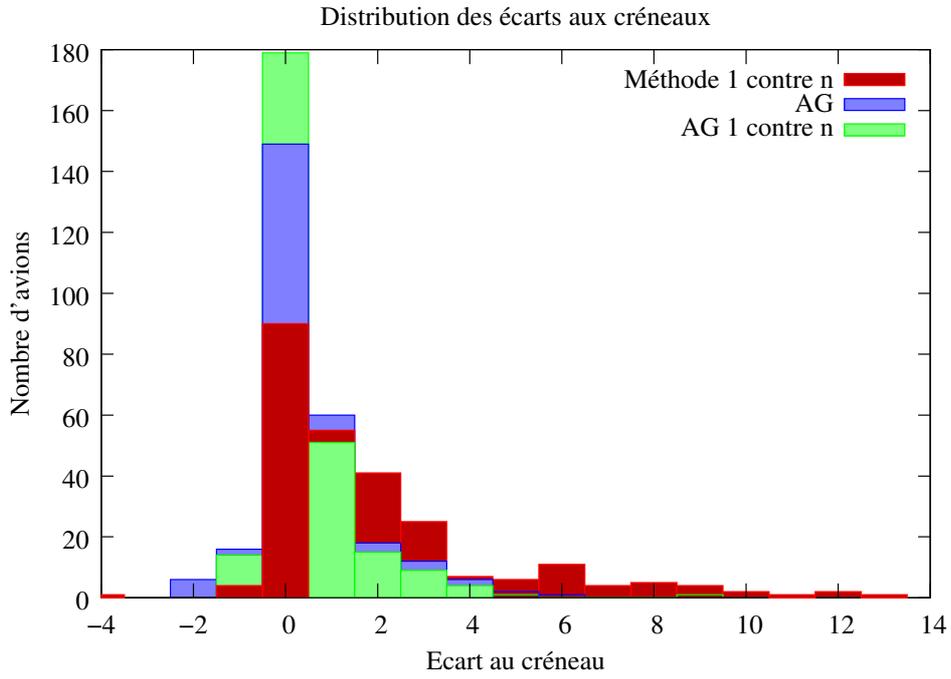


FIG. 11.8 – Écarts aux créneaux sans réservation de bretelle

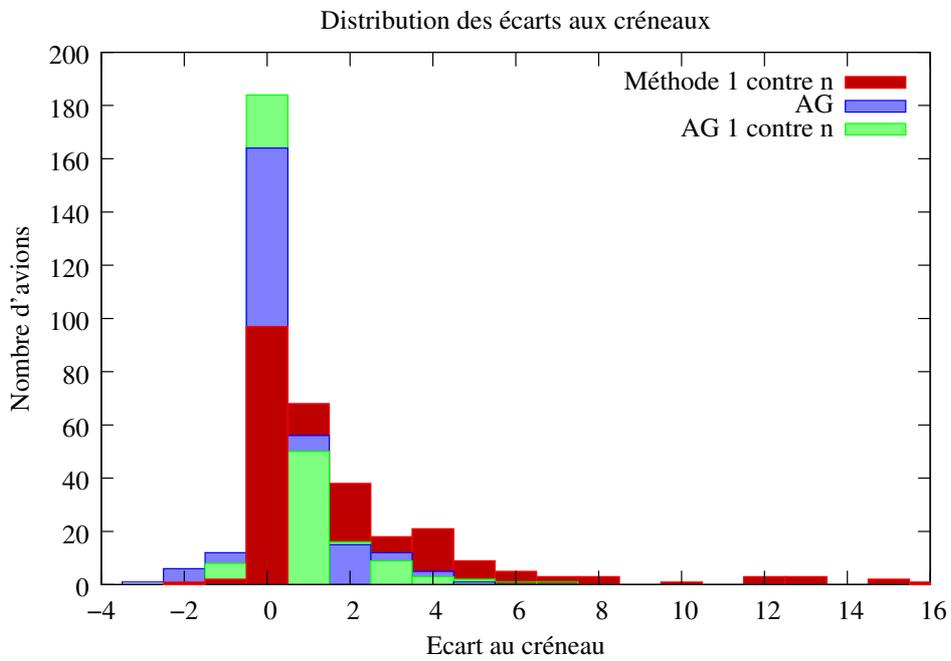


FIG. 11.9 – Écarts aux créneaux avec réservation de bretelle

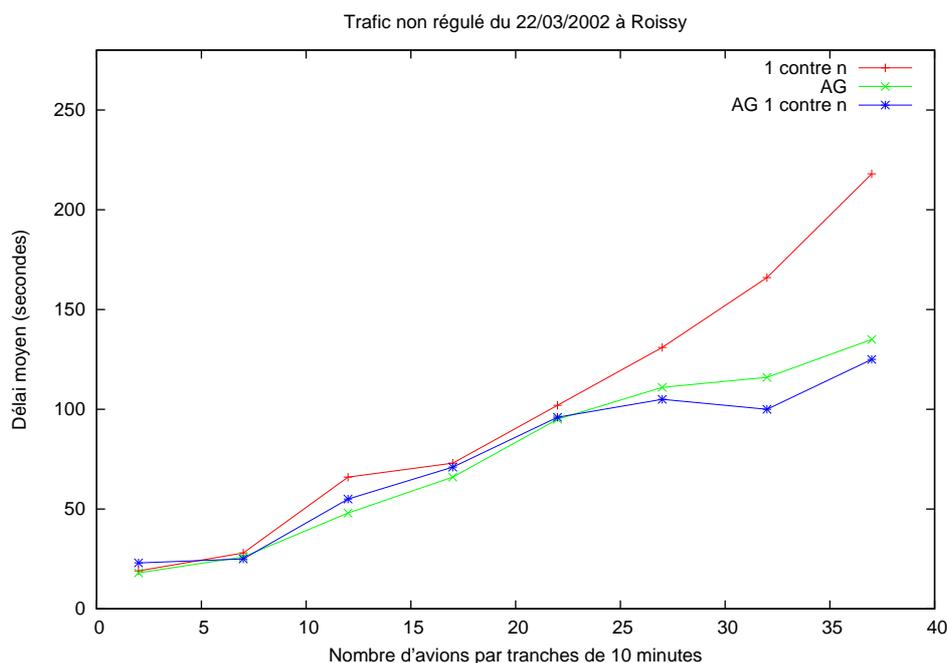


FIG. 11.10 – Délais des vols non régulés sans réservation de bretelle

Les figures 11.8 et 11.9 donnent la distribution des écarts aux créneaux pour les trois méthodes de résolution et pour chaque stratégie d'affectation de bretelle :

- Le premier objectif est atteint, dans la mesure où les créneaux sont effectivement respectés (par rapport aux tolérances officielles), avec les trois méthodes d'optimisation (à de rares exceptions près pour la méthode 1 contre n , où des créneaux sont manqués de quelques minutes).
- Un deuxième résultat intéressant est la similitude entre les deux stratégies d'affectation de bretelles : pour les deux méthodes de résolution (1 contre n ou algorithmes génétiques), réserver des bretelles d'entrée aux avions soumis à régulation ne diminue pas significativement les écarts aux créneaux. Cela signifie que ces écarts sont plus influencés par l'organisation du trafic en amont (avant les bretelles d'accès) et par la qualité de la prédiction des temps de roulage que par la répartition du trafic devant les pistes.
- Enfin, la supériorité des solutions trouvées par les deux méthodes globales utilisant un algorithme génétique est encore confirmée sur ce problème : l'algorithme hybride parvient à réduire l'intervalle des écarts à $[-1; +7]$ minutes avec plus de 80% des vols dans la tranche $[-1; +1]$ minute.

Les figures 11.10 et 11.11 donnent les corrélations entre le nombre d'avions au roulage (sur des tranches de 10 minutes) et le retard des avions non soumis à régulation, avec les deux stratégies d'affectation de bretelle. Ces corrélations montrent l'importance de la pénalisation induite sur le reste du trafic par la méthode avec réservation de bretelles : la diminution du nombre de bretelles d'accès disponible pour les avions non soumis à créneaux (la majorité du trafic) lui cause des retards plus importants. La méthode de résolution hybride parvient, là encore, à trouver des solutions intéressantes, qui minimisent à la fois le délai des avions non régulés et les écarts aux créneaux.

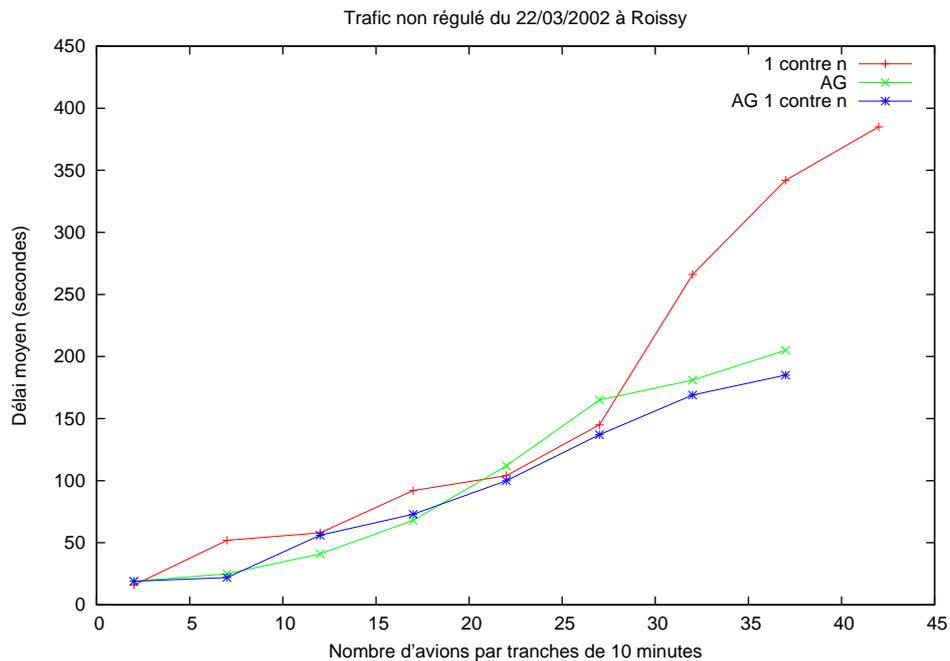


FIG. 11.11 – Délais des vols non régulés avec réservation de bretelle

11.4 Conclusion

L'étude du problème posé par les créneaux de décollage montre qu'il ne peut se traiter avec un horizon de prédiction aussi réduit que celui utilisé pour assurer les séparations : la gestion des créneaux nécessite une anticipation importante des temps de roulage. Dans ce contexte, deux applications prometteuses du simulateur de trafic au roulage sont présentées :

- Dans la première, la simulation est utilisée pour anticiper les temps de roulage à chaque période de la journée. Les résultats montrent que ces temps ne dépendent pas uniquement de la capacité des pistes et que l'estimation est beaucoup plus précise lorsqu'elle prend en compte toutes les portions problématiques des trajectoires des avions.
- La seconde application utilise la technique d'anticipation des temps de roulage pour résoudre le problème des créneaux de décollage, tout en continuant à minimiser les retards des avions non régulés. Deux stratégies d'affectation des bretelles d'accès à la piste sont étudiées. Il en ressort que les vols non régulés sont beaucoup moins pénalisés si aucune bretelle d'accès ne leur est interdite. Cette répartition plus homogène du trafic n'empêche pas, pour autant, de respecter les créneaux imposés, à condition que l'anticipation des temps de roulage soit suffisamment précise.

Conclusions et perspectives

L'outil de simulation du trafic aéroportuaire développé et présenté dans le cadre de cette thèse permet de résoudre les situations de trafic sur des grands aéroports comme Orly et Roissy, en poursuivant un objectif global d'optimisation. Il permet de dégager quelques caractéristiques du trafic au sol et peut apporter des réponses quantifiées aux questions des opérationnels :

- **La prévisibilité du trafic au sol** (temps de roulage « tout-compris ») s'avère très réduite : pour être réellement optimisées, les trajectoires précises des avions doivent être adaptées le plus fréquemment possible, ce qui les rend difficilement prévisibles plus de cinq minutes en avance. Contrairement aux idées reçues, les retards engendrés au niveau de l'aéroport ne dépendent pas uniquement de la capacité des pistes, car la phase de roulage proprement dite peut, par de faibles modifications des trajectoires, aboutir à des séquences de piste très différentes, ce qui induit de grandes variations des retards.

Sur ce point, la simulation détaillée et optimisée des situations de trafic à venir peut profiter en temps réel aux opérationnels, car elle fournit une estimation beaucoup plus précise des temps de roulage minimaux devant être envisagés.

- **La sensibilité des retards** aux solutions successives apportées à chaque situation de trafic est élevée : des solutions, qui peuvent sembler très proches en terme de délais générés dans l'immédiat (quelques secondes par avion), peuvent provoquer assez rapidement des situations totalement différentes (en nombre d'avions présents sur la plateforme par exemple) ce qui peut, à plus long terme, paralyser ou non l'aéroport. Ce phénomène d'accumulation très prononcé correspond au cercle vicieux (ou vertueux) de la congestion aéroportuaire : une bonne répartition du trafic, en plus d'améliorer la circulation des avions en cours de déplacement, est indispensable pour assurer des situations futures moins denses, tandis qu'un problème technique ou une erreur mineure peut avoir des effets négatifs sur tout le reste de la journée.

Dans ce cadre, de nombreux facteurs peuvent être identifiés comme influençant fortement les temps de roulage : incertitudes sur la vitesse de déplacement des avions, application ou non des sens préférentiels, stratégies d'affectation des bretelles d'entrée et de dégagement des pistes, normes de séparation, etc. Ces facteurs, qui illustrent toute la complexité de la gestion du trafic aéroportuaire, sont autant de points sur lesquels le système peut être amélioré.

- **L'efficacité de la résolution par classement** (méthodes un contre n et hybride) doit être exploitée : le classement peut en effet fournir des solutions aux situations de trafic de manière presque instantanée, en traitant un problème simplifié mais sans pour autant que les solutions trouvées ne s'éloignent excessivement de l'optimum. Ce résultat, moins vérifiable dans la résolution de conflits *en l'air*, peut s'expliquer par les règles d'utilisation des pistes qui nécessitent justement que les avions soient séquencés. De nombreux avantages en découlent : un classement est naturellement plus robuste aux incertitudes sur les vitesses de déplacement (qui diminuent lorsque les avions se suivent) et la transmission des solutions aux opérationnels en est facilitée : les contrôleurs peuvent à la fois valider et adapter ce classement, tandis que

les ordres de contrôle donnés aux pilotes consistent simplement à céder le passage à d'autres avions.

- **L'application de méthodes d'optimisation globales** semble très prometteuse pour le contrôle d'aéroport (peut-être plus que dans l'aérien) :
 - Le contexte du trafic au sol semble *a priori* plus adapté au partage de responsabilités entre systèmes d'assistances et contrôleurs, dans la mesure où un avion peut *s'arrêter* : en ultime recours (erreur, panne ou échec du système), la situation peut être temporisée par l'arrêt de certains avions. De plus, seul l'ajout d'une surveillance automatique du trafic peut encore augmenter la sécurité du système de contrôle (notamment par la détection des incursions piste, en filet de sauvegarde).
 - Certains problèmes opérationnels actuels, comme le respect des créneaux de décollage imposés par les organismes de régulation du trafic, se prêtent particulièrement bien à l'optimisation : leur problématique introduit des objectifs ou des contraintes supplémentaires qui guident la recherche de solutions ou qui en réduisent l'espace, ce qui correspond plutôt à une simplification du problème. A l'inverse, ces contraintes peuvent rapidement être jugées « trop nombreuses » pour les opérationnels, lorsqu'ils ne disposent d'aucun outil d'assistance pour les gérer.
 - Les évolutions envisagées du système de contrôle d'aéroport (d'ordres aussi bien technologiques que réglementaires et organisationnels) définissent un contexte d'intégration exceptionnel pour ces méthodes : la disponibilité des informations provenant du radar de surface, la transmission directe de données entre les positions de contrôle et les avions, l'éventualité (encore futuriste) de guidage au sol des avions « aux instruments » sont autant de points qui encouragent le développement d'outils automatiques pour une meilleure gestion du trafic aéroportuaire.

Les perspectives apportées par l'outil de simulation développé sont donc prometteuses, bien que de nombreuses améliorations doivent encore être envisagées avant de prétendre aboutir à un outil opérationnellement viable :

- **La prise en compte de spécificités aéroportuaires** non modélisées, comme les mouvements poste à poste, la gestion des autres véhicules, ou encore les contraintes de dégivrage des avions est indispensable à la finition de l'outil.
- **L'affinement du critère d'optimisation et des contraintes** du problème, en collaboration étroite avec les opérationnels, doit permettre de mieux représenter leurs objectifs et leurs besoins.
- **D'autres méthodes de résolution** peuvent être envisagées en fonction des impératifs temporels et qualitatifs des opérationnels. Sur ce point, les méthodes de séquençage de tâche [Nguyen 04], ainsi que des méthodes utilisant des algorithmes *any-time* [Zilberstein 95] peuvent trouver un cadre d'application qui semble particulièrement bien adapté.

Dans ce contexte, l'intégration de l'outil et de ses méthodes d'optimisation au sein de la plateforme SALSA (Systèmes d'Assistance à la circulation Locale et au Sol sur les Aéroports) du CENA (Centre d'Étude de la Navigation Aérienne) en tant que module d'aide à la planification et au cheminement des avions (*Taxi Planning Tool*) s'annonce comme une première étape aussi déterminante que passionnante.

Annexes

Annexe A

L'environnement aéroportuaire

A.1 Introduction

La connaissance des infrastructures aéroportuaires et des procédures qui lui sont rattachées permet de mieux comprendre les modalités d'écoulement du trafic au sol : l'objectif de cette annexe n'est pas de fournir une description détaillée de tous les équipements pouvant être utilisés sur un aéroport, mais de donner un aperçu général des conditions de circulation des aéronefs et des principales contraintes qui doivent être considérées lors de la conception d'un simulateur de trafic aéroportuaire.

A.2 Infrastructures aéroportuaires

D'une manière très générale, les infrastructures d'un aéroport se divisent en trois catégories principales, qui s'utilisent de manières très différentes : les aires de stationnement, les aires de piste, et les voies de circulation qui les relient (figure A.1).

A.2.1 Postes de stationnement

La circulation des avions dans les aires de stationnement se caractérise par des vitesses très faibles : la proximité des bâtiments et des autres avions impose un suivi précis des tracés au sol. Les postes de stationnement (ou parkings) d'un aéroport sont regroupés par proximité en *terminaux*, eux même regroupés en *aérogares* et deux principaux types de stationnement peuvent être distingués :

- Les parkings reliés à une passerelle d'accès : ils facilitent l'embarquement et le débarquement des passagers et simplifient les opérations de traitement au sol de l'avion, ce qui permet de minimiser le temps d'escale. Ils peuvent par ailleurs procurer un gain de surface, lorsque les avions sont positionnés *nez devant*, face à un bâtiment satellite. Dans ce cas, la sortie du parking en marche arrière nécessite l'assistance d'un *tracteur d'avion* et se fait par une procédure de *push-back* prédéfinie.
- Les parkings non connectés aux installations : l'accès à l'avion se fait par l'intermédiaire d'un *transbordeur*. Ces parkings sont plus souvent utilisés pour des vols longs courriers (la perte de temps à l'embarquement leur est moins dommageable) ou pour le transport de marchandises. La sortie du parking peut en revanche s'effectuer par une manœuvre autonome de l'avion, en marche avant.

Des normes officielles fixées par l'OACI (Organisation de l'Aviation Civile Internationale), imposent des marges de sécurité et définissent les types d'avions autorisés à stationner, en fonction de la

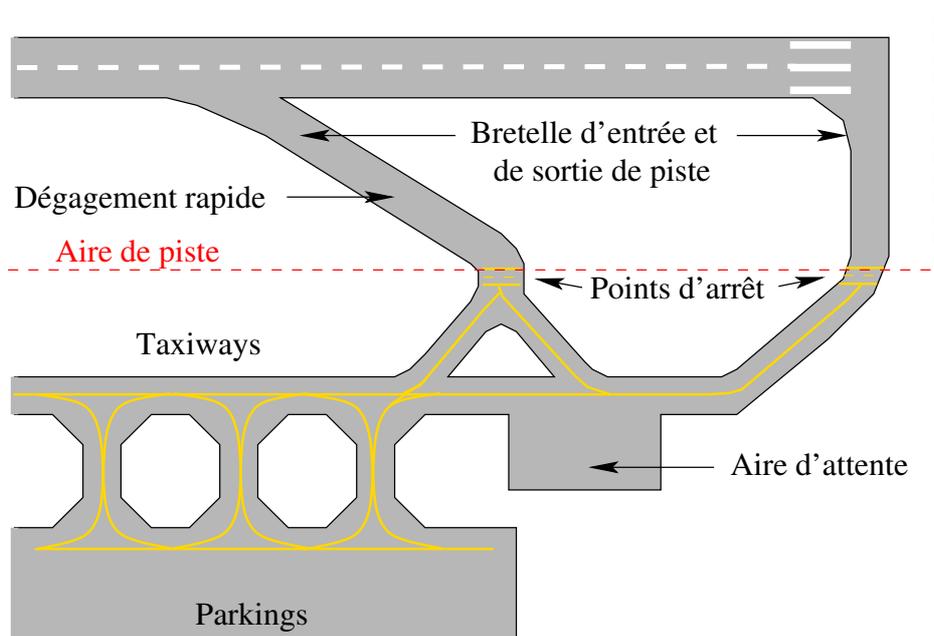


FIG. A.1 – Infrastructures aéroportuaires

géométrie du parking, de ses voies d'accès et de la présence d'éventuels systèmes d'aide au guidage de l'avion. Certaines installations peuvent être configurables : la disponibilité d'un parking dépend alors de l'occupation des parkings voisins.

A.2.2 Pistes

L'utilisation des pistes d'un aéroport est caractérisée par des vitesses élevées et des procédures (de décollages et d'atterrissages) qu'il est impossible d'interrompre : des *aires de piste* délimitent un espace protégé dans lequel aucun mobile ne doit s'engager sans autorisation. Ces aires sont généralement définies par des zones rectangulaires axées sur la piste, dont l'écartement est fixé par des normes officielles, en fonction des équipements et des conditions d'exploitation de l'aéroport (les écartements à l'axe de piste couramment admis sont 90 mètres par *beau temps* et 150 mètres en conditions météorologiques *dégradées*).

En pratique, les aires de piste sont matérialisées par des voies d'accès particulières, appelées bretelles d'entrée ou de dégagement de la piste :

- Les bretelles d'entrée (ou de traversée) de la piste sont caractérisées par un point d'arrêt obligatoire, signalé par un marquage au sol et des panneaux spécifiques : le pilote doit attendre l'autorisation explicite du contrôle avant de franchir ce point.
- Les bretelles de dégagement se divisent en plusieurs catégories, selon la vitesse maximale autorisée à l'entrée de la bretelle après (ou pendant) l'atterrissage : certaines sont notamment *rapides*, car elles permettent d'expédier la libération de la piste avant que la phase de freinage de l'avion soit totalement achevée.

Comme pour les parkings, des normes et des procédures d'exploitation déterminent les types d'avions autorisés à utiliser chaque piste, en fonction de sa largeur, sa longueur et de ses bretelles d'accès. Les pilotes portent la responsabilité finale du choix de la piste, en fonction des performances de l'avion (qui dépendent notamment de son poids, de l'altitude et de la température).

A.2.3 Voies de circulation

Les grandes plateformes aéroportuaires sont caractérisées par des voies de circulation (ou *taxiways*) complexes, matérialisées par des tracés au sol continus :

- Chaque portion est nommée, ce qui permet de décrire un chemin à suivre par une succession de portions à emprunter.
- Des panneaux de signalisation informent les pilotes des particularités liées à chaque zone de l'aéroport et les guident tout au long de leur cheminement. En règle générale, aucune notion de priorité ni de limitation de vitesse n'est précisée : le séquençage des avions à une intersection relève du contrôle au sol et les avions doivent rouler « au pas ».

Là encore, des normes définissent les marges de sécurité et les types d'avions autorisés à circuler sur chaque portion de *taxiway*, en fonction de la largeur de la voie et de sa proximité à certaines installations. Certaines voies peuvent également être dépendantes, dans la mesure où l'accessibilité de l'une se définit en fonction de l'utilisation des autres et des types d'avions concernés.

A.2.4 Autres aménagements

De nombreux aménagements particuliers peuvent être entrepris sur les grandes plateformes, au niveau des parkings, des *taxiways* ou des pistes pour améliorer la gestion du trafic. Deux d'entre eux sont très répandus et interviennent directement dans la définition de la capacité de l'aéroport :

- **Les aires d'attente** permettent d'isoler certains avions du reste du trafic, pour modifier l'ordonnement des avions sur certains axes de circulation ou pour gérer toute sorte de problèmes techniques. Elles se matérialisent par des espaces libres nommés, qui s'avèrent très utiles au séquençage optimisé des mouvements sur la piste et autour des parkings.
- **Les aires de dégivrage** sont indispensables au maintien du flux de départ dans certaines conditions météorologiques dégradées : les avions y sont recouverts d'un produit antigivre pour assurer leur décollage. L'effet du produit n'est bénéfique que pendant un temps limité, de l'ordre d'un quart d'heure (passé ce délai, le décollage devient impossible). Le dégivrage de l'avion ne peut donc se faire au parking que si l'avion est sûr de pouvoir décoller dans le temps imparti. Sinon, seule l'utilisation de ces aires, situées près des pistes, permet le décollage de l'avion. Dans tous les cas, la gestion des opérations de dégivrage des avions s'avère extrêmement délicate car elle cause des retards considérables.

A.3 Exploitation de l'aéroport

Un ensemble de *procédures* définit de manière extrêmement précise les règles à appliquer pour gérer les avions : elles sont décrites par le manuel d'exploitation de l'aéroport, régulièrement mis à jour par l'organisme gestionnaire. Le détail de ces procédures n'a bien évidemment pas sa place dans cette partie, qui n'en donne par conséquent qu'un aperçu très général.

A.3.1 Séparation des avions

La circulation des avions est contrainte par les normes de séparation (le non respect de ces normes n'entraîne pas obligatoirement une collision, mais correspond à une situation dangereuse et fait généralement l'objet d'un rapport d'incident). Les trois principaux types d'infrastructures de l'aéroport (parkings, *taxiways* et pistes) sont logiquement régis par des normes de séparation différentes :

Vitesses	
Catégorie	kts (Nœuds)
L (Légers)	100
M (Moyens)	135
H (Lourds)	145

Séparations en distance (Nm)			
1 ^{er} \ 2 ^{ème}	L	M	H
L	3	3	3
M	5	3	3
H	6	5	4

Séparations en temps (sec)			
1 ^{er} \ 2 ^{ème}	L	M	H
L	108	80	74
M	236	80	74
H	283	144	99

TAB. A.1 – Exemple de séparations à la piste

- **Dans les aires de stationnement**, les normes de séparation (entre avions, mais aussi avec les bâtiments) sont induites par les procédures d'entrée et de sortie de parking, fonction de la configuration de parkings utilisée : l'application précise de ces procédures est sous la responsabilité du pilote (de l'avion ou de l'éventuel véhicule tracteur).
- **Sur les voies de circulation**, les normes de séparation sont définies en distance et dépendent des types d'avions impliqués : pour les plus gros d'entre eux, une distance minimale de 60 mètres doit être respectée. Sur certaines voies particulières réservées à des types d'avions de faibles envergures, la norme de séparation peut être réduite.
- **Dans les aires de piste**, les normes sont plus complexes et sont notamment définies à la fois en distance et en temps :
 - L'espacement minimal de 60 mètres reste naturellement en vigueur entre les avions qui *circulent* dans l'aire de piste ;
 - Lorsqu'un avion décolle ou atterrit, aucun autre avion ne doit se trouver devant lui dans l'aire de piste (ce qui n'empêche pas les avions situés derrière lui de traverser la piste ou de *s'aligner* pour le décollage).
 - Un temps de séparation minimal doit être respecté entre deux mouvements sur la piste (atterrissages et décollages). Ce temps dépend de la catégorie de turbulence de sillage du premier avion et du type (notamment de la vitesse et du poids) du deuxième.
 - Une distance minimale, qui dépend également des catégories de turbulence de sillage, doit séparer les arrivées pendant toute la phase d'approche finale : ces séparations peuvent être transposées en temps, en fonction des vitesses des avions (tableau A.1).

De plus, certaines pistes peuvent être *dépendantes* (pistes sécantes ou proches l'une de l'autre) et impliquer d'autres restrictions d'utilisation (un seul mouvement à la fois sur les deux pistes par exemple).

A.3.2 Configurations de l'aéroport

La répartition du trafic sur l'infrastructure existante influence fortement la capacité de l'aéroport :

- Certaines pistes peuvent être réservées aux départs ou aux arrivées, ce qui les rend généralement plus capacitives. Une telle organisation nécessite cependant un nombre de pistes suffisant.
- Les avions décollent et atterrissent face au vent pour minimiser la distance de décollage et

d'atterrissage : le sens d'utilisation de la piste, appelé *QFU*, est imposé par le vent et modifie le cheminement de chaque mouvement.

- L'attribution des bretelles d'entrée et de sortie de piste modifie sa capacité : le flux de départs peut être réparti par catégorie sur chaque bretelle pour optimiser la séquence de décollage, tandis que l'utilisation des bretelles de dégagement rapides permet de maximiser l'utilisation de la piste.
- Certains types d'avion (notamment ceux de catégorie légère ou de forte turbulence de sillage) nécessitent des temps de séparation à la piste importants et induisent donc localement une baisse de capacité de la piste qui leur est affectée.
- Lorsque la structure des voies de circulation et des zones de stationnement le permettent, l'application de sens uniques permet de fluidifier le trafic au sol, ce qui peut augmenter la capacité de l'aéroport. Les sens de circulation les plus appropriés dépendent bien évidemment du sens d'utilisation des pistes.

Dans ce contexte, chaque *configuration* de l'aéroport correspond à un ensemble de règles spécifiant la répartition du trafic :

- Affectation des pistes et des bretelles d'accès, par type de mouvement ;
- Application de sens préférentiels de circulation ;
- Définition des sens d'entrée et de sortie des parkings.

Le choix d'une configuration se fait en fonction du vent et du trafic prévu sur l'aéroport, en coordination avec les secteurs d'approche : les phases de transition d'une configuration à l'autre provoquent des baisses de capacité importantes de l'aéroport.

A.3.3 Affectation des parkings

L'affectation des parkings aux mouvements relève de nombreuses procédures d'exploitation et d'accords entre les compagnies et le gestionnaire de l'aéroport :

- Le choix d'un parking est fortement contraint, notamment par le type de l'avion, la nature de l'assistance demandée pendant l'escale, ou encore la durée de l'escale.
- Le transfert des passagers d'un aérogare à l'autre prend du temps : pour une compagnie, le choix du parking n'est pas réellement lié au temps de roulage qu'il induit (peu prévisible) mais plutôt à sa proximité à d'autres parkings qu'elle utilise de façon coordonnée pour faciliter les correspondances des passagers (dans le même terminal ou le même aérogare).
- Lorsqu'un avion est inutilisé pendant une grande partie de la journée, il est généralement transféré vers un parking plus éloigné et moins coûteux : l'ensemble de ces mouvements *poste à poste*, bien que non prioritaires, induisent une charge supplémentaire pour le contrôle.

Le choix des parkings s'avère donc très peu flexible pendant la journée, surtout pour les vols passagers, dont le terminal et la porte d'embarquement doivent être connus le plus tôt possible.

A.3.4 Affectation des pistes

La piste affectée à un mouvement est généralement imposée par le contrôle d'aéroport, en fonction de la trajectoire du vol dans les secteurs d'approche, qui peut dépendre de nombreux paramètres :

- La destination (pour un départ) ou la provenance (pour une arrivée), qui correspondent à une direction initiale ou d'origine dans les secteurs d'approche ;
- La route et l'altitude de croisière souhaitées ;
- Le type d'avion, qui peut d'une part imposer directement une piste particulière et qui donne d'autre part l'ordre de grandeur de la vitesse de vol et du taux de montée ou de descente (ce qui

modifie la trajectoire imposable au vol dans les secteurs d'approche).

Ces considérations montrent que le choix de la piste ne peut que très rarement être effectué en fonction des intérêts des compagnies : les temps de roulage induits par la position du parking sur la plateforme ne sont généralement pas pris en compte.

Annexe B

Données disponibles

B.1 Introduction

La collecte de données informatiques relatives à l'organisation du trafic aérien demeure un point crucial et souvent limitant pour la recherche et les études : la précision avec laquelle ces données sont décrites influence de manière non négligeable le résultat de tout modèle.

En ce qui concerne les simulations arithmétiques, leur vocation est de fournir des indicateurs de performance ou de perfectibilité du système de gestion du trafic. Si ces indicateurs sont validés, ils peuvent également servir à orienter les futurs développements d'outils d'aide à la décision. Ceci ne peut se faire que si la pertinence des mesures effectuées est relativisée par les données employées en entrée du modèle : la description précise de ces données s'avère donc un passage obligé avant toute entreprise de modélisation.

Dans cet optique, cette annexe est consacrée à la description et à l'analyse des données utilisées.

B.2 Description topologique de l'aéroport

Une description topologique détaillée des voies de circulation est disponible pour les deux principaux aéroports français : Roissy Charles De Gaulle et Orly. Elle résulte de la collaboration entre le CENA (Centre d'Étude de la Navigation Aérienne), le SCTA (Service du Contrôle du Trafic Aérien) et ADP (Aéroports De Paris).

B.2.1 Éléments descriptifs

Coordonnées

Toutes les coordonnées des éléments descriptifs de l'aéroport sont fournies en deux dimensions et correspondent à des mètres mesurés sur un repère orthonormé $(\mathcal{O}, \vec{i}, \vec{j})$ de l'aéroport.

La position absolue du point \mathcal{O} n'est pas précisée mais les vecteurs \vec{i} et \vec{j} sont toujours orientés respectivement vers l'Est et le Nord. La correspondance entre la latitude φ et la longitude ψ (en radians) d'un point et ses coordonnées (x, y) dans $(\mathcal{O}, \vec{i}, \vec{j})$ est alors donnée par :

$$\begin{cases} x = l(\psi - \psi_{\mathcal{O}}) \cos(\varphi) \\ y = l(\varphi - \varphi_{\mathcal{O}}) \end{cases} \quad \text{ou} \quad \begin{cases} \varphi = \varphi_{\mathcal{O}} + \frac{y}{l} \\ \psi = \psi_{\mathcal{O}} + \frac{x}{l \cos(\varphi)} \end{cases} \quad \text{avec } l \simeq \frac{1852}{60}$$

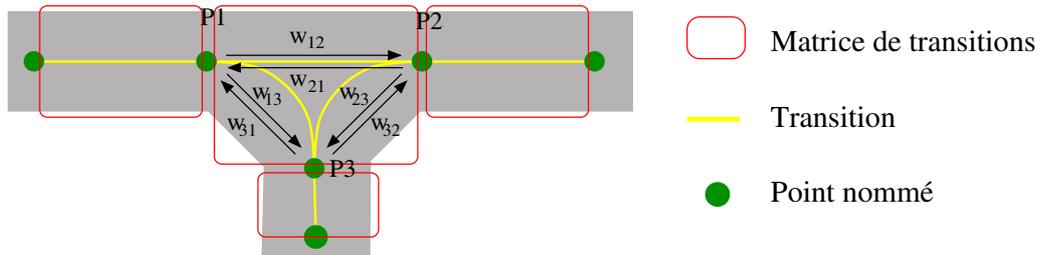


FIG. B.1 – Matrices de transitions

Les points nommés

Les points nommés sont tous les points d'intersection des tracés au sol. Ils sont décrits par l'association d'un nom caractéristique à une position précise sur l'aéroport et sont classés en trois catégories :

- **Les points situés sur un axe de piste** correspondent aux points de début du décollage (fin du roulage pour les départs) et aux points de sortie de la piste après l'atterrissage (début du roulage pour les arrivées).
- **Les points de push-back** sont les positions vers lesquelles un avion peut être repoussé d'un parking et sur lesquels il doit attendre d'être libéré par le véhicule.
- **Les points de décision** au roulage qui correspondent aux intersections de taxiways où le pilote doit choisir sa direction pour suivre le chemin qui lui est imposé.

Les matrices de transitions

Les voies de circulation de l'aéroport sont divisées en trois catégories :

- Les bretelles d'entrée et de sortie de piste sont décrites avec les pistes ;
- Les procédures d'entrée et de sortie de parking sont décrites avec les parkings ;
- **Les transitions** correspondent à toutes les autres portions de lignes tracées au sol.

L'ensemble des transitions est partitionné en matrices : chaque matrice correspond exhaustivement à toutes les lignes tracées au sol entre un sous ensemble de points nommés (figure B.1). Une telle matrice T , relative à n points nommés $(P_i)_{1 \leq i \leq n}$ est composée d'au plus $\frac{1}{2}n(n-1)$ transitions $(t_{ij})_{1 \leq i < j \leq n}$.

Le partitionnement des transitions en matrices est tel que l'enchaînement de deux transitions appartenant à la même matrice est interdit (car cela nécessiterait un demi-tour) : l'enchaînement $(P_1 \rightarrow P_2 \rightarrow P_3)$ sur la figure B.1 est ainsi rendu illicite.

Chaque transition t_{ij} correspond à la ligne tracée entre les points nommés P_i et P_j . Elle est décrite par une ligne brisée allant de P_i vers P_j et par deux coefficients réels w_{ij} et w_{ji} , appelés *poids de sens interdits*. Ces coefficients, fixés en accord avec les pratiques opérationnelles, permettent la définition de sens préférentiels de circulation sur l'aérodrome : w_{ij} (respectivement w_{ji}) est un coefficient multiplicatif à appliquer à la longueur de la ligne, afin de mesurer la gêne occasionnée par le passage d'un avion dans le sens P_i vers P_j (respectivement P_j vers P_i) : plus w_{ij} est grand, plus la portion de P_i vers P_j se rapproche d'un sens interdit.

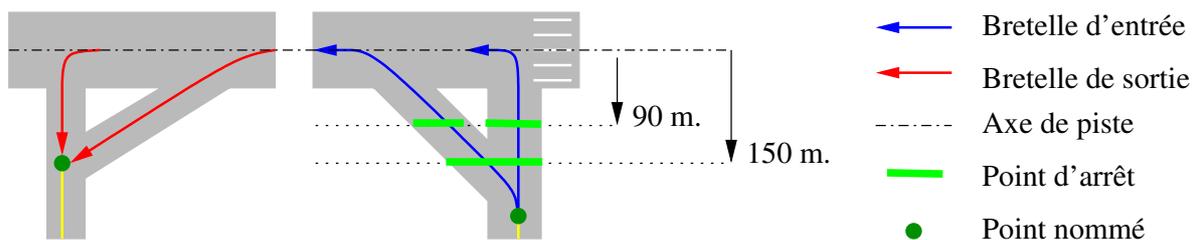


FIG. B.2 – Bretelles d'entrée et de sortie de piste

Les pistes et leurs voies d'accès

Chaque piste de l'aéroport est nommée et décrite par le segment formant l'axe de piste et les deux QFU (qui correspondent aux deux sens d'utilisation possibles de la piste). Chaque QFU est décrit par son nom, son seuil et ses bretelles d'entrée et de de sortie (figure B.2) :

- Le seuil du QFU, défini par un point situé sur l'axe de piste, correspond à la limite inférieure du point de toucher des roues à l'atterrissage.
- Une bretelle d'entrée est définie par une ligne brisée allant d'un point nommé à un point de l'axe de piste et par les deux points d'arrêt, situés à 90 mètres et à 150 mètres de l'axe de piste.
- Une bretelle de sortie est définie par une ligne brisée allant d'un point de l'axe de piste à un point nommé et par une catégorie de vitesse d'utilisation. Cette catégorie correspond à la vitesse maximale autorisée à l'entrée de la bretelle. Elle peut être *rapide* ($14m/s$), *standard* ($10m/s$), *lente* (environ $8m/s$) ou *au pas* ($5m/s$).

Les Parkings et leurs accès

Chaque position de stationnement est nommée et décrite par la position des extrémités avant et arrière de l'avion lorsqu'il y est garé et par les procédures d'entrée et de sortie de cette position (figure B.3) :

- Une procédure d'entrée est définie par une ligne brisée allant d'un point nommé à l'extrémité avant du stationnement. L'accès à cette entrée de parking peut être interdit après le passage par certains points nommés, car cela nécessiterait un demi-tour.
- Une procédure de sortie est définie par une ligne brisée joignant l'extrémité avant du parking à un point nommé. Elle peut être en marche avant ou en marche arrière. En marche arrière, la ligne passe par un point de push-back. Certains points nommés peuvent être interdits après cette sortie car cela nécessiterait un demi-tour.

L'ensemble des positions de stationnement est partitionné en sous-ensembles appelés terminaux. Chaque terminal est nommé et décrit par l'ensemble des parkings (généralement voisins) qu'il contient.

Configurations de l'aéroport

Chaque configuration de l'aéroport est décrite par :

- L'ensemble des QFU utilisables dans cette configuration ;
- Leurs critères d'utilisation, qui définissent le trafic autorisé à utiliser chacun des QFU, en fonction du type de mouvement (arrivée ou départ) et éventuellement de la trajectoire prévue dans le secteur d'approche (STAR : *Standard Terminal Arrival Route* et SID : *Standard Instrument Departure*).

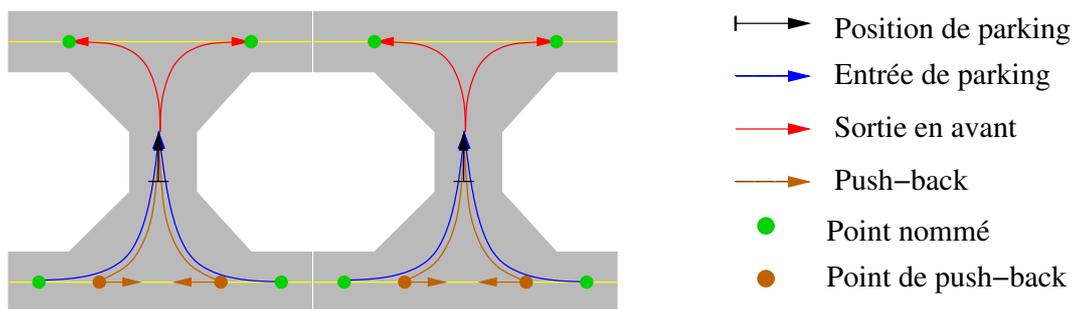


FIG. B.3 – Accès aux parkings

B.2.2 Analyse des données

Cette partie donne une première synthèse des problèmes posés par cette forme de description de l'aéroport. Ces problèmes sont ici classés en trois catégories : ceux qui résultent d'une absence d'information, ceux qui sont inhérents à la description proposée et enfin ceux qui sont relatifs à la fiabilité des données.

Informations non fournies

Les éléments suivants ne sont pas décrits, alors qu'ils pourraient apporter plus de réalisme aux simulations : le CENA prévoit à plus ou moins long terme de les ajouter à la description de l'aéroport.

Les aires de dégivrage (annexe A.2.4) : même si ces infrastructures ne servent que sous certaines conditions climatiques particulières, leur utilisation génère des contraintes fortes dans le cheminement des avions et demande une organisation opérationnelle complexe pour éviter la congestion (le dégivrage d'un avion prend du temps et l'avion ne dispose ensuite que d'un temps limité pour décoller).

L'utilisation de ces aires fait donc partie intégrante du problème d'optimisation du trafic au roulage et il sera souhaitable d'inclure ces notions dans une future version du simulateur. Cependant, la façon dont ces aires pourraient être décrites n'étant pas connue à ce jour, la version actuelle du simulateur ne prend pas en compte les problèmes de dégivrage des avions.

Les aires d'attente pour la piste (annexe A.2.4) : les voies de circulation concernées sont décrites comme des transitions ordinaires (comme décrit en B.2.1). Il ne sera donc pas possible de définir de règles d'utilisation spécifiques de ces aires d'attente.

Elles pourront cependant former des alternatives de cheminement pour les avions au décollage et seront donc utilisables (à juste titre) pour mettre en attente certains avions lorsque cela permet de résoudre une situation et d'améliorer le critère global d'optimisation.

Les aires de piste (annexe A.2.2) : les points d'arrêt sur les bretelles d'accès à la piste sont fournis, mais les transitions traversant ou interférant avec les pistes ne sont pas caractérisées comme telles.

Ces aires peuvent cependant être calculées puisqu'elles sont définies comme des zones rectangulaires axées sur la piste, et larges de 90m de part et d'autre de la piste (ou 150m par mauvais temps). Des calculs d'intersection géométrique permettent donc de connaître l'interférence d'une transition avec une aire de piste.

Ceci permet notamment :

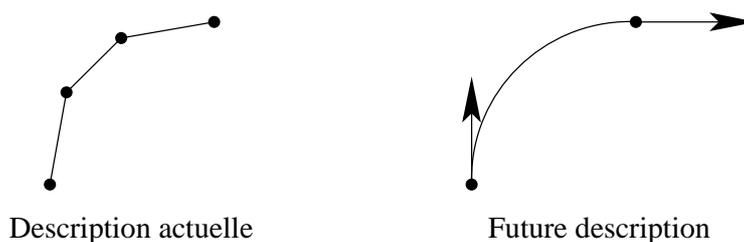


FIG. B.4 – Description des transitions

- d'éviter de proposer aux avions des chemins interférant avec les pistes lorsque cela n'est pas nécessaire ;
- de prendre en compte les règles de séparation spécifiques aux avions traversant une piste (annexe A.3.1).

Les limites d'envergures relatives à l'utilisation des taxiways et des parkings (annexe A.2) : le CENA prévoit d'ajouter ces informations dans les futures versions du fichier : chaque transition et chaque parking de l'aéroport se verra affecter une envergure maximale. Il sera alors possible d'interdire certaines transitions lors de l'affectation de chemins à un avion.

En attendant la disponibilité de ces informations, les limites d'envergures ne seront pas prises en compte pour l'affectation de chemins aux avions. On considérera par ailleurs que les parkings affectés aux avions tiennent compte des limitations : l'affectation de parking ne fera donc pas partie du problème d'optimisation. Ceci n'est pas gênant dans la mesure où les parkings sont généralement attribués bien avant l'heure du mouvement (pour limiter les déplacements des passagers au dernier moment), ce qui empêche toute optimisation dynamique.

Les dépendances entre parkings (annexe A.2.1) : comme pour les limites d'envergures, ceci ne pose pas de problème dans la mesure où l'affectation des parkings est considérée comme une donnée en entrée du problème à résoudre. Cependant, dans la réalité, l'occupation d'un parking peut bloquer certains accès à un parking voisin (sans pour autant le rendre inaccessible), ce qui contraint le cheminement d'un avion devant s'y rendre. Ces contraintes pourraient être modélisées par une distance minimale à respecter par rapport aux avions garés, mais ceci n'est pas implémenté dans la version actuelle du simulateur, car la description du trafic ne permet pas de connaître parfaitement l'occupation des parkings (ce point est détaillé dans la partie B.3.2).

Limites de la description

Les points suivants ont été identifiés comme posant un problème de cohérence et sont d'ores et déjà pris en compte par la division ATT pour les évolutions prévues des fichiers de description.

Les vitesses moyennes de déplacement en chaque point de l'aéroport doivent être calculées, afin de mesurer objectivement le coût d'un cheminement. Certaines vitesses pourront être définies comme des paramètres constants (vitesse de tractage de l'avion ou vitesse d'entrée dans un parking par exemple) mais dans le cas général, il faudra prendre en compte le rayon de virage de chaque transition pour lui attribuer une vitesse nominale de parcours cohérente.

Un problème de précision se pose alors pour ce calcul (détaillé au 3.2), car les transitions sont décrites sous forme d'une ligne brisée. Ce problème sera résolu avec les futures versions des fichiers de description, car une indication de vitesse sera fournie pour chaque transition. De plus, les transitions ne seront plus décrites par une ligne brisée, mais par deux vecteurs vitesses

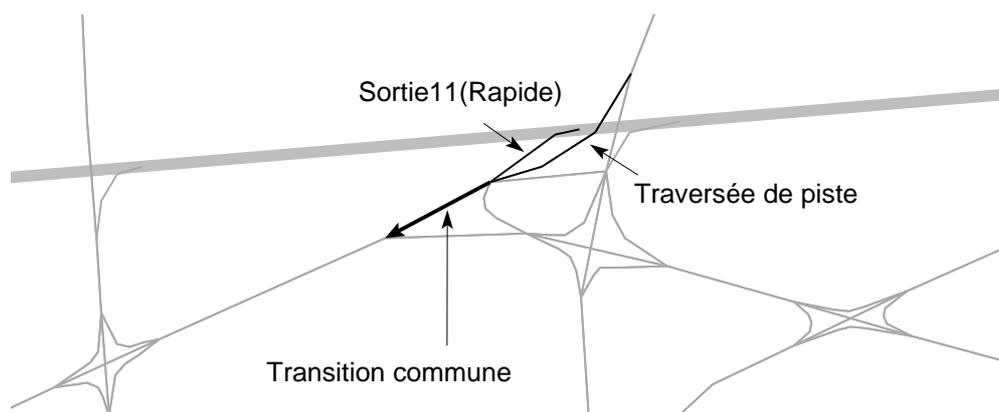


FIG. B.5 – Sorties rapides de piste

(l'un au début de la transition, l'autre à la fin) et la trajectoire de l'avion sera extrapolée par une courbe de bézier (figure B.4).

Les sens préférentiels peuvent dépendre de la configuration de l'aéroport (partie A.3.2) et peuvent influencer le choix des entrées et des sorties de parking, alors que les poids définis pour ces sens préférentiels (cf. B.2.1) ne prennent pas en compte les configurations et ne concernent que les transitions. Le CENA prévoit de préciser la dépendance aux configurations en affectant autant de poids qu'il y a de configurations à chaque transition de l'aéroport. Les procédures d'entrée et de sortie des parkings seront dans la même optique connectées plus directement au graphe des transitions pour permettre l'application des sens préférentiels à ce niveau.

Dans la version actuelle, les règles d'accès aux parkings ne dépendent pas des configurations, mais il existe deux versions de chaque fichier de description de Roissy et d'Orly : une version correspond à l'exploitation de l'aéroport face à l'ouest (ce qui correspond à toutes les configurations où les décollages et les atterrissages se font face à l'ouest) tandis que l'autre version décrit l'exploitation « face à l'est ». Il est donc nécessaire de sélectionner le fichier de description de l'aéroport en fonction de la journée de trafic utilisée et de changer dynamiquement de description lorsque l'aéroport change de configuration pendant la journée.

Les sorties rapides de piste peuvent également poser un problème de modélisation : en effet, les transitions suivant l'alignement d'une sortie rapide de piste sont prises à grande vitesse après un atterrissage et à faible vitesse dans les autres cas. La vitesse de parcours de ces transitions dépend donc de la provenance de l'avion. La figure B.5 montre un exemple à Roissy, où les avions traversant la piste peuvent rejoindre une sortie rapide de piste (sans pour autant rouler à grande vitesse).

Dans la version actuelle du simulateur, la vitesse retenue pour ces transitions correspond à la vitesse de roulage normale (on considère donc que les avions ralentissent avant la fin de la bretelle de sortie). L'utilisation des sorties rapides de piste sera donc moins bénéfique que dans la réalité.

Fiabilité des données

Les fichiers de description de l'aéroport contiennent beaucoup d'informations souvent issues du croisement de données de différents systèmes informatiques. Il est donc inévitable que des erreurs

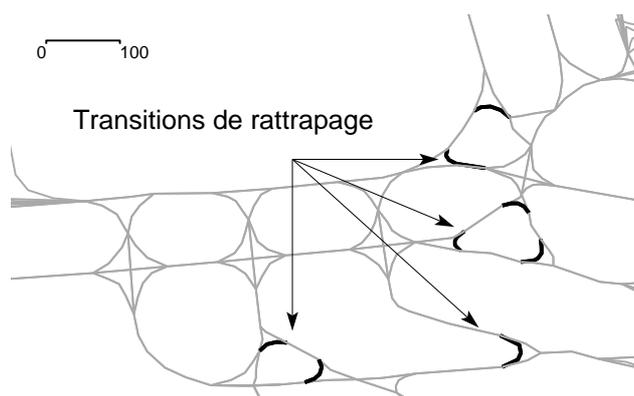


FIG. B.6 – Transitions de rattrapage

puissent se glisser dans la description de certains éléments. Cette partie décrit les types d'erreurs déjà rencontrés dans certaines versions de fichier et les traitements qui ont été développés pour y remédier.

Les coordonnées des voies de circulation peuvent être erronées. Il est arrivé en particulier que les coordonnées décrivant une transition ne soient pas données dans le bon ordre. Ceci aboutit à des angles anormaux dans la trajectoire que doit suivre un avion. La première conséquence néfaste peut être l'attribution d'une vitesse trop lente sur cette transition. Dans certains cas, des chemins réalistes peuvent même être déclarés illicites.

Pour éviter autant que possible ces problèmes, les coordonnées de chaque portion sont analysées : l'angle entre chaque segment successif de la ligne brisée ne doit jamais dépasser 90° . Ceci permet de détecter la plupart des erreurs de ce type.

Les accès aux parkings sont sources de nombreuses erreurs dans les premières versions des fichiers (notamment à Orly) : certains accès ne semblent pas correspondre à la réalité (bien qu'aucun élément écrit ne permette de l'affirmer), et des parkings sont dépourvus de procédure d'entrée ou de procédure de sortie.

Dans la version actuelle du simulateur, seuls sont détectés les parkings non reliés : ces parkings sont alors supprimés, car il ne serait pas possible d'attribuer un chemin aux avions voulant y stationner.

Des transitions de rattrapage sont décrites : ces transitions correspondent à des cheminements envisageables mais ne font pas partie des procédures normales (figure B.6). Ces transitions ont généralement un taux de virage élevé et correspondent toujours à un détour. Il n'est cependant pas évident de les caractériser plus directement.

Ces transitions n'empêchent pas la recherche de chemins réalistes mais elles la complexifient beaucoup, puisqu'elles offrent de nombreuses alternatives peu intéressantes. Pour éviter ces problèmes, une version épurée des fichiers de description a été générée (ces transitions ont été supprimées manuellement) et pourra être utilisée par le simulateur.

B.3 Description du trafic

La prédiction et la simulation de trafic nécessitent deux catégories d'informations : d'une part, les caractéristiques décrivant les avions utilisés (comme le type et les performances de ces avions) et d'autre part, les intentions des compagnies, données sous forme de plans de vol.

L'analyse à posteriori des conditions dans lesquelles s'écoule le trafic réel peut également s'avérer d'une grande utilité pour modéliser correctement les opérations aéroportuaires : les enregistrements radars disponibles à Roissy et à Orly seront en ce sens très appréciables.

B.3.1 Types et performances des avions

Types d'avions

Chaque type d'avion correspond à un ensemble de caractéristiques relatives à la taille, la motorisation et aux effets aérodynamiques de l'avion. Les données disponibles sélectionnées pour la gestion de l'avion au roulage sont les suivantes :

- Le type de moteur(s) (hélices, turbo-propulseurs ou réacteurs) ;
- La catégorie de turbulences de sillage (légère, moyenne, ou élevée), nécessaire mais non suffisante à la définition des séparations pour les séquences de piste.

Performances

Les performances d'un avion décrivent les distances nécessaires au décollage et à l'atterrissage et les vitesses de montée et de descente. Ces données dépendent des conditions météorologiques sur l'aéroport (température, pression atmosphérique, vent, etc), de la configuration de l'avion (sa masse en particulier) mais aussi des consignes d'exploitation de la compagnie.

Le détail de ces informations peut être trouvé dans les manuels de vols et dans les manuels d'exploitation des compagnies, mais des tables en résumé les valeurs moyennes, observées dans des conditions dites *standard*. Pour chaque type d'avion (et parfois pour chaque compagnie particulière), ces tables fournissent :

- Les vitesses et décélérations moyennes pendant chaque phase de l'atterrissage (en descente et sur la piste) ;
- Les vitesses et accélérations moyennes pendant chaque phase du décollage (sur la piste et en montée initiale).

Le tableau B.1 donne à titre indicatif un extrait de ces tables.

Les accélérations et vitesses au décollage et à l'atterrissage permettent le calcul d'une distance de décollage et d'atterrissage : cette distance peut influencer l'affectation des bretelles d'entrée et de libération de la piste proposées à chaque type d'avion, bien que les aéroports comme Roissy et Orly disposent de longueurs de pistes suffisantes dans la plupart des cas.

La vitesse de montée après le décollage est également un facteur important pour les règles de séquençage de piste : le temps de séparation entre deux décollages devra être augmenté lorsqu'un avion rapide suit un avion lent.

Limites identifiées

En ce qui concerne l'utilisation des données disponibles sur les types et les performances des avions au niveau de l'aéroport, les problèmes suivants ont été identifiés :

L'affectation des bretelles de piste, sur des aéroports comme Roissy et Orly qui disposent de longueurs de piste suffisantes, n'est pas directement fonction des limitations et des performances avion. Elle résulte plutôt de procédures opérationnelles prédéfinies, permettant une meilleure gestion des séquences de piste, en fonction de paramètres comme l'intensité du trafic sur l'aéroport, le parking de départ ou d'arrivée, ou encore le caractère régulé du vol (pour un départ).

	Atterrissage				Décollage				
	Courte finale		Freinage		Accélération		Montée initiale		
	V_f	δ_V	δ_V	V_0	δ_V	V_1	θ	δ_V	V_M
A320	70	-1.2	-3.1	19	2.8	75	13%	1.2	161
B747	78	-0.8	-1.3	22	2	80	7%	0.4	110
...				

Vitesses en m/s, Accélérations en m/s²

TAB. B.1 – Performances des avions

Ces procédures peuvent être trouvées en partie dans le manuel d'exploitation de l'aéroport, mais elles ne sont pas informatisées, et sont souvent trop complexes (ou ne sont pas assez rigoureusement définies) pour être modélisées directement. Il faut donc déterminer des règles précises d'affectation des bretelles de piste, spécifiques à chaque aéroport, qui soient le plus réalistes possibles, comme cela est détaillé dans l'annexe C.3.2.

La vitesse en finale peut être modifiée à la demande du contrôleur de tour (dans une plage admissible pour le type d'avion concerné) et cette pratique est essentielle pour la gestion optimisée des séquences de piste : sans cela, les capacités piste ne pourraient atteindre ni les valeurs déclarées, ni les valeurs observées. Il faut donc modéliser, au niveau du simulateur de trafic, la possibilité d'avancer ou de retarder (sous certaines conditions) les heures d'atterrissage initialement prévues. Ce point est précisé dans la partie 4.2.3.

B.3.2 Plans de vols

L'ensemble des informations décrivant les intentions d'une compagnie pour un vol sur un aéroport sera appelé plan de vol aéroport. Ces informations, fournies par le système MOZART (Moyens d'Optimisation pour les Zones d'Approche et de Régulation du Trafic), sont disponibles pour les terrains gérés par ADP (Aéroports De Paris). Elles résultent en réalité de la fusion (automatisée) de plusieurs sources de données :

- **Le système opérationnel COURAGE** (Calcul Optimisé des UCESO (Unité de Contrôle d'Espace Simultanément Ouvrables) et des Régulations pour l'Amélioration de la Gestion de l'Espace), qui archive les plans de vol utilisés et synthétisés par le système CAUTRA, chaque jour, sur toute la France. Ces données détaillent le type d'avion, le type de vol et la route prévue et réalisée (aéroports d'origine et de destination, balises et secteurs de contrôle intermédiaires, niveaux de vol et heures de passage). Lorsque le vol est régulé, les informations sur la régulation sont également disponibles (l'heure du créneau de décollage imposé par exemple).
- **Les mouvements de pistes**, qui sont archivés par le système de traitement radar à Roissy et à Orly. Ces données précisent notamment la piste, la nature (atterrissage, décollage ou remise de gaz) et l'heure réelle de chaque mouvement sur la piste.
- **L'affectation des parkings** aux vols, archivée par un système propre à chaque aéroport.

Description d'un plan de vol aéroport

La tableau B.2 donne la liste exhaustive des informations archivées pour chaque type de mouvement et pour chaque vol d'une journée de trafic dans le système MOZART. Ces informations sont

Donnée	Dispo.	Type	Description
Date	.	*	Date du vol
Indice	.	*	Numéro d'ordre dans le fichier du jour
Type de mouvement	.	*	Arrivée, départ, remise de gaz ou transit
Identifiant	.	*	Identifiant dans le système CAUTRA
Indicatif	o	*	Nom commercial du vol
Immatriculation	o	*	Immatriculation de l'avion
Type avion	o	*	Code OACI du type de l'avion
Transpondeur	o	*	Code transpondeur du vol
Numéro CAUTRA	.	*	Numéro du plan de vol dans le système CAUTRA
Origine	.	*	Code OACI de l'aéroport de départ
Destination	.	*	Code OACI de l'aéroport d'arrivée
Type de ligne	o	*	Régulière, non régulière ou de service
Type de vol	o	*	Cargo ou passager (mixte)
Type d'exploitation	o	*	Commerciale, privée, militaire ou d'état
Régime douanier	o	*	Métropolitain, Européen, International ou DOM-TOM
Aérogare	o	AD	Aérogare dans l'aéroport d'ADP
Parking	o	AD	Poste de parking sur l'aéroport d'ADP
Heure demandée	o	AD	Heure de départ ou de fin prévue
Heure effective	.	AD	Heure de départ ou de fin réalisée
QFU	o	*	QFU réellement utilisé
Heure QFU	.	*	Heure réelle (décollage / atterrissage / remise de gaz)
Entrée	o	A	Point d'entrée dans les secteurs d'approche
Heure entrée	o	A	Heure du début de l'attente en l'air
Heure sortie	o	A	Heure de fin de l'attente en l'air
Heure activation	o	D	Heure d'activation du plan de vol
Créneau de décollage	.	D	Heure du créneau de décollage, s'il y en a un
Délai	.	D	Délai induit par l'éventuel créneau
Sortie	o	D	Point de sortie du vol
Heure sortie	o	D	Heure réelle sur le point de sortie
Niveau	o	D	Niveau de vol demandé
Heure QFU	.	R	Heure prévue d'atterrissage
Heure remise de gaz	.	R	Heure de détection de la remise de gaz

Toujours disponible	.	*	Tout type de mouvement
Optionnel	o	A	Arrivées
		D	Départs
		R	Remises de gaz

TAB. B.2 – Plans de vol aéroport

disponibles pour tous les aéroports de la région parisienne (gérés par ADP). On y trouve notamment :

- Le type du mouvement (arrivée, départ, remise de gaz ou transit) pour l'aéroport géré par ADP (le transit désignant un vol allant d'un aéroport d'ADP à un autre).
- Le parking, le QFU et le type de l'avion ;
- Les heures demandées et effectives de départ du parking ou d'arrivée au parking, données à la minute près ;
- L'heure du décollage ou de l'atterrissage effective, donnée également à la minute près ;
- Et enfin le créneau de décollage pour les départs régulés.

Analyse des données plan de vol

Les données plans de vol seront utilisées pour générer le trafic. Les quelques points suivants précisent les limites de cette description et leurs conséquences sur le trafic qui sera simulé :

Les heures sont fournies à la minute près. Ceci est suffisant pour les départs, dont l'heure de libération du parking n'est jamais anticipée avec plus de précision par les opérationnels. Par contre, les heures réelles d'atterrissages relèvent d'un séquençage précis des arrivées, permettant d'assurer les séparations tout en maximisant l'utilisation de la piste. Pour les simulations, ces heures seront considérées comme une contrainte du problème et devront donc être redéfinies plus précisément (ceci sera détaillé dans l'annexe C.4).

L'adaptation des données à l'aéroport n'est pas toujours parfaite et devra systématiquement être étudiée : on peut notamment relever quelques différences dans le formatage des noms donnés aux parkings ou aux QFU. De plus, la description de l'aéroport correspond à un état de ce dernier à une date fixée, alors que les infrastructures aéroportuaires sont évolutives. Certains éléments référencés dans les plans de vol peuvent donc ne correspondre à aucun élément de la description de l'aéroport (utilisation de nouveaux parkings par exemple). Des traitements spécifiques (annexe C) devront donc être mis en œuvre pour prendre en compte ces plans de vol.

La disponibilité des informations n'est pas systématique : même certaines informations essentielles à la description du vol (comme le parking, le QFU, ou l'heure de début du mouvement) est optionnelle.

Le tableau B.3 précise la fréquence de ces informations sur un échantillon de 5 journées de trafic. Ces chiffres montrent qu'en moyenne, pour pouvoir être pris en compte, 12% des vols nécessiteront une affectation de parking et 2% une affectation de QFU. La manière de réaliser ces affectations est détaillée dans l'annexe C.

Les mouvements poste-à-poste ne sont pas précisés. Même si ces mouvements sont moins prioritaires que le reste du trafic, ils peuvent participer à la congestion des zones de stationnement. De plus, leur description est nécessaire à la connaissance exacte de l'état d'occupation des parkings tout au long de la journée.

Les mouvements corrélés (les couples arrivées - départs partageant le même avion) ne sont pas explicitement décrits. Cette corrélation pourrait être déduite à partir des immatriculations des avions affectés à chaque mouvement, mais cette information n'est pas toujours fournie, comme le montre le tableau B.3 : aussi bien à Orly qu'à Roissy, près d'un quart du trafic ne peut être corrélé correctement.

En réalité, les mouvements poste-à-poste et l'affectation des avions aux vols résultent des choix particuliers de chaque compagnie. Cette gestion opérationnelle pose de nombreux problèmes de coordination en temps réel : un vol peut être retardé suite au retard de l'avion qui lui est affecté. Ceci peut

Informations non fournies pour Roissy						
	Vols	Parking		QFU	Heure début	Immatriculation
		non fourni	inconnu	non fourni	non fournie	non fournie
18/06/1999	1421	335	4	13	160	69
26/11/1999	1406	40	1	46	25	40
29/01/2001	1419	12	32	32	6	12
22/03/2002	1461	18	93	40	9	18
	5707	7%	2%	2%	3%	2%

Informations non fournies pour Orly						
	Vols	Parking		QFU	Heure début	Immatriculation
		non fourni	inconnu	non fourni	non fournie	non fournie
18/06/1999	807	3	61	5	0	3
26/11/1999	785	4	76	2	3	4
29/01/2001	737	1	69	2	0	1
22/03/2002	645	1	46	15	1	1
	2974	0.3%	8%	0.8%	0.1%	0.3%

TAB. B.3 – Disponibilité des informations

inciter les compagnies à modifier dynamiquement ces affectations, en fonction justement de la situation et des retards courants. Ces choix peuvent donc également entraîner la modification du parking et du QFU des vols...

La prise en compte de ces contraintes lors de l'optimisation des situations de trafic au roulage serait donc difficilement réaliste, puisqu'elle nécessiterait la modélisation de nombreux comportements opérationnels et commerciaux pour l'instant non décrits. Ceci aura deux principales conséquences sur les situations de trafic simulées, dans la version actuelle du simulateur :

- les mouvements poste-à-poste ne pourront être modélisés ;
- l'état d'occupation des parkings ne pourra être connu parfaitement, ce qui implique que l'affectation des parkings aux mouvements ne pourra pas faire partie du problème d'optimisation.

B.3.3 Enregistrements radar

Les aéroports comme Roissy et Orly sont munis d'un radar permettant la détection et le suivi de tout objet mobile sur l'aéroport. Les éléments détectés par ce radar sont corrélés au système de traitement des plans de vol avant d'être représentés sur un écran de contrôle. Ceci permet au contrôleur de tour d'avoir une vision synthétique de la situation sur toute la plate-forme.

Ces données corrélées sont enregistrées et archivées chaque jour, ce qui permet de connaître la trajectoire précise de chaque avion (et de chaque véhicule) sur ces aéroports.

Description des pistes radars

Pour chaque mobile repéré par le radar, les enregistrements précisent :

- **Des informations sur l'identification du mobile.** Chaque mobile est ainsi classifié en tant que vol inconnu, vol identifié, survol ou véhicule. Dans le cas d'un vol, les informations sont semblables à celles du plan de vol et elles contiennent notamment :

- Le type avion ;
- Le parking et le QFU sur l'aéroport concerné ;
- Les informations sur la régulation pour un départ régulé.
- **La trajectoire réelle suivie par le mobile**, sous forme d'une suite de plots radars. La fréquence des relèvements de ces plots varie selon les fichiers. Elle est d'environ un plot toutes les une à trois secondes. Chaque plot est décrit par :
 - L'heure précise du relèvement radar, donnée à la seconde près ;
 - Une position sur l'aéroport, donnée par ses deux coordonnées en mètres et une altitude en centaines de pieds ;
 - Les composantes de la vitesse du mobile dans le repère de l'aéroport.

Trajectoires radar vectorisées

Les enregistrements des pistes radar sont également disponibles sous forme de *trajectoires vectorisées* : la trajectoire correspondant aux différents plots radar d'un avion est vectorisée pour être corrélée à la description topologique de l'aéroport par un algorithme d'intersection de polygones. Ceci permet de localiser la trajectoire des avions par rapport aux éléments descriptifs de l'aéroport pour en obtenir une description plus synthétique. Chaque vol est ainsi décrit par :

- Les informations sur le plan de vol de l'avion ;
- Le cheminement de l'avion, décrit par la suite des *points nommés* de l'aéroport (lorsque ceux-ci ont pu être identifiés) accompagnée des heures de passage sur ces points ;
- Les attentes du vol, localisées par rapport aux éléments descriptifs de l'aéroport ;
- Pour un départ, la description complète du décollage (heure de fin de roulage, d'alignement, de début et de fin de décollage, bretelle d'entrée empruntée et distance de décollage) ;
- Pour une arrivée, la description complète de l'atterrissage (heures du poser des roues et de libération de la piste) et éventuellement les traversées de piste après l'atterrissage.

Limites identifiées

Les enregistrements radars disponibles sont des données précieuses pour l'analyse à posteriori des procédures opérationnelles et des problèmes de congestion au niveau de l'aéroport. Ces informations seront utilisées pour comparer les situations générées par le simulateur aux situations réelles de la même journée de trafic. Les points suivants précisent les limites de cette comparaison :

Les trajectoires d'entrée et de sortie de parking ne sont pas enregistrées (les détections du radar sont trop imprécises dans ces zones où de nombreux véhicules circulent). Il sera donc impossible de confronter les trajectoires générées par le simulateur aux trajectoires réelles. En particulier, les vitesses d'entrée et de sortie de parking ainsi que la durée des attentes après *push-back* ne seront pas comparables.

Certaines trajectoires peuvent être erronées, suite aux imprécisions du radar ou aux dysfonctionnements des traitements effectués au cours de l'archivage. On peut en effet constater exceptionnellement :

- des absences de détection d'un avion pendant une durée non négligeable (supérieure à une minute) (figure B.7) ;
- des fusions de trajectoires de deux mouvements différents (figure B.8).

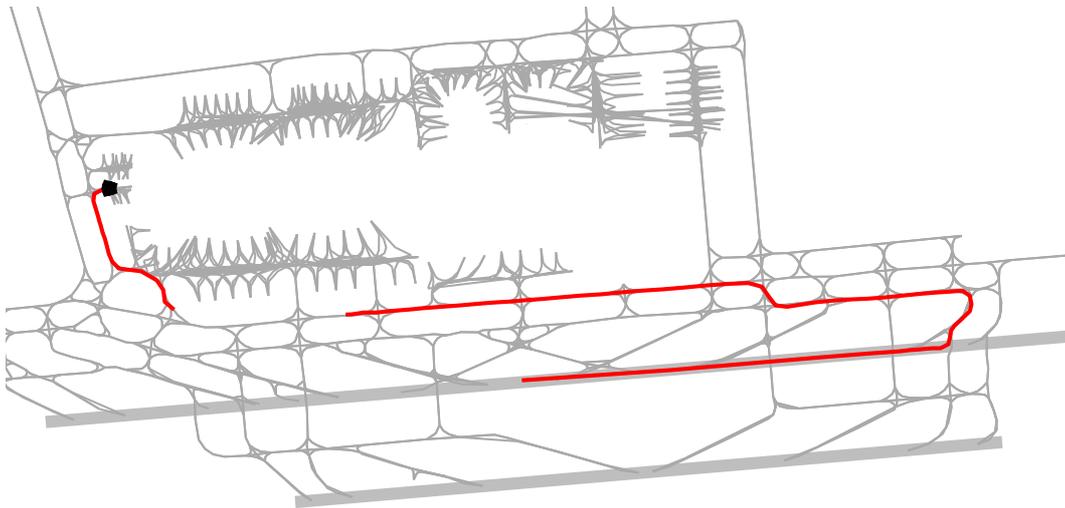


FIG. B.7 – Trace radar interrompue à Roissy

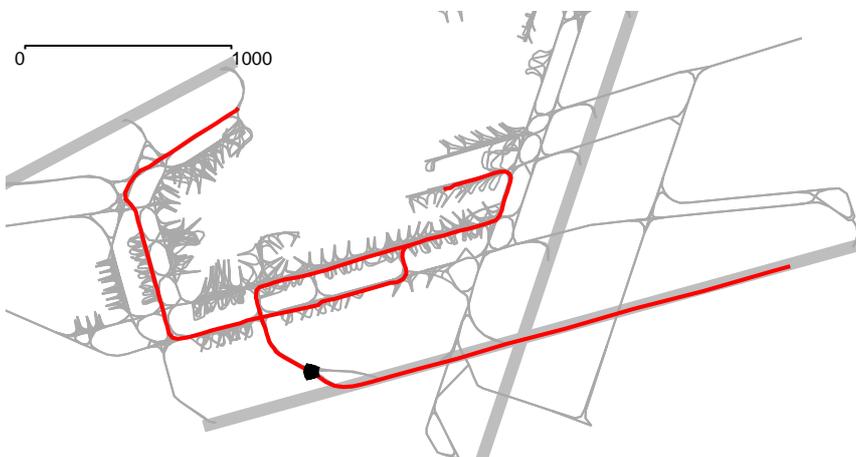


FIG. B.8 – Traces radar fusionnées à Orly

Annexe C

Traitement des plans de vol

C.1 Introduction

Les données disponibles décrivant le trafic ne sont pas toujours complètes (annexe B) et doivent être traitées avant d'être utilisées :

- Certaines informations importantes peuvent ne pas avoir été archivées.
- Dans d'autres cas, les informations archivées ne correspondent à aucun élément connu de l'aéroport (car elles font référence à de nouvelles infrastructures ou à des éléments dont le nom a été modifié).

Même si cela ne concerne qu'une partie relativement faible du trafic, éliminer systématiquement les plans de vol posant de tels problèmes serait inapproprié : les situations qui en découleraient dans les simulations seraient moins denses et moins représentatives des situations réelles. Il sera donc préférable de compléter ou d'adapter ces plans de vol, à condition que ces traitements respectent quelques règles élémentaires :

- Ils doivent être effectués de manière statique (donc avant la simulation proprement dite) : le but des simulations étant de comparer entre elles différentes méthodes d'optimisation, ces traitements doivent être identiques pour toutes les simulations. Ils doivent être reproductibles et ne pas dépendre d'un état particulier d'une des simulations.
- La cohérence de l'échantillon de trafic ne doit pas être mise en cause. Lors de l'affectation de parking par exemple, il faut garantir que l'occupation des parkings reste réaliste : elle ne doit pas impliquer, par exemple, que deux départs libèrent le même parking à quelques minutes d'intervalle. De même, l'affectation des pistes aux mouvements ne doit pas créer de situations improbables où trop d'avions partageraient la même piste (pendant qu'une autre piste serait inutilisée).

Cette annexe est consacrée à la description des traitements mis en œuvre pour extraire des fichiers de plans de vol actuellement disponibles une description du trafic aussi cohérente que possible. Les informations essentielles à la description d'un vol sont étudiées : le parking, la piste, la bretelle de piste et l'heure du début du mouvement. Ces traitements sont détaillés ici pour illustrer les problèmes devant absolument être posés avant d'entreprendre toute simulation et pour rendre moins opaque le trafic qui sera finalement simulé. Ces traitements restent cependant évolutifs : ils demanderaient certainement à être adaptés, à chaque fois que de nouveaux échantillons de trafic seront utilisés.

	Affectations de parking à Roissy			
	Vols	/ Aéroport	/ Compagnie	non pris en compte
18/06/1999	1421	4	311	24
24/03/2000	1419	1	30	5
29/01/2001	1419	32	9	3
22/03/2002	1461	88	18	5
	5707	2%	6%	0.6%

	Affectations de parking à Orly			
	Vols	/ Aéroport	/ Compagnie	non pris en compte
18/06/1999	807	59	2	3
24/03/2000	759	61	0	0
29/01/2001	737	67	3	0
22/03/2002	645	45	2	0
	2974	8%	0.3%	0.2%

TAB. C.1 – Affectations de parking

C.2 Affectation de parkings

L'analyse des plans de vol (annexe B.3.2) montre que l'occupation des parkings tout au long d'une journée ne peut être connue parfaitement car les mouvements poste-à-poste ne sont pas décrits. Dans ce contexte, seules les occupations de parkings sûres seront considérées :

- Pour un couple arrivée - départ corrélé (même immatriculation d'avion) ne changeant pas de parking, l'avion sera supposé rester au parking du débarquement à l'embarquement (des passagers ou des marchandises) : le parking sera donc réservé à l'avion pendant toute la période intermédiaire.
- Pour les autres mouvements (près d'un quart du trafic), le parking ne sera réservé que pendant la phase de débarquement (pour une arrivée) ou d'embarquement (pour un départ) : un paramètre T_{pkg} fixe alors le temps d'occupation du parking pour ces deux phases.

Pour affecter un parking aux plans de vol incomplets sans risquer de créer d'interférences entre les mouvements, il suffira de chercher un parking libre pendant la période d'occupation relative au plan de vol (si un tel parking existe). Lorsque plusieurs parkings sont disponibles, ce choix est orienté en fonction des règles opérationnelles suivantes :

1. Les aéroports correspondent à un ensemble de parkings voisins. Lorsque le parking d'un plan de vol est précisé mais ne correspond à aucun parking connu, il peut être échangé avec un parking du même aéroport (lorsque ce dernier est précisé). Cette règle sera utilisée en priorité car elle assure que le cheminement du mouvement sera proche de son cheminement réel (ce cas correspond à la création d'un nouveau parking dans l'aéroport).
2. Une compagnie utilise souvent les mêmes parkings. Lorsqu'un de ces parkings est libre, il pourra être affecté au mouvement : pour ce faire, la compagnie est déduite de l'identifiant OACI du plan de vol (les lettres d'en-tête correspondent à l'identifiant de la compagnie), et chaque compagnie est associée à l'ensemble des parkings de ses plans de vol.
3. Si aucun parking ne peut être affecté par les règles 1 et 2, le plan de vol ne sera pas pris en

	Unicité du QFU / Configuration et type de mouvement		
	/ Parking	/ Orig.-Dest.	/ Parking et Orig.-Dest.
	18/06/1999	61%	61%
24/03/2000	47%	55%	97%
29/01/2001	43%	47%	96%
22/02/2002	49%	56%	96%

TAB. C.2 – Observations des affectations de QFU à Roissy

compte (car choisir un parking totalement au hasard ne semble pas une solution réaliste).

Dans ces conditions, le tableau C.1 donne les pourcentages d'affectation de parkings et de vols non pris en compte pour des fichiers de plans de vol de différentes années, avec des descriptions de Roissy et d'Orly datant de 1999 : on constate que le nombre de réaffectations par aérogare (dues à des parkings précisés mais inconnus) augmente en 2002, mais que la grande majorité des vols (plus de 99%) est finalement prise en compte.

C.3 Affectation des pistes

C.3.1 Choix d'un QFU

La piste et son sens d'utilisation est définie par le *QFU* : compléter les plans de vol pour lesquels le QFU n'est pas précisé nécessite la connaissance de la configuration de l'aéroport au moment du mouvement. De plus, des procédures d'exploitation peuvent être définies en fonction de la provenance (pour une arrivée) ou de la destination (pour un départ) du vol :

- À Orly, seules deux pistes sont utilisées : l'une est dédiée aux départs, l'autre aux arrivées. L'affectation de QFU ne dépend donc que de la configuration de l'aéroport (face à l'Est ou face à l'Ouest).
- À Roissy, trois pistes sont utilisées actuellement, dont une est partagée par les arrivées et les départs (la piste Nord). Deux QFU sont donc utilisables pour un mouvement d'un type donné. L'étude du trafic (tableau C.2) montre qu'étant donné une configuration de l'aéroport et un type de mouvement, le QFU affecté n'est pas directement corrélé au parking ni à l'origine ou à la destination du vol, mais plutôt à la réunion de ces deux informations (plus de 96% des cas sont ainsi reliés à un unique QFU).

Dans le cas général (sans présager du nombre de pistes et des pratiques opérationnelles), le QFU d'un mouvement, lorsqu'il n'est pas précisé, sera choisi aléatoirement parmi les QFU utilisés dans la même configuration par les mouvements de même type (arrivée ou départ), de même parking et de même origine ou destination.

Une précaution doit cependant être prise concernant les arrivées, pour assurer que les séquences d'atterrissages, qui seront considérées comme une contrainte du problème, restent réalisables : dans certains cas, l'affectation de QFU nécessitera un décalage des heures d'atterrissage, comme cela est précisé dans la partie C.4.

Départs à Roissy				
QFU	Bretelle	L	M	H
26R	W7	5	52	
	W9		53	
	W10		56	34
27L	17		61	
	18		33	
	19	5	66	73

Arrivées à Roissy				
QFU	Bretelle	L	M	H
26L	V2		11	16
	V3	2	160	32
	V4	13	59	2
27L	10	11	137	6
	11		32	20
	13		4	10

Départs à Orly				
QFU	Bretelle	L	M	H
25	W41	2	235	25
	W42	50	17	

Arrivées à Orly				
QFU	Bretelle	L	M	H
26	W34	43	17	
	W35	16	219	10
	W36		8	5
	W37	1	4	3

TAB. C.3 – Observation des affectations des bretelles de piste

C.3.2 Choix des bretelles de piste

Pour définir les chemins devant être envisagés pour chaque mouvement, il est également nécessaire de fixer l'ensemble des bretelles de libération ou d'accès à la piste qui peuvent être empruntées par l'avion. Dans la réalité, la bretelle empruntée est le résultat d'un accord entre les pilotes (ou les compagnies) et les contrôleurs (ou les procédures d'exploitation de l'aéroport) :

- Pour les départs, la bretelle doit être compatible avec la distance de décollage réglementaire pour l'avion, mais cette règle est rarement restrictive sur des aéroports comme Roissy ou Orly. Le choix est en réalité fonction des catégories de turbulence de sillage des avions (tableau C.3), même si certaines bretelles sont partagées par plusieurs catégories.
- Pour les arrivées, une bretelle est souvent conseillée par le contrôle au sol, mais le choix final revient au pilote, en fonction du déroulement de l'atterrissage : le tableau C.3 montre une plus grande diversité des bretelles utilisées pour libérer la piste.

Dans le modèle, une stratégie d'affectation de bretelles est donc décrite, sous forme de données ajoutées en entrée : cette stratégie fixe un sous-ensemble de bretelles en fonction des informations clefs disponibles sur le vol (comme la catégorie ou le type d'avion). La définition de ces affectations reste naturellement évolutive, pour rendre possible la comparaison de différentes stratégies.

C.4 Début des mouvements

Pour le contrôle au sol, le début d'un mouvement correspond à l'heure d'atterrissage pour une arrivée et à l'heure où l'avion est prêt à quitter le parking pour un départ. Dans la description des plans de vol, ces heures sont fournies à la minute près, ce qui pose deux principaux problèmes :

- Il faut affiner ces heures pour simuler précisément le trafic : ceci peut être effectué par application d'un bruit gaussien (d'une minute d'amplitude maximale) sur les heures initialement fournies.

	Roissy			Orly		
	Arrivées	Retardées	Retard max.	Arrivées	Retardées	Retard max.
18/06/1999	715	34	60 s	407	52	80 s
24/03/2000	719	36	80 s	380	33	20 s
29/01/2001	797	50	80 s	369	44	25 s
22/03/2002	738	69	80 s	324	23	45 s

TAB. C.4 – Adaptation des heures d’atterrissage

- Il faut également assurer que les séquences d’atterrissages sur chaque piste soient exploitables : en effet, les règles de séparation des mouvements au niveau de la piste (modélisées dans la partie 4.3.1) sont relatives aux séparations « standards », décrites par les opérationnels. Dans les séquences réelles (qui découlent des plans de vol fournis), ces règles ne sont pas forcément respectées (adaptations des règles en fonction du vent ou de procédures exceptionnelles). Ajoutés à ces considérations, l’insertion des arrivées lors de l’affectation de QFU et le brouillage nécessaire des heures d’atterrissage risquent d’aboutir à des situations sans solution pour les simulations.

Le problème des séquences d’atterrissage se situe en réalité à la limite du cadre étudié, puisqu’il est effectué à la fois par les contrôleurs d’approche et par les contrôleurs au sol (comme cela est décrit dans la partie 1.2.1) :

- L’action des contrôleurs « LOC » (harmonisation et adaptation des vitesses en finale) est modélisée dans la partie 4.2.3 en tant que manœuvre imposable aux avions pour résoudre une situation de trafic.
- L’action des contrôleurs d’approche (séparation préalable et plus grossière des arrivées) est quand à elle effectuée lors du traitement initial des heures d’atterrissages. Une durée minimale (fonction du modèle de séparation pour les séquences de piste) est imposée entre chaque atterrissage sur chaque piste : le traitement initial décale donc les arrivées (sans que leur ordre soit changé) lorsque cela est nécessaire.

Le tableau C.4 donne le nombre et l’amplitude des décalages ainsi opérés sur les arrivées, pour quelques échantillons de trafic : ces modifications touchent une faible partie du trafic et restent très faibles (décalages inférieurs à 80 secondes), mais sont essentielles pour assurer l’existence de solutions au problème du flux des arrivées.

Annexe D

Algorithmes de parcours de graphe

D.1 Notations

Dans cette annexe, les notations définies dans les parties 3.3.1 et 6.2.2 sont conservées :

- $\mathcal{G} = (\mathcal{N}, \mathcal{A}, c)$ désigne un *graphe orienté* :
 - \mathcal{N} est un ensemble de *nœuds*, ou *sommets* ;
 - $\mathcal{A} \subseteq \mathcal{N} \times \mathcal{N}$ est un ensemble d'*arcs* reliant ces nœuds ;
 - $c : \mathcal{A} \rightarrow \mathbb{R}^+$ est la *fonction de coût*, associant à chaque arc (i, j) un réel positif ou nul, noté c_{ij} .
- $\mathcal{Fils}(n)$ désigne l'ensemble des nœuds *fils* d'un nœud n :

$$\mathcal{Fils}(n) = \{m \in \mathcal{N} \mid (n, m) \in \mathcal{A}\}$$

- Pour tout couple de nœuds (s, t) de \mathcal{N}^2 , un *chemin* de s à t est une suite finie (non vide) de sommets $(n_i)_{1 \leq i \leq l}$, telle que :

$$\begin{cases} n_1 = s \\ n_l = t \\ \forall i \in \{1, \dots, l-1\}, (n_i, n_{i+1}) \in \mathcal{A} \end{cases}$$

- La relation (\cdot) désigne la *composition* entre nœuds et chemins, qui permet d'ajouter un nœud à un chemin (aussi bien en aval qu'en amont) : un chemin $\gamma = (n_i)_{1 \leq i \leq l}$ de n_1 à n_l peut ainsi s'écrire $(n_1 \cdot n_2 \cdot \dots \cdot n_l)$.
- Le *coût* d'un chemin $\gamma = (n_i)_{1 \leq i \leq l}$ est noté \bar{c}_γ :

$$\bar{c}_\gamma = \sum_{i=1}^{l-1} c_{n_i n_{i+1}}$$

- \mathcal{C}_{st} désigne l'ensemble des chemins du graphe allant de s à t .
- $\bar{c}_{st} = \min_{\gamma \in \mathcal{C}_{st}} \bar{c}_\gamma$, lorsqu'il est défini ($\mathcal{C}_{st} \neq \{\}$), est appelé *coût minimal* de s à t .
- $\gamma = (n_i)_{1 \leq i \leq l}$ est appelé *plus court chemin* lorsque $\bar{c}_\gamma = \bar{c}_{n_1 n_l}$. Il peut naturellement exister plusieurs plus courts chemins (de même coût) entre deux nœuds donnés du graphe.

D.2 Algorithme de Floyd

L'algorithme de Floyd calcule toute la matrice (\bar{c}) des coûts minimaux \bar{c}_{ij} entre tous les couples de nœuds (i, j) du graphe :

1. Initialement, chaque élément \bar{c}_{ij} vaut :
 - 0 si $i = j$;
 - c_{ij} si $(i, j) \in \mathcal{A}$;
 - $+\infty$ sinon.
2. Chaque nœud k est ensuite envisagé comme un nœud de passage d'un chemin entre chaque couple de nœuds (i, j) : si le coût de ce chemin $(\bar{c}_{ik} + \bar{c}_{kj})$ est inférieur à \bar{c}_{ij} , l'élément de la matrice est mis à jour (algorithme IV.1).

Floyd(\mathcal{G}) :

```

soit  $\bar{c}_{ij} : +\infty$ , pour chaque  $(i, j) \in \mathcal{N}^2$ 
pour chaque  $i \in \mathcal{N}$ ,  $\bar{c}_{ii} \leftarrow 0$ 
pour chaque  $(i, j) \in \mathcal{A}$ ,  $\bar{c}_{ij} \leftarrow c_{ij}$ 
pour chaque  $k \in \mathcal{N}$ ,
  pour chaque  $i \in \mathcal{N}$ ,
    pour chaque  $j \in \mathcal{N}$ ,
      si  $\bar{c}_{ik} < +\infty$  et  $\bar{c}_{kj} < +\infty$  alors  $\bar{c}_{ij} \leftarrow \min(\bar{c}_{ij}, \bar{c}_{ik} + \bar{c}_{kj})$ 
retourner  $(\bar{c})$ 

```

Algorithme IV.1 – Floyd : Coûts minimaux entre tous les couples de nœuds

La complexité de l'algorithme est trivialement en $\mathcal{O}(n^3)$, ce qui restreint son application à des graphes de tailles « raisonnables », pour lesquels une méthode d'énumération exhaustive des nœuds est connue. Son avantage est de simplifier ensuite la recherche de plus courts chemins entre tout couple de nœuds : une fois les coûts minimaux calculés, cette recherche est linéaire par rapport à la longueur (nombre de nœuds) du chemin.

D.3 Algorithme de Dijkstra

D.3.1 Description

L'algorithme de Dijkstra [Ahuja 93] permet de calculer tous les coûts minimaux (et un ensemble de plus courts chemins associés) depuis un nœud source s dans un graphe $\mathcal{G} = (\mathcal{N}, \mathcal{A}, c)$ à coûts positifs ou nuls. L'algorithme développe itérativement tous les plus courts chemins depuis s , par ordre de coût croissant. Pour cela, il fait évoluer deux structures à chaque étape (algorithme IV.2) :

- Une file à priorité Ψ de nœuds *générés*, classée par coûts, pour lesquels un chemin depuis s est connu mais n'est pas nécessairement de coût minimal ;
- Une table associant à chaque nœuds n *développé* (pour lequel un plus court chemin depuis s est connu), le coût minimal c_n depuis s et le père p_n de ce nœud dans un plus court chemin de s à n .

```

Dijkstra( $\mathcal{G}, s$ ) :
  soient  $\Phi$  : Table sur  $\mathcal{N}$  et  $\Psi$  :  $\{(0, s, s)\}$  (Nœuds développés et nœuds générés)
  tant que  $\Psi \neq \{\}$ 
    soit  $(c_n, p_n, n) : \text{Extraire\_min}(\Psi)$  (Extraction d'un nœud généré de  $c$  minimal)
    si  $n \notin \Phi$  alors (Si  $n$  n'est pas encore développé)
       $\Phi(n) \leftarrow (c_n, p_n)$  (Mémorisation du coût minimal et du père de  $n$ )
      pour chaque  $m \in \text{Fils}(n)/\Phi, \Psi \leftarrow \Psi \cup \{(c_n + c_{nm}, n, m)\}$  (Génération des fils de  $n$ )
    retourner  $\Phi$  (Coûts minimaux et pères des nœuds)

```

Algorithme IV.2 – Dijkstra : Arbre des plus courts chemins depuis un nœud

D.3.2 Complexité

L'itération principale de l'algorithme de Dijkstra consiste à *développer* un nœud *généré*, ce qui inclut la *génération* de ses fils. Cette opération ne peut s'effectuer qu'une fois par nœud, soient $n = |\mathcal{N}|$ **développements au maximum**.

- Chaque développement d'un nœud nécessite d'extraire de Ψ un nœud de c minimal. Comme Ψ contient au maximum n nœuds, l'extraction peut être réalisée en $\mathcal{O}(\log n)$ opérations, si la file à priorité des nœuds générés est implémentée par un arbre binaire équilibré par exemple.
- Chaque arc descendant du nœud développé permet éventuellement de générer un nœud fils : cette génération nécessite une vérification de non-appartenance à Φ et éventuellement une insertion dans Ψ et peut donc être réalisée en $\mathcal{O}(\log n)$ opérations élémentaires si Ψ et Φ sont implémentés par des arbres binaires équilibrés.

Chaque génération ne peut s'effectuer qu'une fois par arc du graphe, soit **un nombre maximal** $m = |\mathcal{A}|$ **de générations** au total. Globalement, le nombre d'opérations élémentaires s'écrit donc :

$$\mathcal{O}((n + m) \log n)$$

On peut ainsi remarquer que l'algorithme de Dijkstra appliqué itérativement à chaque nœud du graphe donne le même résultat que l'algorithme de Floyd, mais en $\mathcal{O}((n + m)n \log n)$ opérations, ce qui est plus efficace lorsque le degré moyen du graphe (rapport entre m et n) est faible.

De plus, si les coûts du graphe sont discrets et bornés (graphe à coûts unitaires par exemple) et que le nombre K de valeurs possibles des coûts est « raisonnable », Ψ peut être implémenté par un tableau indicé par ces coûts : les extractions s'effectuent alors en moins de K opérations et les insertions sont élémentaires. Dans ce cas, la complexité de l'algorithme de Dijkstra devient $\mathcal{O}(Kn + m)$.

D.4 L'algorithme A*

D.4.1 Description

L'algorithme A* trouve un chemin reliant un nœud source s à l'un des nœuds *terminaux* $\mathcal{T} \subset \mathcal{N}$, dans un graphe $\mathcal{G} = (\mathcal{N}, \mathcal{A}, c)$ à coûts positifs ou nuls (l'algorithme détecte également le cas où un tel chemin n'existe pas).

Il est en cela moins ambitieux que l'algorithme de Dijkstra (qui trouve l'intégralité d'un arbre de *tous* les plus courts chemins depuis s) mais peut (pour cette raison) s'avérer beaucoup plus rapide : l'exploration du graphe est ciblée par une fonction heuristique h , qui donne une estimation $h(n)$ du coût minimal entre un nœud n et les nœuds terminaux \mathcal{T} :

$\forall n \in \mathcal{N}$,
 si $\bar{c}(n)$ désigne le coût minimal de s à n ,
 alors $f(n) = \bar{c}(n) + h(n)$ est une estimation du coût minimal de s à un nœud terminal.

L'exploration du graphe est similaire à celle de l'algorithme de Dijkstra, mais les nœuds sont explorés par f croissants et non plus directement par coûts croissants (algorithme IV.3). Sur ce point, l'algorithme de Dijkstra s'apparente à un A* dont l'heuristique est nulle.

A*($\mathcal{G}, s, \mathcal{T}, h$):

soient $\Phi : \{\}$ et $\Psi : \{(h(s), 0, (s))\}$	(Nœuds développés et nœuds générés)
tant que $\Psi \neq \{\}$	
soit $(f_n, c_n, \gamma_n) : \text{Extraire_Min}(\Psi)$ (Extraction d'un élément de c maximal parmi ceux minimisant f)	
soit n : le nœud final de γ_n	
si $n \in \mathcal{T}$ alors retourner (c_n, γ_n)	(Chemin solution trouvé)
sinon si $n \notin \Phi$ alors	(si n n'est pas encore développé)
$\Phi \leftarrow \Phi \cup \{n\}$	(Développement de n)
pour chaque $m \in \text{Fils}(n)/\Phi$, $\Psi \leftarrow \Psi \cup \{(c_n + c_{nm} + h(m), c_n + c_{nm}, (\gamma_n \cdot m))\}$	
lever Aucun_Chemin	(Tout le graphe a été exploré depuis s sans jamais atteindre un nœud terminal)

Algorithme IV.3 – A* : Recherche d'un chemin entre un nœud source et des nœuds terminaux

D.4.2 Complexité

Dans le cas général, la complexité de l'A* est identique à celle de l'algorithme de Dijkstra (soit $\mathcal{O}((n+m) \log n)$). Cependant, l'efficacité de l'algorithme dépend de la qualité de l'heuristique, qui correspond aux différences entre l'estimation $h(n)$ du coût minimal d'un nœud n vers un nœud terminal et le coût minimal réel, noté $h^*(n)$.

Deux propriétés particulières de l'heuristique sont souvent utilisées :

- **Une heuristique minorante** ($\forall n \in \mathcal{N}, h(n) \leq h^*(n)$) garantit l'obtention d'un plus court chemin s'il en existe un (mais ne modifie pas la complexité).
- **Une heuristique presque parfaite** ($\forall (u, v) \in \mathcal{N}^2, h(u) < h(v) \implies h^*(u) < h^*(v)$) implique que l'algorithme ne va développer que les nœuds d'un plus court chemin de s à un nœud terminal. La complexité de l'algorithme est donc réduite à $\mathcal{O}(n)$, mais dans ce cas, l'A* est souvent moins efficace que d'autres algorithmes utilisant cette propriété forte de l'heuristique à meilleur escient (un *Branch & Bound* avec recherche *meilleur en premier* par exemple).

D.5 Algorithme de type *Branch & Bound*

Le terme très général de *Branch & Bound* désigne une famille d'algorithmes de recherche qui peuvent s'avérer très différents les uns des autres et qu'il est parfois difficile de présenter sur le même plan : cette partie est essentiellement relative aux algorithmes utilisés pour la résolution de conflit par la méthode 1 contre n , ce qui n'en donne certainement qu'un aperçu simplifié.

Globalement, les algorithmes de *Branch & Bound* poursuivent le même objectif que celui d'un A*, à savoir la recherche d'un plus court chemin reliant un nœud source s à l'un des nœuds terminaux $\mathcal{T} \subset \mathcal{N}$ dans un graphe $\mathcal{G} = (\mathcal{N}, \mathcal{A}, c)$ à coûts positifs ou nuls. Cependant, ces algorithmes sont

également souvent utilisés pour rechercher le minimum d'une fonction f définie sur un espace de recherche \mathcal{E} : dans ce cas, l'exploration de \mathcal{E} doit être transposée en un parcours d'arbre, défini comme suit :

- Chaque nœud représente un sous-ensemble $\mathcal{S} \subset \mathcal{E}$ et ses nœuds fils correspondent à une partition de \mathcal{S} . Pour définir cette partition, un **principe de séparation**, détermine si deux points de \mathcal{S} doivent ou non être représentés par le même nœud fils.
- Le nœud source s représente tout l'espace de recherche tandis que les nœuds terminaux à atteindre sont des singletons de solutions admissibles.
- Les coûts associés aux arcs doivent être définis de telle façon que le coût d'un chemin reliant s à un nœud terminal $t = \{x\}$ (avec $x \in \mathcal{E}$) corresponde à $f(x)$.

L'exploration du graphe diffère de celle d'un A* sur les points suivants :

- **Une stratégie d'exploration** définit l'ordre dans lequel les nœuds du graphe seront visités : l'exploration n'est donc pas strictement définie par une heuristique et peut par exemple s'effectuer *en profondeur d'abord*. Ceci implique que le premier chemin admissible trouvé ne correspond pas forcément à une solution optimale : l'algorithme ne s'arrête généralement pas au premier nœud terminal rencontré et doit reprendre l'exploration plus haut dans l'arbre par un mécanisme de *backtrack* à définir.
- **Des bornes permettent d'abandonner l'exploration** de certaines branches, lorsque les coûts des solutions qu'elles sont susceptibles de contenir se révèlent sous-optimaux ou en dehors d'un objectif fixé. Pour cela, deux types de bornes, mises à jour au fur et à mesure de l'exploration, peuvent être associées à deux types d'heuristiques :
 - La borne supérieure définit le coût maximal des solutions recherchées (en général, elle correspond au coût d'une solution déjà trouvée, mais qui n'est pas forcément optimale). Elle peut être associée à une heuristique minorant le coût de la partie du chemin restant à développer, pour éliminer le plus de branches possible.
 - La borne inférieure définit le coût minimal des solutions recherchées (elle peut correspondre à la borne supérieure d'une exploration antérieure infructueuse, qui démontre qu'aucune solution ne peut être trouvée avec un coût inférieur à cette borne). Pour être utilisable, elle doit être associée à une heuristique majorant le coût d'un éventuel chemin solution passant par le dernier nœud développé (si un tel chemin existe, son coût ne peut dépasser cette valeur).

Lorsque l'exploration d'une branche est abandonnée à cause d'une borne, le mécanisme de *backtrack* doit définir l'endroit où l'exploration peut reprendre.

L'algorithme IV.4 est un exemple de *Branch & Bound* avec exploration *meilleur en premier* et *en profondeur d'abord* : il n'utilise qu'une borne supérieure et explore les nœuds fils dans l'ordre donné par une heuristique h , supposée minorante. Le mécanisme de *backtrack* est ici présenté dans sa forme la plus simple, puisqu'il est directement induit par le caractère récursif de la fonction d'exploration. Le principal avantage de cet algorithme, par rapport à un A*, est de ne pas nécessiter l'insertion et l'extraction (coûteuses en temps de calcul) de nœuds dans un ensemble classé par coûts : dans beaucoup de problèmes, cette stratégie d'exploration du graphe s'avère plus rapide bien que l'algorithme explore plus de nœuds, comme cela est illustré dans la partie 7.5.2, lors de la résolution du problème 1 contre n .

```

BB( $\mathcal{G}, s, \mathcal{T}, h$ ):
  soit  $B : \infty$                                      (Borne supérieure)
  soit  $\gamma : ()$                                    (Chemin trouvé)
  Explorer( $\gamma_n, c_n$ ):                             (Exploration du nœud  $n$ )
    soit  $n$  : nœud final de  $\gamma_n$ 
    si  $n \in \mathcal{T}$  alors
      si  $c_n < B$  alors                                (Un chemin plus court est trouvé)
         $B \leftarrow c_n$ 
         $\gamma \leftarrow \gamma_n$ 
      sinon si  $c_n + h(n) < B$  alors                    (Exploration dans l'ordre donné par l'heuristique)
        pour chaque  $m \in \mathcal{Fils}(n)$  dans l'ordre donné par  $c_{nm} + h(m)$ 
          Explorer( $(\gamma_n \cdot m), c_n + c_{nm}$ )
  Explorer( $(s), 0$ )
  retourner  $(\gamma, B)$ 

```

Algorithme IV.4 – *Branch & Bound* avec recherche *meilleur en premier*

Glossaire

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.

APR : *Airport operations PRogramme*. Département d'EUROCONTROL chargé des projets sur la gestion des aéroports.

ASMGCS : *Advanced Surface Movement Guidance and Control System*. Concepts définis par l'OACI pour le futur système de contrôle des avions au sol.

ATC : *Air Traffic Control*. Système de contrôle du trafic aérien.

ATFM : *Air Traffic Flow Management*. Régulation des flux de trafic aérien.

ATHOS : *Airport Tower Harmonised Controller System*. Projet lancé par la Commission Européenne, étudiant l'amélioration de la position de contrôle d'aéroport.

ATM : *Air Traffic Management*. Gestion du trafic aérien.

ATT : *Aéroports, Tours et zones Terminales*. Division du CENA étudiant le contrôle d'aéroport et d'approche.

CAUTRA : *Coordonnateur AUtomatique du TRafic Aérien*. Système assurant les traitements et la diffusion des plans de vol aux contrôleurs.

CDM : *Collaborative Decision Making*. Projet visant à améliorer le partage et la diffusion des informations entre tous les intervenants du transport aérien.

CENA : *Centre d'Etudes de la navigation Aérienne*.

CFMU : *Central Flow Management Unit*. Cellule européenne de régulation des flux de trafic.

Clairance : Autorisation donnée par le contrôle à un pilote, pour le roulage, le décollage, l'atterrissage ou l'entrée dans un espace contrôlé.

Conflits entre avions : Risques de pertes de séparation.

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.

DEFAMM : *Demonstration Facilities for Airport Movement Management*. Projet de la Commission Européenne visant à centraliser les systèmes d'aide au contrôle d'aéroport et proposant notamment l'affectation automatique de chemins aux avions.

DGAC : *Direction Générale de l'Aviation Civile*.

D-GPS : *Differential Global Positionning System*.

DMAN : *Departure Manager*. Système diffusant les informations sur les vols au départ d'un aéroport vers les espaces qu'il va traverser.

DST : *Decision Support Tools*. Outils d'aide aux contrôleurs.

EDIT : *Ecole Doctorale d'Informatique et Télécommunications*.

ENAC : *Ecole Nationale de l'Aviation Civile*.

EUROCAE : *European Organisation for Civil Aviation Equipment*. Organisme européen étudiant et développant les équipements relatifs à la gestion du trafic aérien.

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.

GPS : *Global Positionning System*. Système de positionnement par satellite provenant du département de la défense américaine.

IFR : *Instrument Flight Rules*. Règles de vol aux instruments.

Incursion piste : Pénétration sans autorisation de tout mobile dans l'aire de piste.

Hub : Centralisation des lignes d'une compagnie sur un aéroport, pour faciliter les correspondances des passagers.

LEEA : *Laboratoire d'Économie et d'Économétrie de l'Aérien*. Structure de recherche commune à l'ENAC et au CENA.

LEONARDO : *Linking Existing ON-Ground, Arrival and Departure Operations*. Projet de la Commission Européenne étudiant l'intérêt d'une centralisation des données des compagnies et du contrôle aérien, dans le cadre du CDM.

LOG : *Laboratoire d'Optimisation Globale*. Structure de recherche commune à l'ENAC et au CENA.

Low cost : Principes de fonctionnement de nouvelles compagnies, visant à diminuer les coûts d'exploitation et de transport.

MACAD : *MANTEA Airfield Capacity and Delay*. Système d'anticipation des retards pour MAN-TEA.

MANTEA : *Management of Surface Traffic in European Airports*. Projet de la Commission Européenne définissant de nouveaux outils d'assistance au contrôle d'aéroport.

MI : *Mathématiques et Informatique*. Département de l'ENAC.

NASA : *National Aeronautics and Space Administration*. Agence nationale américaine de l'aéronautique et de l'espace.

Niveau de vol : Altitude de croisière d'un vol (exprimée en centaines de pieds).

NLR : *Nationaal Lucht & Ruimtevaartlaboratorium*.

OACI : *Organisation de l'Aviation Civile Internationale*.

ODS : *Operational Display System*. Interface du poste de contrôle, présentant les informations données par le radar aux contrôleurs.

OPAL : *Optimisation Platform for Airports, including Land-side*. Plateforme d'analyse de l'ensemble des opérations aéroportuaires, développée par le NLR.

- Plan de vol** : Ensemble des informations décrivant les intentions des compagnies pour chacun de leurs vols.
- Push-back** : Sortie d'un parking en marche arrière, avec l'aide d'un véhicule tracteur.
- Remise de gaz** : Manœuvre d'urgence consistant à annuler un atterrissage en cours dont la décision appartient au pilote ou aux contrôleurs.
- RFM** : *Réseaux, Flux et Modélisation*. Division du CENA étudiant les possibilités d'améliorations du contrôle aérien en route.
- Secteur de contrôle** : Espace aérien délimité sous la surveillance d'une position élémentaire de contrôle.
- SIMMOD** : *SIMulation MODel*. Simulateur de trafic aérien développé par la FAA.
- SMA** : *Surface Movement Advisor*. Projet d'optimisation du contrôle d'aéroport, mené par la FAA.
- SMGCS** : *Surface Movement Guidance and Control System*. Concepts définis par l'OACI, ayant aboutis à l'actuel système de surveillance du trafic au sol.
- SMS** : *Surface Management System*. Outils d'aide au contrôle d'aéroport, développé par la FAA.
- Stack** : Hypodromes d'attente en l'air (dans les secteurs d'approche) pour les vols à l'arrivée.
- Strips** : Supports papier de format prédéfini permettant le suivi d'un avion par les contrôleurs.
- TAAM** : *Total Airspace and Airport Modeler*. Simulateur de trafic aérien.
- Taxiway** : Voie de circulation au sol, matérialisée par une ligne continue que doit suivre le pilote.
- 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.
- UCESO** : *Unités de Contrôle d'Espace Simultanément Ouvrables*. Ensemble des configurations des centres de contrôle (définies comme des partitions de ses secteurs élémentaires).

Bibliographie

- [Adamson 02] P. ADAMSON. A-SMGCS history and EUROCONTROL project. *EUROCONTROL*, 2002. www.eurocontrol.int/airports/projects/asmgcs. Cité p 16
- [Adamson 03a] P. ADAMSON. *Operational Concept & Requirements for A-SMGCS Implementation Level I*. EUROCONTROL, septembre 2003. Cité p 16
- [Adamson 03b] P. ADAMSON. *Operational Concept & Requirements for A-SMGCS Implementation Level II*. EUROCONTROL, septembre 2003. Cité p 16
- [Ahuja 93] R.K. AHUJA, T.L. MAGNANTI et J.B. ORLIN. Network flows, theory, algorithms and applications. *Prentice Hall*, 1993. Cité p 60, 194
- [Alliot 92] Jean-Marc ALLIOT et Thomas SCHIEX. *Intelligence Artificielle et Informatique Théorique*. Cepadues, 1992. ISBN : 2-85428-324-4. Cité p 104
- [Alliot 96] Jean-Marc ALLIOT. *Techniques d'optimisation stochastique appliquées aux problèmes du contrôle aérien*. Thèse d'habilitation : Institut National Polytechnique de Toulouse, 1996. Cité p 108
- [Atkins 03] S. ATKINS, C. BRINTON, D. WALTON, K. ARKIND et P. MOERTL. Results from the initial surface management system field tests. *Proc. of the Air Traffic Management R&D Seminar*, 2003. Cité p 17
- [Auger 04] Anne AUGER, Marc SCHOENAUER et Nicolas VANHAECKE. LS-CMA-ES : a second-order algorithm for covariance matrix adaptation. *Proc. of PPSN VIII Eighth International Conference on Parallel Problem Solving from Nature*. Edited by SPRINGER. 2004, p 182–191. Cité p 104
- [Barnier 97] Nicolas BARNIER. *Optimisation par hybridation d'un algorithme génétique avec la programmation par contraintes*. Thèse : Institut National Polytechnique de Toulouse, 1997. Cité p 125
- [Beyer 01] H.-G. BEYER. *The Theory of Evolution Strategies, Natural Computing Series*. Springer, 2001. Cité p 98
- [Bourdon 98] B. BOURDON. Moyen d'aide à l'écoulement séquencé du trafic. *STNA, Revue technique n° 55*, novembre 1998. Cité p 14
- [Bowen 96] J. BOWEN et G. DOZIER. Constraint satisfaction using a hybrid evolutionary hill-climbing algorithm that performs opportunistic arc and path revision. *Proc. of A.A.A.I.-96*, 1996, p 326–331. Cité p 126
- [Bridges 91] C.L BRIDGES et D.E GOLDBERG. An analysis of multipoint crossover. *Proceedings of the Foundation Of Genetic Algorithms*. 1991, p 301–315. Cité p 103
- [Catoni 90] O. CATONI. *Large deviations for Annealing*. Thèse : Université de Paris XI, 1990. Cité p 104

- [Cerf 94] R CERF. *Une Théorie Asymptotique des Algorithmes Génétiques*. Thèse : Université Montpellier II (France), 1994. *Cité p 104*
- [Clarke 63] S. CLARKE, A. KRIKORIAN et J. RAUSEN. Computing the N best loopless paths in a network. *J. SIAM*, 1963, vol 11, p 1096–1102. *Cité p 67*
- [Cooper 03] W. COOPER, S. LANG, A. MUNDRA, C. LUNSFORD et J. TITTSWORTH. A phased approach to increase airport capacity through safe reduction of existing wake turbulence constraints. *Proc. of the Air Traffic Management R&D Seminar*, 2003. *Cité p 12*
- [Day 84] W. DAY et H. EDELSBRUNNER. Efficient algorithms for agglomerative hierarchical clustering methods. *Journal of Classification*, 1984, vol 1-24. *Cité p 107*
- [De Clercq 03] G. DE CLERCQ. *Enhancing Airside Capacity*. EUROCONTROL, septembre 2003. *Cité p 16*
- [Donohue 03] G. DONOHUE, R. HAYNIE et D. RUTISHAUSER. Measurements of aircraft wake vortex separation at high arrival rates and a proposed new wake vortex separation philosophy. *Proc. of the Air Traffic Management R&D Seminar*, 2003. *Cité p 12*
- [Dreyfus 69] S. E. DREYFUS. An appraisal of some shortest path algorithms. *Operations Research*, 1969, vol 17, p 395–412. *Cité p 64*
- [Drého 03] J. DRÉHO, A. PÉTROWSKI, P. SIARRY et E. TAILLARD. *Métaheuristiques pour l'Optimisation Difficile*. Eyrolles, 2003. ISBN : 2-212-11368-4. *Cité p 2*
- [Durand 96a] Nicolas DURAND. *Optimisation de Trajectoires pour la Résolution de Conflits en Route*. Thèse : Institut National Polytechnique de Toulouse, 1996. *Cité p 36, 41, 54, 55, 113*
- [Durand 96b] Nicolas DURAND, Jean-Marc ALLIOT et Joseph NOAILLES. Genetic operators adapted to partially separable functions. *Submitted to Evolutionary Programming*. EP, 1996. *Cité p 108*
- [Eiben 99] A.E. EIBEN, R. HINTERDING et Z. MICHALEWICZ. Parameter control in evolutionary computation. *IEEE Trans. on Evolutionary Computation*. 1999, p 124–141. *Cité p 104*
- [Eppstein 98] D. EPPSTEIN. Finding the k shortest paths. *SIAM J. Computing*, 1998, vol 28, n°2, p 652–673. www.ics.uci.edu/~eppstein/pubs/Epp-SJC-99.pdf. *Cité p 60*
- [FAA 98] FAA et ATAC CORPORATION. *SIMMOD's User's Manual*. Mountain View, 1998. www.atac.com/simmod. *Cité p 15*
- [Fogel 66a] L. J. FOGEL, A. J. OWENS et M. J. WALSH. *Artificial Intelligence through Simulated Evolution*. Wiley, 1966. *Cité p 98*
- [Fogel 66b] L.J FOGEL, A.J OWENS et M.J WALSH. *Artificial Intelligence Through Simulated Evolution*. Wiley and sons. NY, 1966. *Cité p 103*
- [Freidlin 83] M.I FREIDLIN et A.D WENTZELL. *Random Perturbations of Dynamical Systems*. Springer-verlag, New-York, 1983. *Cité p 104*
- [Freuder 95] E. FREUDER, R. DECHTER, M. GINSBERG, B. SELMAN et E. TSANG. Systematic versus stochastic constraint satisfaction. *Proc. of I.J.C.A.I-95*, 1995, vol 2, p 2027–2032. *Cité p 124*

- [Gilbo 03] E. GILBO. Arrival/departure capacity tradeoff optimization : a case study at the st. louis lambert international airport (stl). *Proc. of the Air Traffic Management R&D Seminar*, 2003. *Cité p* 13
- [Goldberg 89] David GOLDBERG. *Genetic Algorithms*. Addison Wesley, 1989. ISBN : 0-201-15767-5. *Cité p* 98, 102, 104
- [Goldberg 91] D.E GOLDBERG. Real-coded genetic algorithms, virtual alphabets and blocking. *Complex Systems*, 1991, vol 5, p 139–167. *Cité p* 101
- [Gondran 85] M. GONDRAN et M. MINOUX. *Graphes et algorithmes*. Eyrolles, 1985. *Cité p* 124
- [Gotteland 01] J-B. GOTTELAND, N. DURAND, J-M. ALLIOT et E. PAGES. Aircraft ground traffic optimization. *Air Traffic Management R&D Seminar*, décembre 2001. atm2001.eurocontrol.fr. *Cité p* 2
- [Gotteland 02] J-B. GOTTELAND. Optimisation du roulage sur les grands aéroports. *Colloque des Doctorants de l'EDIT*, février 2002. *Cité p* 2
- [Gotteland 03a] J-B. GOTTELAND. Respect des créneaux CFMU par le contrôle au sol. *Colloque des Doctorants de l'EDIT*, février 2003. *Cité p* 2
- [Gotteland 03b] J-B. GOTTELAND, N. DURAND et J-M. ALLIOT. Genetic algorithms applied to airport ground traffic optimization. *Proc. of Congress on Evolutionary Computation*, décembre 2003. www.cs.adfa.edu.au/cec_2003. *Cité p* 2
- [Gotteland 03c] J-B. GOTTELAND, N. DURAND et J-M. ALLIOT. Handling CFMU slots in busy airports. *Air Traffic Management R&D Seminar*, juin 2003. atm2003.eurocontrol.fr. *Cité p* 2
- [Granger 02] Géraud GRANGER. *Détection et résolution de conflits aériens : modélisation et analyse*. Thèse : École Polytechnique, 2002. *Cité p* 40, 41, 97, 108, 113
- [Griewank 82] A. GRIEWANK et Ph. L. TOINT. On the unconstrained optimization of partially separable functions. *Nonlinear Optimization 1981*. Edited by M. J. D. POWELL. London and New York, 1982. Academic Press. *Cité p* 108
- [Hogg 96a] T. HOGG. Refining the phase transition in combinatorial search. *Artificial Intelligence.*, 1996, vol 81. *Cité p* 141
- [Hogg 96b] T. HOGG, B.A. HUBERMAN et C.P. WILLIAMS. Phase transitions and the search problem. *Artificial Intelligence.*, 1996, vol 81. *Cité p* 141
- [Holland 62] John HOLLAND. Outline for a logical theory of adaptive systems. *Journal of the Association of Computing Machinery*, 1962, vol 3. *Cité p* 98
- [Horst 95] Reiner HORST et Hoang TUY. *Global Optimization, Deterministic Approaches*. Springer, 1995. *Cité p* 123
- [Idris 98] A.H IDRIS, B. DELCAIRE, I. ANAGNOSTAKIS, W.D. HALL, J.P. CLARKE, R.J. HANSMAN, E. FERON et A.R. ODONI. Observations of departure processes at logan airport to support the development of departure planning tools. *Air Traffic Management R&D Seminar*, 1998. *Cité p* 13, 155
- [Ingber 92] Lester INGBER et Bruce ROSEN. Genetic algorithm and very fast simulated reannealing : a comparison. *Mathematical and Computer Modeling*, 1992, vol 16, n°1, p 87–100. *Cité p* 111

- [Jiménez Pelayo 99] V. M. JIMÉNEZ PELAYO et A. MARZAL VARÓ. Computing the K shortest paths : a new algorithm and an experimental comparison. *Proc. 3rd Worksh. Algorithm Engineering*, juillet 1999. terra.act.uji.es/REA/papers/wae99.ps.gz. Cité p 65
- [Joines 94] J.A. JOINES et C.R. HOUCK. On the use of non stationary penalty functions to solve nonlinear constrained optimization problems with GAs. *Proceedings of the IEEE ICEC 1994*, 1994, p 579–584. Cité p 100
- [Lane 03] R. LANE et E. LAGIOS. *Airport CDM Applications*. EUROCONTROL, juillet 2003. Cité p 16
- [Lawson 97] D. R. LAWSON. Surface movement advisor (SMA). *Proc. of the Air Traffic Management R&D Seminar*, 1997. Cité p 17
- [Lobjois 98] L. LOBJOIS et M. LEMAÎTRE. Coopération entre méthodes complètes et incomplètes pour la résolution de (V)CSP : une tentative d’inventaire. *3ièmes Journées Nationales sur la Résolution Pratique de Problèmes NP-Complets (JPNC-97)*, 1998, p 67–73. Cité p 123
- [Martin 01] P. MARTIN, O. DELAIN et F. FAKHOURY. Collaborative decision making : Results of experiments to identify limitations of information exchanges in stand and gate operations. *Proc. of the Air Traffic Management R&D Seminar*, 2001. Cité p 13
- [Mazure 96] B. MAZURE, L. SAÏS et E. GRÉGOIRE. Boosting complete techniques thanks to local search methods. *Proc. of the Symp. on Math. and A. I.*, 1996. Cité p 124
- [Miart 04] E. MIART. APR (Airport operations PRogramme) business case assessment. *EUROCONTROL*, février 2004. www.eurocontrol.int/airports. Cité p 16
- [Michalewicz 91] Z. MICHALEWICZ et C.Z. JANIKOV. Handling constraints in genetic algorithms. *Proceedings of the Fourth International Conference on Genetic Algorithm*. Edited by Belew R.K et Booker L.B. ICGA, Morgan Kaufmann, 1991, p 151–157. Cité p 100
- [Michalewicz 92] Z. MICHALEWICZ. *Genetic algorithms+data structures=evolution programs*. Springer-Verlag, 1992. ISBN : 0-387-55387-. Cité p 105
- [Monzel 98] F. MONZEL. *DEFAMM (Demonstration Facilities for Airport Movement Management) Final report*. DFS (Deutsche Flugsicherung GmbH), 1998. www.se.dfs.de. Cité p 16
- [Médioni 98] Frédéric MÉDIONI. *Méthode d’optimisation pour l’évitement aérien : systèmes centralisés, systèmes embarqués*. Thèse : École Polytechnique, 1998. Cité p 97
- [Nguyen 04] Huy Hoang NGUYEN. *Coordination des avions pour la résolution de conflits : une approche basée sur le graphe OERT disjonctif*. Thèse : Université de Technologie de Compiègne, 2004. Cité p 164
- [Nicolaon 03] J.-P. NICOLAON, A. VIDAL, P. CRICK et E. FREVILLE. Potential benefits of a time-based separation procedure to maintain the arrival capacity of an airport in strong head-wind conditions. *Proc. of the Air Traffic Management R&D Seminar*, 2003. Cité p 12
- [Page 03] Y. PAGE. *European Action Plan for the Prevention of Runway Incursions*. EUROCONTROL, septembre 2003. Cité p 16

- [Papadimitriou 82] C.H PAPANIMITRIOU et K STEIGLITZ. *Combinatorial Optimization : Algorithms and Complexity*. Prentice Hall, Englewood Clifs. NJ (USA), 1982. Cité p 87, 123
- [Pascoal 98] M. M. B. PASCOAL, J. E. DOS SANTOS et E. DE QUEIRÓS VIEIRA MARTINS. An algorithm for ranking loopless paths. *INFORMS/CORS Montréal 98*. 1998. Cité p 67
- [Paul 99] S. PAUL, K. ZOGRAFOS et H. HESSELINK. *MANTEA (Management of Surface Traffic in European Airports) Final report*. EUROCONTROL, avril 1999. TR 1036 D8.3. Cité p 16
- [Perko 86] A. PERKO. Implementation of algorithms for K shortest loopless paths. *Networks*, 1986, vol 16, p 149–160. Cité p 67
- [Pesic 00] B. PESIC. *Optimisation de la circulation des aéronefs au sol sur la plate-forme de Roissy*. Rapport DEA : INPT, 2000. Cité p 61, 78
- [Pham-Dumesnil 98] M. PHAM-DUMESNIL. *ATHOS (Airport Tower Harmonised Controller System) Final Report*. EUROCONTROL, août 1998. Cité p 15
- [Preston 99] PRESTON. *TAAM Reference Manual*. Preston Aviation Solutions, 1999. www.preston.net. Cité p 15
- [Prestwich 01] Steven PRESTWICH. Local search and backtracking versus non-systematic backtracking. *Papers from the AAAI 2001 Fall Symposium on Using Uncertainty Within Computation*. North Falmouth, Cape Cod, MA, novembre 2001. Cité p 125
- [Radcliffe 94] N.J. RADCLIFFE et P.D. SURRY. Formal memetic algorithms. *Proc. of Evolutionary Computing : AISB Workshop*. Edited by Springer-Verlag T. FOGARTY. 1994. Cité p 126
- [Ramalingam 96] G. RAMALINGAM et T. REPS. An incremental algorithm for a generalization of the shortest path problem. *J. of Algorithms*, 1996, n°21 :267-305. Cité p 73
- [Rechenberg 65] I. RECHENBERG. *Cybernetic Solution Path of an Experimental Problem*. Royal Aircraft Establishment Library Translation, 1965. Cité p 98
- [Ringeisen 87] R. D. RINGEISEN. Implementation of algorithms for K shortest loopless paths. *Mathematical Reviews*, 1987, vol 87j, n°05102. Cité p 67
- [Schwefel 81] H.-P. SCHWEFEL. *Numerical Optimization of Computer Model*. Wiley, 1981. Cité p 104
- [Selman 98] B. SELMAN, H. LEVESQUE et D. MITCHELL. A new method for solving hard satisfiability problems. *Proc. of A.A.A.I-98*, 1998, p 440–446. Cité p 124
- [Simon 92] H.A. SIMON, M. EGIDI, R. MARIS et R. VIALE. *Economics, Bounded Rationality, and the Cognitive Revolution*. Edwar Elgar, 1992. Cité p 17
- [Stoica 02] D. STOICA, A. ACHAIBOU et F. MORA CAMINO. Analyse, représentation et optimisation de la circulation des avions au sol. Approche par les systèmes multi-agents. *Troisième Congrès des Doctorants de l'École Doctorale Système*, mai 2002. Cité p 18
- [Suarez 03] N. SUAREZ. *APPROVE (Advanced airport aProach PROCedures including Validation and Elaboration)*. EUROCONTROL, 2003. www.approve.isdefe.es. Cité p 11

- [Supino 03] M. SUPINO, M. GARCIA-MEDINA, S. SANTIAGO, M. VAN BOVE, C. CAFIERO et C.S. BEERS. *SOURDINE II (Study of Optimisation procedURes for Decreasing the Impact of NoisE around airports) - Definition Of New Noise Abatement Procedures*. EUROCONTROL, mars 2003. CEC121. *Cité p 9*
- [Toulouse 85] G TOULOUSE. *Application de la Physique Statistique aux Problèmes Complexes*. ESPCI, 1985. *Cité p 111*
- [Trouvé 93] A. TROUVÉ. *Parallélisation massive du recuit simulé*. Thèse : Université de Paris XI, 1993. *Cité p 104*
- [Vallée 01] J.-C. VALLÉE. L'évolution de la surveillance des mouvements au sol sur les aéroports. *STNA, Revue technique n°61*, décembre 2001. *Cité p 13, 14*
- [van Eenige 02] M.J.A. VAN EENIGE. *OPAL (Optimisation Platform for Airports, including Land-side)*. NLR (Nationaal Lucht & Ruimtevaartlaboratorium), 2002. www.nlr.nl/public/library/2002/2002-595-dcs.html. *Cité p 15*
- [Wallace 96] R. WALLACE. Enhancements of branch and bound methods for the maximal constraint satisfaction problem. *Proc. of A.A.A.I.-96*, 1996. *Cité p 124*
- [Wright 91] A.H WRIGHT. Genetic algorithms for real parameter optimization. *Proceeding of the Foundation Of Genetic Algorithms*. FOGA, Morgan Kaufmann, 1991, p 205–218. *Cité p 101*
- [Yang 02] S. YANG. Adaptive non-uniform crossover based on statistics for genetic algorithms. *Proc. of the Genetic and Evolutionary Computation Conference, GECCO*. Langdon. W. B., 2002, p 650–657. *Cité p 104*
- [Yen 71] J. Y. YEN. Finding the K shortest loopless paths in a network. *Management Science*, 1971, vol 17, p 712–716. *Cité p 60*
- [Yin 93] X YIN et N GERMA. A fast genetic algorithm with sharing scheme using cluster analysis methods in multimodal function optimization. *Proceedings of the Artificial Neural Nets and Genetic Algorithms*. Edited by SPRINGER-VERLAG. 1993. *Cité p 107*
- [Zilberstein 95] Shlomo ZILBERSTEIN. Operational rationality through compilation of anytime algorithms. *AI Magazine*, 1995, vol 16, n°2, p 79–80. citeseer.ist.psu.edu/zilberstein93operational.html. *Cité p 164*